



**HAL**  
open science

# Reconstruction et analyse automatiques pour le rééclairage d'objets basés-image

Bruno Mercier

► **To cite this version:**

Bruno Mercier. Reconstruction et analyse automatiques pour le rééclairage d'objets basés-image. Autre [cs.OH]. Université de Poitiers, 2006. Français. NNT: . tel-00145124

**HAL Id: tel-00145124**

**<https://theses.hal.science/tel-00145124>**

Submitted on 7 May 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# THÈSE

pour l'obtention du grade de

## Docteur de l'Université de Poitiers

**Faculté des Sciences Fondamentales et Appliquées**

(Diplôme national – arrêté du 7 août 2006)

*Secteur de recherche* : **Informatique**

*École Doctorale* : **Sciences Pour l'Ingénieur**

Présentée par

### BRUNO MERCIER

## Reconstruction et analyse automatiques pour le rééclairage d'objets basés-image

Soutenue le 24/10/2006 devant la Commission d'Examen composée de :

Kadi BOUATOUCH, Professeur à l'Université de Rennes I ..... Rapporteur  
Jean-Michel DISCHLER, Professeur à l'Université Louis Pasteur de Strasbourg ..... Rapporteur  
Alan CHALMERS, Professeur à l'Université de Bristol, Royaume Uni ..... Examineur  
Michel MÉRIAUX, Professeur à l'Université de Poitiers ..... Directeur de Thèse  
Daniel MENEVEAUX, Maître de Conférences à l'Université de Poitiers ..... Co-directeur de Thèse



---

---

# TABLE DES MATIÈRES

---

<b>INTRODUCTION</b>	<b>1</b>
<b>I ETUDE BIBLIOGRAPHIQUE</b>	<b>5</b>
A Rendu Basé-Image (RBI) . . . . .	6
A.1 Rendu sans modèle géométrique . . . . .	6
A.2 Exploitation des champs de lumière . . . . .	14
A.3 Discussions . . . . .	16
B Reconstruction géométrique . . . . .	17
B.1 Acquisition de la géométrie . . . . .	17
B.2 Estimation de la forme . . . . .	19
B.3 Estimation de la normale . . . . .	22
B.4 Calibrage de la caméra . . . . .	24
B.5 Discussions . . . . .	25
C Détection de sources lumineuses . . . . .	26
C.1 Les types de sources lumineuses . . . . .	27
C.2 Utilisation d'une seule image . . . . .	29
C.3 Utilisation d'une ou plusieurs sphères . . . . .	32
C.4 Discussions . . . . .	37
D Estimation de BRDFs . . . . .	37
D.1 Types de surfaces . . . . .	38
D.2 Modèles de BRDFs . . . . .	39
D.3 Acquisition d'un champ de réflectance . . . . .	44
D.4 Estimation d'un modèle paramétrique . . . . .	47
D.5 Discussions . . . . .	50
E Rééclairage d'objets réels . . . . .	51
E.1 Utilisation de BRDFs réelles acquises . . . . .	51
E.2 Utilisation de modèles de BRDFs paramétriques . . . . .	53
E.3 Discussions . . . . .	56
<b>II NOTRE SYSTÈME DE RECONSTRUCTION GÉOMÉTRIQUE</b>	<b>59</b>
A Acquisition . . . . .	60
A.1 Choix du matériel . . . . .	60
A.2 Système d'acquisition . . . . .	61
A.3 Construction de lumigraphes synthétiques . . . . .	74
B Reconstruction de la forme . . . . .	75



B.1	Choix de la méthode . . . . .	76
B.2	Reconstruction basée-voxel . . . . .	77
B.3	Construction de la surface discrète . . . . .	84
B.4	Amincissement de la surface discrète . . . . .	89
B.5	Génération de maillage fermé . . . . .	93
B.6	Utilisation des pixels pour affiner le maillage . . . . .	100
C	Estimation de la normale . . . . .	108
C.1	Choix de la représentation . . . . .	109
C.2	Normales à partir du maillage . . . . .	109
C.3	Normales à partir de la surface discrète . . . . .	111
C.4	Analyse des résultats . . . . .	112
D	Résultats . . . . .	114
D.1	Statistiques sur les rayons projetés . . . . .	114
D.2	Occupation mémoire . . . . .	116
D.3	Complexité des algorithmes . . . . .	117
D.4	Temps de calcul . . . . .	118
D.5	Objets synthétiques reconstruits . . . . .	119
D.6	Objets réels reconstruits . . . . .	119
<b>III</b>	<b>DÉTECTION DE SOURCES LUMINEUSES ET ESTIMATION DE BRDFs</b>	<b>125</b>
A	Représentation des données . . . . .	126
A.1	Rayons incidents . . . . .	126
A.2	Modèles utilisés . . . . .	130
B	Principe général . . . . .	132
B.1	Classification des voxels selon la réflectance . . . . .	133
B.2	Classification des voxels selon l'éclairement . . . . .	136
B.3	Evaluation du type de surface . . . . .	137
B.4	Algorithme général . . . . .	138
B.5	Aspects techniques . . . . .	139
C	Détection d'une source ponctuelle . . . . .	143
C.1	Utilisation de la composante diffuse . . . . .	143
C.2	Utilisation de la composante spéculaire . . . . .	149
D	Détection d'une source directionnelle . . . . .	152
D.1	Utilisation de la composante diffuse . . . . .	152
D.2	Utilisation de la composante spéculaire . . . . .	153
E	Estimation conjointe : sources lumineuses / propriétés de réflectance . . . . .	153
E.1	Surface d'aspect diffus . . . . .	154
E.2	Surface d'aspect diffus directionnel . . . . .	154
E.3	Validation de l'estimation conjointe . . . . .	155
E.4	Résultats . . . . .	156
F	Raffinement sur chaque BRDF . . . . .	160
G	Raffinement sur l'espace image . . . . .	161
<b>IV</b>	<b>RÉÉCLAIRAGE D'OBJETS RÉELS</b>	<b>165</b>
A	Intégration d'objets réels dans des environnements virtuels . . . . .	166
A.1	Champs de lumière précalculés . . . . .	166
A.2	Nouvelle structure de représentation . . . . .	167

B	Compression de champs de lumière . . . . .	168
B.1	Quantification vectorielle . . . . .	168
B.2	Résultats . . . . .	172
C	Rééclairage par lancer de rayons . . . . .	172
C.1	Equation d'illumination locale . . . . .	173
C.2	BRDFs uniformes . . . . .	173
C.3	BRDFs variées . . . . .	176
C.4	Temps de calcul global . . . . .	179
C.5	Objet synthétique rééclairé . . . . .	180
C.6	Objet réel rééclairé . . . . .	180
<b>CONCLUSION</b>		<b>183</b>
<b>ANNEXES</b>		<b>185</b>
<b>I</b>	<b>DÉFINITIONS ET GRANDEURS UTILISÉES</b>	<b>187</b>
A	Les angles solides . . . . .	188
B	Les grandeurs physiques . . . . .	190
C	La fonction de distribution de réflectance bidirectionnelle (BRDF) . . . . .	195
D	L'équation de luminance . . . . .	197
<b>II</b>	<b>ORIGINES DU RENDU BASÉ-IMAGE</b>	<b>199</b>
A	Les différents types de textures . . . . .	200
B	Les premiers modèles de Rendu Basé-Image . . . . .	207
<b>BIBLIOGRAPHIE</b>		<b>211</b>



---

# INTRODUCTION

---

L'utilisation d'objets réels dans le monde numérique est un enjeu important pour de nombreux domaines tels que la réalité augmentée, la cinématographie, les jeux vidéos ou la publicité. Cette problématique relève de plusieurs domaines de recherche : l'analyse et le traitement d'images, la vision par ordinateur et l'informatique graphique.

Un certain nombre de travaux ont porté sur l'acquisition de la forme des objets, de leur réflectance et de leur texture. Dans le contexte de l'héritage culturel et du patrimoine, plusieurs méthodes ont été proposées pour reconstruire des objets avec l'acquisition précise de leur réflectance. Par exemple, P. Callet digitalise des statuettes pour estimer les propriétés physiques de matériaux comme le bronze [Cal04] ; le projet *Michelangelo* [LPC<sup>+</sup>00] a pour objectif de réaliser l'acquisition de grandes statues et une reconstitution virtuelle des derniers vestiges du Parthénon est proposée dans [Deb04].

Ce mémoire de thèse se place dans le contexte du rendu réaliste et notre objectif est de rééclairer des objets réels représentés par un ensemble de photographies. A terme, ces travaux doivent permettre une intégration réaliste d'objets basés-image dans des environnements virtuels.

La difficulté du travail concerne le problème du rééclairage d'objets à partir uniquement d'un ensemble d'images, sans leur forme, sans information de profondeur, sans propriétés photométriques et sans les conditions d'éclairage utilisées lors de l'acquisition. En outre, leur intégration dans des environnements virtuels pose des problèmes supplémentaires car il est impossible de positionner correctement l'objet dans l'environnement, de calculer les ombres portées par la scène sur l'objet et inversement. Les méthodes existantes dans la littérature traitent l'estimation d'une partie des paramètres en supposant les autres connus. Le travail réalisé au cours de cette thèse propose une chaîne complète permettant de retrouver l'ensemble des informations nécessaires au rééclairage d'objets basés-image.

Les séries d'images que nous utilisons peuvent être représentées par un champ de lumière. Il s'agit d'une structure de représentation basée-image permettant une visualisation interactive pour n'importe quel nouveau point de vue. Seuls les paramètres intrinsèques et extrinsèques de la caméra sont connus et l'acquisition est effectuée avec des conditions d'illumination fixes mais non connues. La surface de l'objet peut comporter à la fois des textures, des régions spéculaires et des régions diffuses.

L'acquisition des objets est réalisée à l'aide d'un système mis en œuvre au laboratoire dans le cadre de cette thèse. Un mécanisme de compression efficace, basé sur une mesure visuelle qualitative, permet de réduire le coût de stockage de lumigraphes tout en préservant une visualisation réaliste.

A partir d'un tel système de représentation d'objets, le problème du rééclairage est mal posé et non solvable. Toutefois, en combinant diverses analyses à partir des images, nous pouvons

extraire les informations utiles à la reconstruction et à l'estimation des paramètres nécessaires au rééclairage.

Pour cela, nous proposons une méthode de reconstruction géométrique basée sur un algorithme de sculpture utilisant la silhouette de l'objet. La géométrie reconstruite est nécessaire pour guider l'estimation de la réflectance de l'objet. Nous étendons l'algorithme des *marching cubes* pour affiner la géométrie reconstruite en utilisant les pixels des images. Nous proposons également deux méthodes d'estimation des normales utiles pour la poursuite de l'analyse et pendant le rééclairage.

A partir de la forme et de la normale de l'objet, l'utilisation des images permet de déterminer si la surface est plutôt diffuse ou spéculaire. Cette information est utilisée pour retrouver une première estimation (i) des caractéristiques des sources lumineuses et de leur position / direction et (ii) une réflectance approximative de l'objet. Un algorithme itératif raffine conjointement la position des sources lumineuses et la réflectance des objets. Les résultats obtenus montrent que la méthode proposée est à la fois robuste et précise pour des objets réels.

Nous proposons également une première méthode de rééclairage utilisant les différents paramètres reconstruits. Les premiers résultats obtenus sont visuellement encourageants et nous prévoyons une étude plus approfondie sur ce sujet.

L'originalité du travail proposé dans ce mémoire, comparé aux méthodes précédentes, concerne les hypothèses posées sur les images utilisées. En effet, la plupart des méthodes de reconstruction, que ce soit pour la géométrie ou pour la fonction de réflectance, utilisent soit des objets parfaitement diffusés, soit un nombre de sources connues, soit un seul type de sources. Souvent une sphère spéculaire ou diffuse est ajoutée lors de la prise de vue.

Voici de notre point de vue les contributions les plus importantes de ce mémoire :

- la mise en place d'un système d'acquisition d'objets réels ;
- la prise en compte de critères psycho-visuels pour la compression de lumigraphes par méthode de quantification ;
- l'utilisation des pixels des images pour raffiner le maillage géométrique de l'objet lors de la méthode des *marching cubes* ;
- une méthode d'estimation conjointe de la réflectance de l'objet et des sources lumineuses qui l'éclairent ;
- une structure de données efficace pour stocker les informations reconstruites (maillage, normale, réflectance) ;
- une méthode permettant de rééclairer les objets reconstruits de manière réaliste.

Le travail présenté dans ce mémoire est organisé de la manière suivante. Le chapitre **I** propose une étude bibliographique complète sur les sujets du rendu à base d'images, de la reconstruction géométrique d'objets, de l'estimation des sources de lumière et de la réflectance d'objets réels pour le rééclairage.

Le chapitre **II** décrit le système d'acquisition que nous avons mis en œuvre pour produire des images calibrées d'objets réels et des lumigraphes. La méthode de reconstruction géométrique que nous proposons est également détaillée dans cette partie.

Le chapitre **III** expose notre méthode d'estimation conjointe des sources lumineuses et de la réflectance des objets en s'appuyant sur la géométrie reconstruite.

Enfin, nous proposons un premier système de rééclairage d'objets réels utilisant l'ensemble des informations reconstruites : géométrie et réflectance (chapitre **IV**).

Après la conclusion et les perspectives de ce travail, une série d'annexes rappelle la définition des termes physiques employés (annexe **I**) et décrit l'évolution des méthodes de rendu à base d'images (annexe **II**).



---

# ETUDE BIBLIOGRAPHIQUE

---

## Sommaire

---

<b>A</b>	<b>Rendu Basé-Image (RBI)</b> . . . . .	<b>6</b>
A.1	Rendu sans modèle géométrique . . . . .	6
A.2	Exploitation des champs de lumière . . . . .	14
A.3	Discussions . . . . .	16
<b>B</b>	<b>Reconstruction géométrique</b> . . . . .	<b>17</b>
B.1	Acquisition de la géométrie . . . . .	17
B.2	Estimation de la forme . . . . .	19
B.3	Estimation de la normale . . . . .	22
B.4	Calibrage de la caméra . . . . .	24
B.5	Discussions . . . . .	25
<b>C</b>	<b>Détection de sources lumineuses</b> . . . . .	<b>26</b>
C.1	Les types de sources lumineuses . . . . .	27
C.2	Utilisation d'une seule image . . . . .	29
C.3	Utilisation d'une ou plusieurs sphères . . . . .	32
C.4	Discussions . . . . .	37
<b>D</b>	<b>Estimation de BRDFs</b> . . . . .	<b>37</b>
D.1	Types de surfaces . . . . .	38
D.2	Modèles de BRDFs . . . . .	39
D.3	Acquisition d'un champ de réflectance . . . . .	44
D.4	Estimation d'un modèle paramétrique . . . . .	47
D.5	Discussions . . . . .	50
<b>E</b>	<b>Rééclairage d'objets réels</b> . . . . .	<b>51</b>
E.1	Utilisation de BRDFs réelles acquises . . . . .	51
E.2	Utilisation de modèles de BRDFs paramétriques . . . . .	53
E.3	Discussions . . . . .	56

---



## A Rendu Basé-Image (RBI)

Les méthodes conventionnelles de rendu réaliste utilisent un modèle géométrique précis pour décrire l'ensemble des objets composant une scène. Un algorithme d'illumination locale ou globale (lancer/tracé de rayons, tracé de photons, méthode de radiosité, etc) est ensuite utilisé pour générer de nouvelles vues de la scène : la complexité des calculs dépend essentiellement de la description du modèle géométrique (nombre de polygones, nombre d'intersections évaluées par pixel). Les algorithmes d'illumination utilisés en synthèse d'images sont basés sur les grandeurs physiques décrites dans l'annexe I.

Les méthodes de *Rendu Basé-Image (RBI)* décrivent la scène par un ensemble d'images ou de photographies acquises sous différents points de vue. Des mécanismes d'interpolation entre les images originales permettent ensuite de générer de nouvelles vues. Aucun modèle géométrique de la scène n'est requis pour produire les nouvelles images : celles-ci sont donc générées indépendamment de la complexité géométrique de la scène, leur évaluation dépend uniquement de leur résolution.

Néanmoins, seules les méthodes récentes de RBI permettent la production de nouveaux points de vue uniquement à partir d'une série d'images calibrées, i.e. sans utiliser la géométrie de la scène. Ces méthodes sont apparues dans les années 90, après la définition de la *fonction plénoptique* par E. Adelson et J. Bergen [AB91]. Depuis toutes les méthodes de RBI échantillonnent la fonction plénoptique plus ou moins précisément suivant les données utilisées ou les applications proposées.

Nous voulons un système de représentation capable de stocker une série d'images représentatives d'un objet et qui puisse générer n'importe quelle nouvelle vue sans avoir besoin de la géométrie de l'objet. Cette étude bibliographique rappelle les principales méthodes de RBI et explique pourquoi nous avons choisi la structure de données des *champs de lumière* (ou *lumi-graphes*).

L'annexe II peut être consulté pour un survol de l'ensemble des méthodes de rendu utilisant des images mais ne s'appuyant généralement pas sur la définition de la fonction plénoptique : ce sont des méthodes qui utilisent un modèle géométrique (placage de textures) ou un ensemble de vues fixes (navigation 3D interactive).

### A.1 Rendu sans modèle géométrique

Les méthodes utilisant uniquement un ensemble d'images pour représenter une scène réduisent de manière drastique la complexité géométrique de la scène et s'affranchissent également de l'étape de modélisation souvent fastidieuse.

Le rendu d'un nouveau point de vue dépend uniquement de la résolution de l'image générée. Par conséquent il est obtenu en temps constant, quel que soit la complexité de la scène représentée. En contrepartie, la fonction plénoptique doit être échantillonnée de manière dense pour permettre la synthèse de n'importe quel point de vue : ceci entraîne généralement un stockage important de données photographiques.

#### A.1.1 La fonction plénoptique

Cette fonction est définie en 1991 par E. Adelson et J. Bergen pour modéliser l'ensemble des paramètres nécessaires à la description d'un environnement lumineux complexe [AB91].

La fonction plénoptique, définie sur 7 paramètres, permet de représenter tout échange de flux lumineux à l'intérieur d'un environnement (figure 1.1.a).

Cette fonction  $P$  exprime le flux lumineux émis en chaque point de l'espace  $(V_x, V_y, V_z)$ , dans

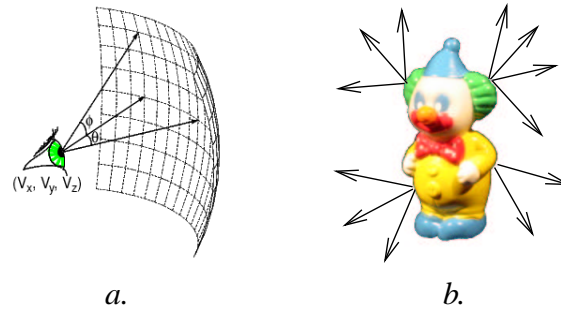


Figure 1.1 – a. Représentation de la fonction plénoptique en un point  $(V_x, V_y, V_z)$  et dans une direction  $(\theta, \phi)$  de l'espace (image extraite de [AB91]); b. la fonction plénoptique peut être utilisée pour modéliser le flux lumineux émis par la surface d'un objet.

chaque direction  $(\theta, \phi)$ , pour chaque longueur d'ondes  $\lambda$  et en chaque instant  $t$  :

$$P_7 = P(V_x, V_y, V_z, \theta, \phi, \lambda, t) \quad (1.1)$$

Comme cette fonction est inutilisable sous sa forme complète, elle peut être simplifiée en fixant le paramètre temps pour des scènes statiques et des conditions d'éclairage fixes, et en fixant la longueur d'onde aux trois composantes de l'espace couleur ( $RGB$ ) :

$$P_5 = P(V_x, V_y, V_z, \theta, \phi) \quad (1.2)$$

Naturellement, il est illusoire de trouver une formulation analytique (même sur 5 dimensions) pour prendre en compte tous les transferts lumineux dans un environnement. En revanche, un échantillonnage judicieux de l'espace permet de représenter une partie exploitable de la fonction plénoptique et de nouvelles méthodes de rendu uniquement basées sur des images calibrées ont pu être développées.

### A.1.2 Les panoramas 2D

Les images panoramiques sont une première représentation de la fonction plénoptique par uniquement deux dimensions  $(\theta, \phi)$ . Par conséquent, cette méthode est restreinte à la représentation d'un environnement depuis un point de vue fixe.

#### Le panorama cylindrique

Le système de navigation 3D *QuickTime VR*, développé par S. Chen en 1995 pour Apple, utilise des images panoramiques cylindriques sur 360 degrés pour représenter un environnement [Che95]. Des points de vue sont acquis à divers endroits de la scène et l'utilisateur peut ensuite revisualiser la scène en ces différents points avec un champ de vue vertical limité. La visualisation de l'environnement est effectuée de manière interactive, mais elle est contrainte à positionner la caméra virtuelle aux centres des points de vue panoramiques. Lorsque la caméra est positionnée

sur un de ces points de vue, l'utilisateur peut librement tourner la caméra autour de lui et effectuer des zooms pour visualiser les détails de l'environnement.

L'utilisateur est contraint à suivre certains chemins prédéfinis lors de ses déplacements : il n'a pas une totale liberté de mouvement dans l'environnement.

### Le panorama sphérique

Le système de création de mosaïques d'images panoramiques, proposé par R. Szeliski et H. Shum en 1997 (*panoramic image mosaics*), permet de construire des cartes d'environnement sphériques à partir d'un ensemble non ordonné d'images représentant un environnement réel [SS97]. Le système a uniquement besoin d'un nombre conséquent d'images avec de multiples recouvrement d'informations entre les images pour reconstruire la mosaïque d'images panoramiques.

Un ensemble d'images acquises par une caméra tenue à la main peut être utilisé pour définir un panorama cylindrique de l'environnement. La méthode permet également de construire des images panoramiques sphériques, similaires aux cartes d'environnement, lorsque les images disponibles couvrent la totalité de l'espace 3D pour chaque caméra (figure 1.2).



Figure 1.2 – Panorama sphérique obtenu à partir de 54 images (image extraite de [SS97]).

Cette méthode permet un échantillonnage complet des paramètres  $\theta$  et  $\phi$  de la fonction plénoptique, contrairement aux panoramas cylindriques où l'angle d'élévation de la caméra est limité. Néanmoins, ce système ne permet que l'acquisition d'environnement en un point fixe et n'est pas adapté à l'acquisition d'objets.

### A.1.3 La modélisation plénoptique

La modélisation plénoptique, développée en 1995 par L. McMillan et G. Bishop sous le nom de *plenoptic modeling*, est le premier système de rendu dit *basé-image* échantillonnant la fonction plénoptique sur cinq dimensions [MB95]. Un échantillonnage complet de la fonction plénoptique 5D peut être vu comme un ensemble de cartes sphériques en des points de vue différents d'une scène statique. Néanmoins, l'échantillonnage proposé dans ce papier construit des cartes sphériques incomplètes en restreignant l'angle d'élévation de la caméra.

La méthode permet, à partir d'un échantillonnage discret de la fonction plénoptique, de générer une représentation quasi-continue de cette fonction sur une portion de l'environnement. Plusieurs panoramas cylindriques en divers points de vue assez proches les uns des autres sont utilisés pour représenter l'environnement à visualiser (figure 1.3). De nouvelles images cylin-

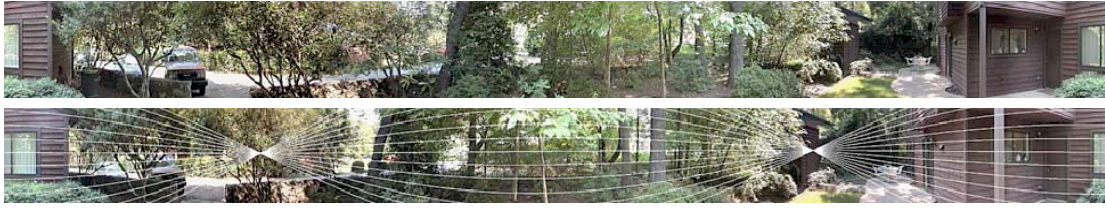


Figure 1.3 – Deux panoramas cylindriques utilisés pour générer de nouveaux panoramas. Plusieurs courbes épipolaires sont représentées sur le second panorama (images extraites de [MB95]).

driques peuvent être générées pour des points de vue choisis à proximité des points de vue originaux : si le nouveau point de vue est trop éloigné, des trous peuvent apparaître sur l'image panoramique générée, correspondant à des régions non visibles sur les vues de référence (exactement les mêmes problèmes de visibilité que dans l'interpolation de vue ou la déformation de vue).

Ce système de modélisation de la fonction plénoptique est proche des panoramas cylindriques : malgré la modélisation de plusieurs panoramas, ce système est centré sur la représentation d'environnements et n'est pas approprié à la représentation d'objets (certaines informations sont inutiles, d'autres sont manquantes).

#### A.1.4 Les champs de lumière / les lumigraphes

En 1996, un nouveau modèle de représentation basée-image est développé pour visualiser soit une scène, soit un objet uniquement à partir d'une série ordonnée d'images. C'est le premier modèle permettant de représenter des objets à partir d'un ensemble d'images. Nous avons choisi de l'utiliser car il offre une représentation compacte et complète de la fonction plénoptique sur 5 dimensions.

Deux variantes de ce modèle ont été proposées simultanément. M. Levoy et P. Hanrahan définissent une structure appelée *champ de lumière* (*light field*) [LH96] tandis que S. Gortler et al. définissent une structure de *lumigraphe* (*lumigraph*) [GGSC96]. Ces deux représentations sont quasi-identiques : la structure de données est similaire, seul l'échantillonnage nécessaire à la visualisation de l'objet diffère.

Les deux systèmes permettent, contrairement aux méthodes présentées précédemment, de générer de nouvelles images d'un objet depuis n'importe quel point de vue de l'environnement 3D de manière interactive. Le désavantage principal concerne la taille importante des informations à stocker sur le disque. La structure est constituée de plusieurs paires de polygones parallèles, appelés *slabs*, maillés de manière régulière. Chaque slab échantillonne une partie de la fonction plénoptique 5D (comme la modélisation plénoptique) : il représente le flux lumineux traversant un volume délimité par deux polygones parallèles  $(u, v)$  et  $(s, t)$  pour toutes les directions définies par le maillage (figure 1.4).

La fonction plénoptique 5D est échantillonnée par la luminance émise depuis une position de l'espace (maille du plan  $(s, t)$ ) vers une direction de l'espace (maille du plan  $(u, v)$ ) : l'absence d'élément occultant dans la scène ou l'objet à visualiser permet de réduire la représentation des informations à 4 dimensions  $(u, v, s, t)$ . La luminance  $(u, v, s, t)$  émise reste invariante sur sa ligne de propagation et la position exacte d'émission d'une luminance n'est pas connue (aucune information de profondeur n'est enregistrée).

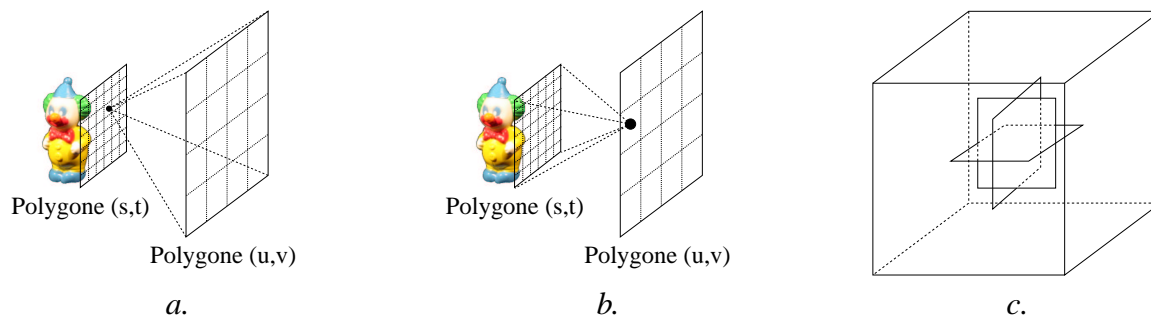


Figure 1.4 – L'échantillonnage de la fonction plénoptique par un slab  $(u, v, s, t)$  représente : a. le flux lumineux passant par un point (une maille du polygone  $(s, t)$ ) vers toutes les directions (traversant le polygone  $(u, v)$ ); b. l'image de l'objet (plaquée sur le polygone  $(s, t)$ ) acquise depuis un point (une maille du polygone  $(u, v)$ ); c. six slabs sont généralement utilisés pour représenter complètement la fonction plénoptique 5D autour d'un volume (boite englobante de l'objet).

Les deux sections suivantes détaillent les particularités de chaque système de représentation : les champs de lumière / les lumigraphes.

### Les champs de lumière

Dans le papier *Light Field Rendering* [LH96], les auteurs préconisent un sur-échantillonnage de la fonction plénoptique afin d'assurer une meilleure qualité de rendu. En effet, ils disposent d'un nombre conséquent de photographies : le maillage des plans  $(u, v)$  entre  $16 \times 16$  et  $64 \times 64$  nécessite une acquisition de 1000 à 16000 images de taille  $256 \times 256$  pour représenter un objet sur 4 slabs (entre 300Mo et 5Go d'espace disque). La résolution du plan  $(s, t)$  est souvent réduite à  $128 \times 128$  lors d'un sur-échantillonnage important ( $64 \times 64$ ) du plan  $(u, v)$ .

Avec un tel échantillonnage, deux prises de vue voisines sur le plan  $(u, v)$  correspondent à une différence angulaire d'environ  $2^\circ$  : les techniques de déformation d'images (section B de l'annexe II) ne sont donc pas nécessaire pour générer une nouvelle vue entre-deux, un filtrage suffit à réduire les artéfacts visuels de l'image produite. Ainsi, le rendu d'un champ de lumière est effectué par interpolation quadrilinéaire entre 4 mailles du plan  $(u, v)$  et 4 mailles du plan  $(s, t)$ , i.e. entre 16 échantillons de luminance (sur les 4 dimensions). Aucune estimation géométrique de la surface de l'objet n'est nécessaire pour revisualiser un objet. Les problèmes de décalage pouvant intervenir entre les différentes vues utilisées sont masqués par le flou dû à l'interpolation. Le recalcul des coordonnées  $(u, v, s, t)$  pour chaque pixel de l'image finale est effectué par placage de textures, les projections perspectives permettent l'obtention d'un rendu interactif.

Les auteurs proposent une méthode de compression des images originales basée sur une quantification vectorielle (facteur 24) et suivie d'un codage entropique (facteur 4). La quantification vectorielle utilise la redondance des informations sur les plans  $(u, v)$  et  $(s, t)$  pour créer un dictionnaire contenant des blocs de 16 échantillons de luminance : 4 mailles du plan  $(u, v) \times 4$  mailles du plan  $(s, t)$ . La compression n'est pas sans perte mais de forts taux peuvent être obtenus : le sur-échantillonnage des champs de lumière augmente la redondance, donc le taux de compression. Cette méthode réduit considérablement les coûts de stockage (facteur de compression : 100) tout en gardant un accès aléatoire à un bloc de données lors du rendu. La dégradation visuelle est en partie masquée par l'interpolation quadrilinéaire effectuée lors du rendu.



## Les lumigraphes

Les auteurs de l'article *The Lumigraph* utilisent un échantillonnage similaire à celui des champs de lumière : le plan  $(s, t)$  est maillé entre  $16 \times 16$  et  $64 \times 64$  ; le plan  $(u, v)$  est maillé entre  $128 \times 128$  et  $512 \times 512$  ; attention, la définition des plans  $(s, t)$  et  $(u, v)$  est inversée par rapport à celle utilisée pour les champs de lumière.

Contrairement aux champs de lumière, où l'acquisition est effectuée slab après slab avec une caméra pilotée par ordinateur se déplaçant sur le plan  $(u, v)$ , l'acquisition de lumigraphes est réalisée à l'aide d'une caméra tenue à la main. Les auteurs utilisent une mire 3D pour pouvoir retrouver les paramètres extrinsèques de la caméra, i.e. sa position et son orientation.

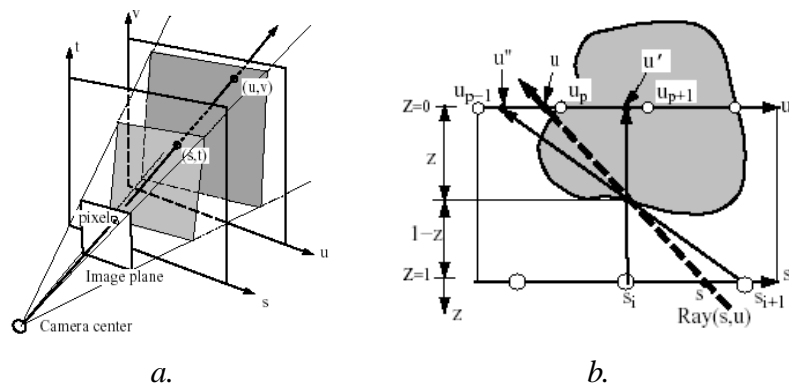


Figure 1.5 – a. Relation entre une image régénérée et le lumigraphe : évaluation des coordonnées d'un rayon traversant l'image (en un pixel), le premier plan du lumigraphe (en un point  $(s, t)$ ) et le second plan (en un point  $(u, v)$ ); b. Utilisation de l'information de profondeur  $z$  pour calculer les échantillons à interpoler sur le plan  $(u, v)$  en fonction des échantillons choisis sur le plan  $(s, t)$  : en 2D, interpolation des échantillons  $(s_i, u')$  et  $(s_{i+1}, u'')$  après correction de la profondeur (images extraites de [GGSC96]).

La principale originalité de cette approche concerne l'utilisation d'un maillage pour évaluer la profondeur lors de la génération de nouvelles vues et ainsi améliorer fortement la qualité du rendu. La silhouette de l'objet vue sur chacune des images constituant le lumigraphe est utilisée pour reconstruire un maillage géométrique rudimentaire de l'enveloppe visuelle de l'objet. Le maillage est obtenu par combinaison d'un algorithme de construction d'octree à partir des silhouettes (développé par R. Szeliski [Sze93]) et de l'algorithme des *marching cubes* [LC87] : un maillage de quelques centaines de triangles est finalement construit (voir section B.2.1 pour plus de détails sur la reconstruction géométrique).

Ainsi, la visualisation de lumigraphes à faible densité d'échantillonnage est possible : l'information de profondeur supprime les effets de flou obtenus précédemment avec les champs de lumière (figure 1.5.b pour des détails). Malgré l'utilisation de la profondeur, le nombre de vues par slab doit toujours être au moins égal à  $16 \times 16$  pour obtenir une visualisation sans artefact.

Pour chaque image générée, la liste des triangles est projetée sur la carte graphique afin d'obtenir l'information de profondeur en chaque pixel (utilisation du tampon de profondeur). Ensuite, cette information de profondeur est utilisée pour modifier la liste des échantillons à interpoler et générer le nouveau point de vue (figure 1.5). Comme pour les champs de lumière, 16 échantillons peuvent être interpolés afin de générer chaque pixel de l'image finale mais 4 échantillons sont généralement utilisés, correspondant aux 4 points de vue les plus proches.

En conclusion, les champs de lumière / lumigraphes sont particulièrement bien adaptés à la représentation d'objets par des images : c'est la structure utilisée dans nos travaux pour stocker les objets. Toutefois, la méthode de visualisation ne permet pas d'intégrer de manière réaliste un champ de lumière dans un environnement. Nous avons comme objectif de rééclairer les objets insérés avec les conditions d'illumination de l'environnement : or le rendu de champ de lumière n'est pas capable de modifier l'éclairage de l'objet, il affiche toujours l'objet avec les conditions d'illumination fixées lors de l'acquisition.

### A.1.5 Les mosaïques concentriques

Les mosaïques concentriques [SH99], développées en 1999 par H-Y. Shum et L-W. He sous l'appellation *concentric mosaics*, est un système de rendu inspiré des panoramas 2D et des lumigraphes. En échantillonnant la fonction plénoptique sur seulement trois dimensions, ce système limite la taille des données acquises (très nettement inférieure à celle des lumigraphes) et permet de visualiser interactivement un environnement depuis un point de vue limité à une petite région circulaire (celle de l'acquisition de données). La limitation sur le point de vue est due à la perte d'une dimension par rapport aux champs de lumière / lumigraphes.

Lors de l'acquisition, une série dense de photographies (un millier) est prise avec une caméra fixée sur le bord d'un plateau tournant et orientée dans la direction tangente au bord du plateau. Les images acquises sur  $360^\circ$  sont fusionnées par petites bandes verticales afin de former une mosaïque concentrique limitant la taille des données réellement stockées (une bande verticale est récupérée sur chaque image). La mosaïque concentrique obtenue ressemble à un panorama 2D (section A.1.2) mis à part que le centre de projection de la mosaïque varie (figure 1.6).

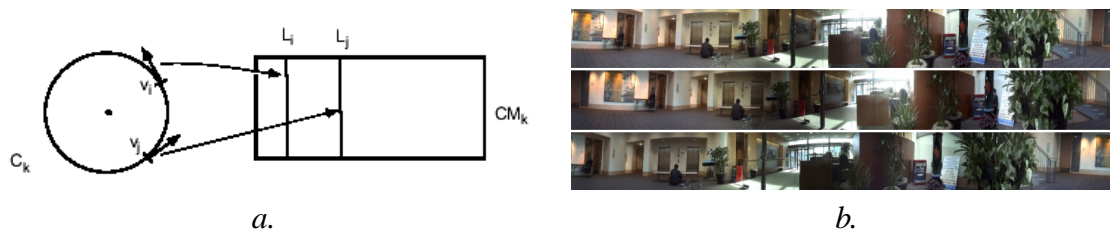


Figure 1.6 – a. Construction d'une mosaïque concentrique à partir de plusieurs photographies. L'acquisition est effectuée par une caméra fixée sur un plateau tournant :  $V_i$  et  $V_j$  représentent deux caméras avec une direction tangentielle au bord du plateau ; chaque prise de vue est stockée dans la mosaïque concentrique  $CM_k$  en position  $L_i$  et  $L_j$  (seulement une bande de pixels est utilisée) ; b. acquisition de trois mosaïques concentriques : la première est acquise avec une caméra placée sur le centre de rotation du plateau tournant ; les autres avec des positions différentes de caméra (images extraites de [SH99]).

Pour obtenir un échantillonnage suffisant de la fonction plénoptique, une vingtaine de mosaïques concentriques sont construites à partir de caméras positionnées à différentes distances du centre de rotation du plateau tournant (figure 1.7.a). En général, une des mosaïques est construite à partir d'une caméra placée au centre de rotation.

Le rendu de nouvelles images peut être effectué pour tout point de vue à l'intérieur du plus grand cercle d'acquisition : pour chaque pixel de l'image à générer, l'algorithme utilise les échantillons adéquats dans la mosaïque concentrique la plus proche du point de vue. Chaque ligne verticale de l'image produite est obtenue par la même mosaïque, minimisant ainsi les calculs d'intersection de rayons avec les mosaïques concentriques (figure 1.7.b pour plus de précisions).

Quelques distorsions verticales apparaissent sur l'image finale mais elles sont corrigées en utilisant une carte de profondeur. Cette carte de profondeur est soit obtenue lors de l'acquisition (exemple des scènes synthétiques), soit approchée manuellement en dessinant le contour des objets les plus proches sur chaque mosaïque concentrique. L'estimation manuelle suffit à réduire les distorsions de manière significative.

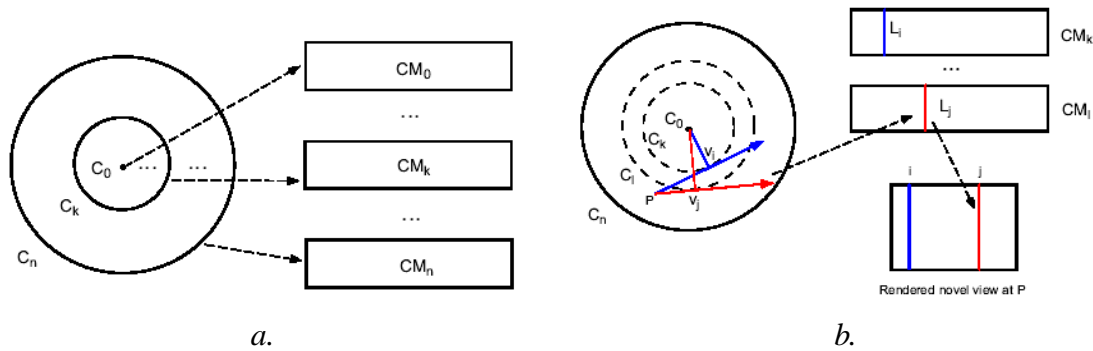


Figure 1.7 – a. Plusieurs images concentriques sont construites, pour des acquisitions effectuées avec différentes positions de la caméra (par rapport au centre de rotation) ; b. rendu d'un nouveau point de vue à partir des mosaïques concentriques : une ligne verticale de l'image est générée par recopie d'une ligne verticale d'une des mosaïques en entrée (images extraites de [SH99]).

Comme chaque point de vue correspond à un panorama 2D avec un centre de projection variant à l'intérieur d'un cercle, ce modèle de rendu ne peut être adapté aux petits objets.

### A.1.6 L'échantillonnage plénoptique

En rendu basé-image, l'analyse de l'échantillonnage plénoptique permet de contrôler la qualité du modèle représenté par les images. En effet beaucoup de systèmes de rendu basé-image, tels que les champs de lumière, sur-échantillonnent la représentation afin de supprimer les artefacts visibles sur les nouvelles images synthétisées (décalages fréquents entre les différents points de vue interpolés). Le sur-échantillonnage implique un grand espace de stockage pour représenter un objet donc plus de redondance sur l'ensemble des informations disponibles.

L'analyse de l'échantillonnage plénoptique est un problème difficile car il implique de juger la précision des informations de profondeur et de texture de l'objet par rapport au nombre d'images de référence disponibles et à la résolution des nouvelles images à synthétiser. Il n'existe pas de relation directe entre ces trois entités, elles sont généralement inter-dépendantes. Dans *plenoptic sampling* [CTCS00], J-X. Chai et al. définissent une relation entre le nombre d'images de référence et la quantité d'informations géométriques et de texture utile à la génération d'une représentation continue de la fonction plénoptique. Les auteurs fournissent un nombre minimum d'images utiles pour le rendu réaliste d'objets représentés par des champs de lumière (i.e. sans aucune information de profondeur) et pour le rendu réaliste d'objets représentés par des luminigraphes (une relation entre le nombre d'images de référence et la précision de l'information géométrique est également établie).

W. Heidrich et al. évaluent la discordance entre chaque paire d'images d'un champ de lumière afin d'ajouter des points de vue aux endroits où l'interpolation entre deux images produit le plus de distorsion et inversement : suppression de points de vue inutiles [HSKS99, SHS99]. Les



champs de lumière obtenus n'ont plus de structure régulière sur le plan  $(u, v)$  mais un nombre restreint d'images est défini et la qualité visuelle est constante lors de la visualisation de nouveaux points de vue. L'ajout d'images supplémentaires dans la structure est réalisé soit par interpolation entre les vues existantes avec utilisation d'informations géométriques (maillage polygonal, cartes de profondeur), soit par synthèse d'images (pour les lumigraphes synthétiques).

Contrairement à W. Heidrich et al., nous préférons utiliser un échantillonnage régulier du champ de lumière afin de conserver l'accès instantané à n'importe quel échantillon  $uvst$  et ajouter un algorithme de compression comme la quantification vectorielle qui permette de réduire la taille des champs de lumière tout en conservant une qualité visuelle constante. C'est le travail proposé dans la section [B](#) du chapitre IV sur la compression de champs de lumière.

## A.2 Exploitation des champs de lumière

Nous présentons dans cette section les principales utilisations des champs de lumière. En effet, de nombreux travaux récents sont orientés sur la réutilisation des champs de lumière afin de développer un modèle de représentation plus complet, intégrant par exemple des données du modèle géométrique ou des conditions d'éclairage dynamiques. Voici les exemples les plus représentatifs de leur utilisation.

### A.2.1 La déformation de champs de lumière

Deux champs de lumière représentant deux objets différents sont utilisés pour transformer de manière dynamique un objet en un autre [\[ZWGS02\]](#). La déformation est guidée par la définition manuelle de rayons de correspondance entre les deux champs de lumière : un rayon 4D du premier objet doit être mis en correspondance avec un rayon 4D du second objet (le rayon ne provient pas nécessairement du même point de vue). Pour faciliter la définition de ces rayons de correspondance, une interface permet de dessiner à main levée un polygone sur une vue du premier objet et le même polygone (déformé mais avec le même nombre d'arêtes) sur une des vues du second objet. Après avoir paramétré un ensemble de couples de polygones, la déformation est effectuée par interpolation entre les deux champs de lumière. Comme toute déformation, plus l'ensemble de polygones est défini précisément, plus la déformation est lisse avec de faibles transitions. Cette méthode est intéressante car elle permet le déplacement de la caméra autour de l'objet pendant la phase de déformation.

### A.2.2 Les champs de lumière paramétrés dynamiquement

Les champs de lumière peuvent également être utilisés pour rendre des objets ou des scènes avec une variation dynamique de la mise au point [\[IMG00\]](#). La profondeur de champ est modifiée interactivement afin que chaque nouvelle image produite semble provenir d'un appareil photographique pour lequel la mise au point peut être modifiée en temps réel. Ce traitement est effectué par une modification de la position et de l'orientation du plan  $(s, t)$  où sont projetées les images d'origine. Seules les zones de l'objet se situant proche du plan  $(s, t)$  sont dans la profondeur de champ donc nettes, les autres parties de l'objet apparaissent floues. Un procédé similaire est exploité dans les lumigraphes [\[GGSC96\]](#) : la géométrie approximative de l'objet permet de corriger la profondeur de champ de chaque pixel afin d'obtenir une image parfaitement nette.

### A.2.3 Les champs de lumière paramétrés sur la surface

Un champ de lumière paramétré sur une surface, appelé *surface light field*, est un champ de lumière dont le plan  $(s, t)$  est positionné sur le modèle géométrique de l'objet à visualiser [MRP98]. Par conséquent, un champ de lumière sur un polygone de la surface représente les informations de texture dépendant du point de vue : lors du rendu, la texture la plus proche du nouveau point de vue à synthétiser est plaquée sur le polygone. Une interpolation bilinéaire suffit pour effectuer le rendu (interpolation sur le plan  $(u, v)$ ) contrairement à l'interpolation quadrilinéaire proposée dans [LH96] et une meilleure qualité de rendu est obtenue. La profondeur de champ est réglée par la géométrie des polygones, elle permet d'obtenir une image entièrement nette.

Cette méthode requiert le stockage de plusieurs champs de lumière afin de représenter la surface complète d'un objet : elle prend plus d'espace mémoire qu'un champ de lumière classique et une compression des données est inévitable. Néanmoins cette structure est particulièrement efficace pour un rendu réaliste interactif, elle sert également à stocker diverses textures par polygones, utiles pour un rééclairage de l'objet (voir section E sur le rééclairage).

### A.2.4 Le placage de champ de lumière

Le placage de champ de lumière, appelé *light field mapping*, est une méthode d'approximation d'un champ de lumière permettant d'accélérer les temps de rendu en utilisant directement une structure de représentation compressée [CBCG02]. La méthode utilise les champs de lumière paramétrés sur la surface avec une factorisation de la fonction 4D représentant chaque champ de lumière de la surface par une somme de produits de fonctions 2D :

$$f_{SLF}(s, t, u, v) \approx \sum_{k=1}^K g_k(s, t) \cdot h_k(u, v) \quad (1.3)$$

avec  $g(s, t)$  une fonction 2D paramétrée par les coordonnées paramétriques de la surface de l'objet ;  $h(u, v)$  une fonction 2D paramétrée par les coordonnées polaires de la direction de vue et  $f_{SLF}(s, t, u, v)$  un champ de lumière paramétré sur une surface. Une petite valeur de  $K$  (généralement entre 1 et 4) permet de réduire significativement la quantité de données par rapport aux champs de lumière paramétrés sur la surface tout en conservant une propriété de réflectance unique en chaque point de la surface et un rendu de haute qualité.

La méthode de rendu est simplifiée par l'utilisation de fonctions  $g$  et  $h$  stockées comme des textures 2D et elle synthétise une nouvelle image par des opérations de plaquage de texture multi-passes. La qualité du rendu peut être améliorée progressivement en augmentant le nombre de passes de rendu. Cette méthode est la référence pour le rendu interactif de champs de lumière mais elle ne permet pas de rééclairer les objets manipulés.

### A.2.5 Les champs de réflectance

Un champ de réflectance (*reflectance field*) est une structure de représentation des données sur 8 dimensions permettant d'acquérir tout transfert lumineux avec la surface d'objet afin de capturer les surfaces spéculaires, anisotropes, etc. Il peut être vu comme le produit du champ de lumière incident à l'objet par son champ de lumière réfléchi. Ainsi la propagation de la lumière le long de la surface, évoquée sur la figure 1.8, peut être acquise avec  $(\theta_i, \phi_i, X_i)$  représentant un

échantillon  $(u_i, v_i, s_i, t_i)$  du champ de lumière incident et  $(\theta_r, \phi_r, X_r)$  représentant un échantillon du champ de lumière réfléchi. Si le champ de réflectance est paramétré sur la surface (comme

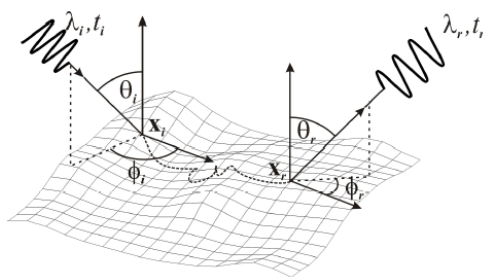


Figure 1.8 – L’interaction lumière-matière n’a pas toujours lieu uniquement à la surface d’un matériau. Une certaine quantité d’énergie lumineuse peut se propager à travers la matière et être réfléchi en un autre point de la surface, différent du point d’incidence de la lumière : ces interactions lumineuses sont représentées par des BSSRDFs (image extraite de [MMS<sup>+</sup>04]).

dans la section A.2.3) il représente alors la BRDF variant sur la surface, appelée BSSRDF (*Bidirectional Scattering-Surface Reflectance Distribution Function*).

Dans [DHT<sup>+</sup>00], P. Debevec et al. se sont intéressés à l’acquisition de champs de réflectance de visages humains : la peau est difficile à représenter par une BRDF classique car une partie non négligeable de la lumière se propage à l’intérieur des tissus et ne peut être rendue que par une BSSRDF. L’acquisition d’un champ de réflectance complet n’est pas envisageable en un temps raisonnable car même pour une seule vue, l’acquisition d’un champ de lumière incident nécessite le déplacement d’un filet de lumière (par exemple un rayon laser) sur toute la surface et sur toutes les directions d’incidence. Une source lumineuse directionnelle est utilisée, placée à 2048 positions différentes tout autour du visage. Par conséquent, le champ de réflectance est réduit à 6 dimensions (seule la direction d’incidence est utile pour paramétrer le champ de lumière incident) et est appelé *champ de réflectance non local*. Le banc d’acquisition motorisé permet l’acquisition d’un visage en une minute à partir de deux caméras fixes enregistrant 30 images à la seconde. Ce système permet l’acquisition en seulement deux points de vues mais l’acquisition de points de vue supplémentaires demanderait un nombre plus important de caméras ou des caméras mobiles.

Grâce à cette méthode, un rééclairage du visage sous de nouvelles conditions d’illumination est rendu possible par combinaison de données extraites du champ de réflectance. Cette méthode est détaillée plus tard, dans la section E consacrée à la comparaison des différentes méthodes de rééclairage d’objets réels.

### A.3 Discussions

Depuis la définition de la fonction plénoptique [AB91], les nouvelles méthodes de rendu basé-image échantillonnent cette fonction afin de pouvoir effectuer le rendu d’une scène ou d’un objet sans utiliser la représentation géométrique. Les méthodes de rendu basé-image les plus utilisées sont certainement les champs de lumières / lumigraphes car la représentation est simple et permet un rendu de scènes et d’objets indépendamment de leurs complexités géométriques et réfléchives. Les temps de synthèse d’une nouvelle image dépendent uniquement de sa résolution.

Malgré un échantillonnage de la fonction plénoptique sur seulement 5 dimensions, les champs de lumières restent volumineux en espace mémoire : suivant la résolution et le nombre d'images, la représentation de chaque objet peut demander plusieurs giga-octets. L'application de méthodes de quantification vectorielle permet toutefois de réduire jusqu'à un facteur 100 la quantité de mémoire nécessaire au stockage des champs de lumière avec un faible coût de décompression, permettant un rendu interactif [LH96].

De nombreux travaux ont porté sur la réutilisation des champs de lumière et notamment en vue d'un rééclairage avec de nouvelles conditions d'illumination. P. Debevec et al. en 2000 proposent les champs de réflectance. Cette structure, simple est basée sur les champs de lumière, mais elle ne permet pas l'acquisition de multiples points de vue car elle nécessite des capacités de stockage gigantesques même pour l'acquisition d'un éclairage directionnel. En effet, en prenant un nombre d'images restreint à  $8 \times 8$  vues par slab et un nombre de sources également restreint à  $8 \times 8$  positions par slab, l'acquisition d'images de résolution  $256 \times 256$  pour construire un champ de réflectance 6D nécessite 27 giga-octets de données. Même après une compression des données, il est difficilement envisageable d'utiliser simultanément plusieurs champs de réflectance lors de l'insertion d'objets dans un environnement. Cette méthode reste tout de même prometteuse car l'augmentation constante des capacités de calcul et de stockage des ordinateurs peut permettre la gestion de champs de réflectance complets sur 8 dimensions et une utilisation pour un rééclairage interactif.

Notre approche du rééclairage d'objets réels est différente des champs de réflectance : plutôt que d'effectuer l'acquisition d'un objet basé-image avec des conditions d'éclairage variées, nous utilisons un champ de lumière, moins coûteux, pour estimer les informations d'illumination et de réflectance de la surface nécessaires au rééclairage. Nous utilisons une représentation compressée pour la visualisation interactive d'objets (développée dans la section B du chapitre IV) et une représentation non compressée pour l'analyse des propriétés de réflectance et des conditions d'illumination afin de travailler avec les luminances réellement acquises. Nous limitons la quantité de données stockées en restreignant le champ de lumière à  $8 \times 8$  vues par slab.

La reconstruction de la géométrie de l'objet paraît par conséquent incontournable. C'est une information indispensable pour pouvoir estimer les conditions d'éclairage et les propriétés de réflectance de l'objet. Elle est également utilisée pour le rééclairage.

## B Reconstruction géométrique

Les diverses techniques utilisées pour obtenir la géométrie d'un objet réel sont fortement dépendantes du matériel d'acquisition. En effet soit la géométrie est acquise par une méthode active, utilisant un matériel spécifique comme par exemple un scanner laser, soit elle est estimée par une méthode passive, à partir d'un ensemble de photographies représentant cet objet.

Dans ce manuscrit, l'état de l'art est axé sur les méthodes automatiques de reconstruction d'objets et n'aborde pas les multiples méthodes spécifiques à la reconstruction d'environnements complexes qui sont généralement semi-automatisées.

### B.1 Acquisition de la géométrie

Le scanner 3D est l'appareil de mesure le plus utilisé pour obtenir une représentation précise de la géométrie d'un objet réel. Ce type de matériel est composé d'un projecteur émettant une

fine bande de lumière sur l'objet et d'une caméra capturant cette bande de lumière sur la surface de l'objet avec une orientation différente du projecteur d'environ  $30^\circ$  (voir figure 1.9). Le système

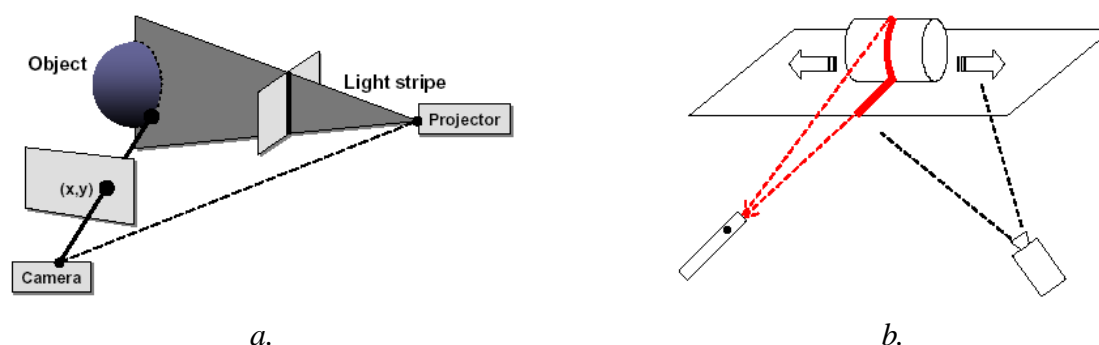


Figure 1.9 – a. Scanner 3D à lumière structurée (image extraite de [RHHL02]); b. scanner 3D laser (image extraite de [DC01]).

doit être parfaitement calibré (connaissance des positions et orientations exactes de la caméra et du projecteur par rapport à l'objet) pour permettre une triangulation optique sur chaque pixel représentant un point éclairé de la surface de l'objet. Cette triangulation détermine les coordonnées 3D de chacun de ces pixels dans un repère global. Une image issue de la caméra est donc transformée en une série de points 3D représentant le contour de l'objet tracé par la bande de lumière. L'objet ou le scanner doit ensuite être déplacé pour effectuer l'acquisition de nouvelles bandes de points et ainsi balayer toute la surface de l'objet. Pour être précise, cette méthode exige un calibrage robuste de la caméra (voir section B.4 pour plus de détails) et l'utilisation d'une bande de lumière la plus fine possible. Les méthodes traditionnelles utilisent une lumière structurée pour projeter une ou plusieurs bandes de lumière sur l'objet (figure 1.9.a, [SWI97, RHHL02]) mais le standard actuel est l'utilisation d'un faisceau laser (figure 1.9.b).

La géométrie finale de l'objet est représentée par un nuage de points 3D. De nombreuses techniques permettent ensuite la construction d'un maillage de la surface (par exemple, la triangulation de Delaunay) et l'estimation de la normale en chaque point à partir de ce nuage de points [She05, MN03]. En 1999, F. Bernardini et al. ont proposé un algorithme de génération de maillage par interpolation d'un ensemble de points non organisés [BMR<sup>+</sup>99].

Les scanners 3D sont également utilisés en translation horizontale devant un objet afin de construire des cartes de profondeur de l'objet stockées dans des images (voir figure 1.10). Chaque image, contenant des valeurs de profondeurs réparties régulièrement, est utilisée pour construire un maillage polygonal d'une partie de l'objet. La fusion de ces maillages produit généralement un maillage complet de l'objet de meilleure qualité que les méthodes par rotation (pour un même taux d'échantillonnage) [CL96].

Généralement, les scanners 3D sont utilisés en complément d'appareils de mesure traditionnels (caméra couleur, appareil photographique). En effet, le scanner récupère les données géométriques de l'objet (forme, normale de la surface), tandis que la caméra couleur récupère plutôt les données photométriques (chrominance, réflectance de la surface). Une mise en correspondance est ensuite nécessaire entre les données provenant de matériels d'acquisition différents. Un scanner laser intégrant les deux types de caméra a été récemment utilisé dans le projet Michelangelo ayant pour but de digitaliser de larges statues atteignant plusieurs mètres de haut [LPC<sup>+</sup>00]. Les principaux inconvénients de ce genre de système sont le prix (pour les scanners laser de grande

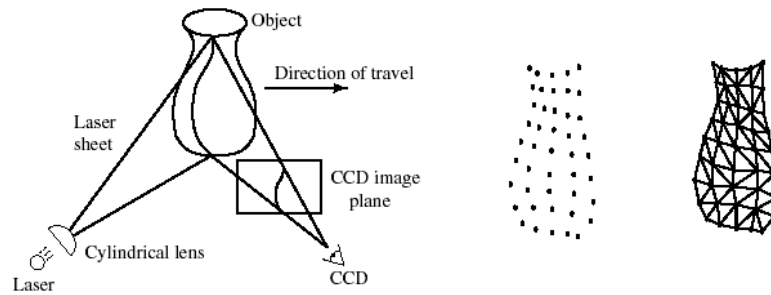


Figure 1.10 – Scanner 3D laser à translation horizontale permettant le stockage du nuage de point sous forme de carte de profondeur (image extraite de [CL96]).

précision) et le montage du banc d'acquisition : le système, qu'il soit motorisé ou non, doit être précis afin d'assurer un calibrage robuste du scanner en mouvement.

## B.2 Estimation de la forme

De nombreuses techniques de reconstruction de la forme par des méthodes passives ont été développées dans le domaine de la vision, s'appuyant uniquement sur une description des objets par une série d'images. Les objets basés-image sont le plus souvent reconstruits en extrayant les informations de silhouette ou de texture des images (approches par *enveloppe visuelle* ou *stéréovision*) mais d'autres techniques utilisent les informations de *photo-cohérence* ou d'*ombrage* présentes dans les images.

### B.2.1 Enveloppe visuelle

A partir d'une série d'images, l'objet peut être représenté par son enveloppe visuelle : la silhouette de l'objet, définie sur chacune des images, est une information permettant de reconstruire la forme de l'objet [MA83, CA86]. Cette méthode de construction, appelée *shape from silhouette*, utilise un ensemble de vues différentes de l'objet à reconstruire, segmente l'objet présent sur chacune de ces vues pour extraire la silhouette, et calcule l'enveloppe visuelle de l'objet par intersection de volumes : à partir de chaque image, un volume conique est construit par projection de la silhouette depuis le centre optique (voir figure 1.11.a). L'*octree* (arbre octal) est la structure de données traditionnellement utilisée pour stocker chaque volume reconstruit. R. Szeliski a repris cette structure en 1993 pour proposer un algorithme plus efficace : il sculpte un seul octree par projections successives des silhouettes (pas de calcul d'intersection de volumes) [Sze93].

Cette approche basée-silhouette (*shape from silhouette*) ne peut reconstruire les cavités présentes à la surface de l'objet. Par exemple, à partir d'un ensemble d'images couvrant toute la surface d'une tasse, l'anse de la tasse est entièrement reconstruite (à condition que le trou à l'intérieur de l'anse soit visible sur au moins une image) mais l'intérieur de la tasse ne peut être reconstruit : les strictes informations sur le contour de l'objet ne permettent pas de détecter la cavité (voir figure 1.11.b).

Cette reconstruction basée-voxel peut être complétée par la production d'un maillage polygonal de la surface en utilisant par exemple l'algorithme des *marching cubes* proposé par W.



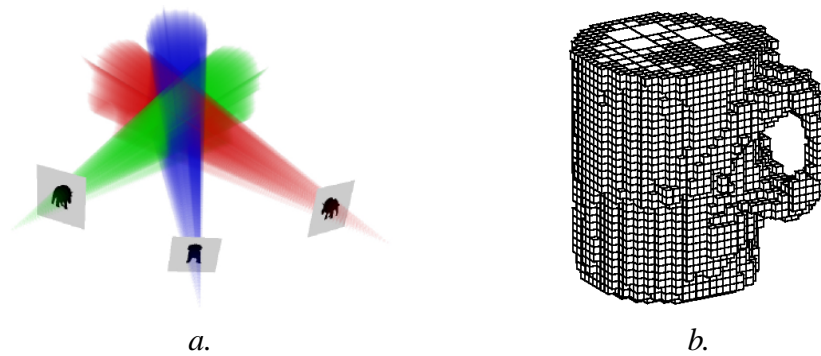


Figure 1.11 – a. Construction de volumes coniques à partir de chaque vue ; b. Reconstruction d'une tasse, l'intérieur de la tasse ne peut pas être reconstruit (image extraite de [Sze90]).

Lorenson et H. Cline en 1987 [LC87] complété par des algorithmes de résolution de ses ambiguïtés [WG90, NH91]. Hasenfratz et al. utilisent la reconstruction basée-silhouette pour intégrer interactivement une personne réelle dans un environnement virtuel à partir de plusieurs caméras vidéo [HLGB03].

En 1995, T. Fromherz et M. Bichsel ont proposé un algorithme de sculpture raffinant la géométrie basée-voxel précédemment obtenue par l'approche *shape from silhouette* [FB95] Cette nouvelle approche permet ainsi de creuser les cavités par comparaison de la luminance émise dans les diverses directions d'observation, jusqu'alors impossible par la méthode de *shape from silhouette*, mais les propriétés de réflectance de la surface doivent être lambertiennes pour assurer un résultat cohérent. En effet tout point d'une surface contenant de la spécularité admet des variations de luminance réfléchié dépendant du point d'observation. Cette méthode est précurseur des techniques de sculpture présentées ci-après.

### B.2.2 Photo-cohérence

Cette méthode permet de supprimer des voxels n'appartenant pas à l'objet par évaluation de la cohérence de la couleur de chaque voxel vu sur plusieurs images : si cette différence de couleur est importante, le voxel concerné n'appartient pas à l'objet et est oté de la surface. Pour accélérer les algorithmes de sculpture, cette méthode utilise une représentation géométrique initiale composée d'une grille régulière dense de voxels englobant l'objet à reconstruire. Introduite en 1997 par S. Seitz et D. Dyer sous le nom de *voxel coloring* [SD97], cette méthode est largement inspirée du raffinement proposé par F. Fromherz et al. : la différence de couleur est évaluée dans l'espace  $(R,G,B)$  en l'interprétant comme une différence de luminance sur chacune de ses composantes chromatiques. Cette approche permet ainsi de retrouver les cavités d'un objet mais elle impose des propriétés de réflectance lambertiennes sur toutes les surfaces de l'objet, restreignant le champ d'application (voir figure 1.12 pour plus de détails sur le fonctionnement de l'algorithme).

Cette approche a ensuite été améliorée sous le nom de *space carving* en 1998 puis 2000 [KS00]. Les principales améliorations concernent l'ordre des tests de visibilité des voxels de la surface et la suppression des contraintes de placement de la caméra (contraintes présentes dans l'approche *voxel coloring*). La méthode a de nouveau été améliorée par W. Culbertson et al.

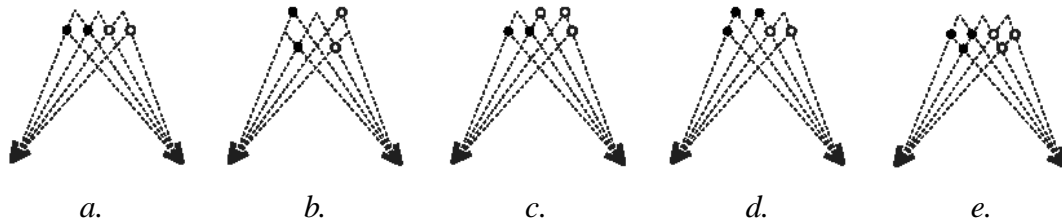


Figure 1.12 – *Reconstruction par voxel coloring* : les voxels sont colorés uniquement si (i) ils sont projetés sur un pixel de même couleur pour les deux images et (ii) si aucun voxel coloré ne les cache dans une image. En fait, à partir de deux images composées de deux pixels noirs suivis de deux pixels blancs, il existe 5 configurations différentes de voxels (les 5 figures) vérifiant l'hypothèse (i). Néanmoins, lorsque l'hypothèse (ii) est ajoutée, seule la configuration e. est retenue : il n'y a donc aucune ambiguïté sur la construction.

[CMS99] : elle s'affranchit des problèmes de calculs de visibilité en utilisant la structure des cartes de profondeur multi-couche présentée en section A de l'annexe II.

### B.2.3 Stéréovision

Les méthodes de stéréovision, souvent utilisées pour une reconstruction précise et robuste, mettent en correspondance des pixels entre plusieurs images afin d'obtenir une estimation de la profondeur sur chacune des images. Cette mise en correspondance est effectuée sur un voisinage de pixels avec des mesures de corrélation croisée entre plusieurs voisinages afin d'obtenir une méthode d'estimation robuste. Cette méthode est très coûteuse en temps de calcul mais l'évaluation de contraintes (sur la géométrie épipolaire établie entre les vues, sur l'ordre, l'unicité et la continuité entre les pixels corrélés) permet de restreindre les zones de recherche de corrélations entre les différentes images. La profondeur des pixels est estimée par un algorithme de triangulation similaire aux méthodes d'acquisition présentées en section B.1 et la géométrie estimée est stockée sous forme de cartes de profondeur ou d'un nuage de points 3D [Fua93]. Un maillage peut être estimé après débruitage du nuage de points.

Le principal inconvénient de cette méthode est la difficulté de détecter les occlusions présentes sur certaines vues de l'objet. Pour restreindre ces occlusions, des vues de l'objet très proches l'une de l'autre sont utilisées. Néanmoins, les vues rapprochées rendent l'algorithme de triangulation moins fiable et obligent une plus grande précision sur la calibration des caméras. Le second inconvénient est la mesure de corrélation sur une fenêtre de pixels qui est efficace uniquement sur les surfaces texturées de l'objet, qui ont un comportement lambertien (réflexion diffuse de la surface pour que la corrélation soit de meilleure qualité). Les éventuels halos spéculaires auraient le même impact sur la corrélation que les occlusions partielles de l'objet.

Depuis quelques années, de nouvelles approches basées-stéréo permettent de reconstruire un modèle géométrique de l'objet très précis en utilisant un modèle déformable. Des contraintes sur la surface de l'objet sont définies par des équations aux dérivées partielles (EDP) et la satisfaction de ces contraintes entraîne une évolution de la surface vers sa forme finale (figure 1.13). Généralement, la surface est initialement représentée par un maillage englobant l'objet [FK98] mais elle peut également être représentée par une grille dense de voxels.



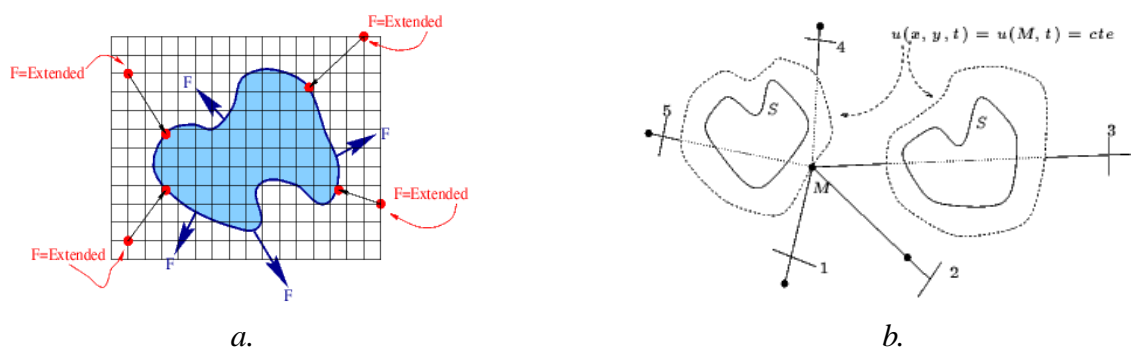


Figure 1.13 – a. Les EDP définissent un champ de forces autour de la surface déformable (image extraite de [Set96]); b. ensemble de niveaux représentant la surface avant et après déformation (image extraite de [FK98]).

## B.2.4 Variations de luminance

Une autre famille de méthodes s’appuie sur la variation progressive de la luminance renvoyée par la surface de l’objet vers l’image pour retrouver des informations géométriques. Dans la littérature, ces méthodes sont classées sous l’appellation *shape from shading*. Les surfaces de l’objet sont supposées lambertiennes (parfaitement diffuses) pour réduire le nombre de paramètres nécessaires à la représentation de l’équation d’illumination directe des surfaces. Ainsi, chaque luminance extraite d’un pixel ne dépend que de la direction de la source lumineuse et de l’orientation de la normale à la surface de l’objet (voir l’équation de luminance définie en section D de l’annexe I). La connaissance a priori de la source lumineuse permet ainsi de déduire la normale à la surface en chaque pixel. L’évaluation de la normale est ensuite utilisée pour évaluer la pente de la surface en chaque pixel et reconstruire un maillage de la surface visible.

Parmi les différentes méthodes développées, il existe les approches par minimisation [LK93], les approches par propagation [BP92], les approches locales [LR85] et les approches globales [Pen88].

Une étude comparative des différentes approches est proposée par R. Zhang et al. où les algorithmes référencés ci-dessus sont testés sur une batterie commune d’images [ZTCS99]. Quelle que soit la méthode utilisée, de fortes hypothèses sont nécessaires sur les conditions d’éclairage pour obtenir une estimation correcte de la surface de l’objet : une seule source lumineuse infinie, de direction connue et une surface parfaitement diffuse, cela restreint le champ d’application de ces méthodes.

## B.3 Estimation de la normale

La reconstruction de la forme n’estime généralement pas l’information de normale à la surface de l’objet reconstruit, excepté pour les méthodes basées sur les variations de luminance. Or nous avons besoin de connaître précisément la normale à la surface de l’objet reconstruit pour estimer les conditions d’illumination et les propriétés de réflectance de la surface. Elle est également utile pour rééclairer l’objet lors de son intégration dans un environnement.

Il existe deux grandes familles d’estimation de la normale à partir d’une représentation de la forme de la surface : celle s’appuyant sur une description de la surface par un maillage composé de triangles et celle utilisant une description discrète de la surface par des voxels. Nous verrons

qu'il est également possible d'évaluer la normale de la surface lambertienne sans construction explicite de la géométrie.

### B.3.1 Méthodes d'interpolation

Lorsque le maillage est disponible, il est possible d'évaluer une normale à la surface de l'objet par une méthode d'interpolation. La plus populaire est l'interpolation de Phong : elle consiste à évaluer une normale en chacun des sommets des triangles constituant le maillage puis à calculer la normale de tout point de la surface par interpolation entre les normales des trois sommets du triangle considéré. La normale en chaque sommet est généralement évaluée par une moyenne des normales des triangles incidents à ce sommet.

Par la suite, d'autres méthodes ont été développées pour améliorer l'estimation de la normale en chaque sommet. G. Thürmer et al. ont proposé de pondérer la moyenne des normales de tous les triangles incidents à un sommet par l'angle formé entre les deux arêtes de chaque triangle incidentes au sommet [TW98]. La pondération peut également se faire par l'aire de chaque triangle incident au sommet.

D'autres approches estiment, en chaque sommet de triangle, le gradient (dérivée partielle) d'une fonction d'approximation de la surface sur un voisinage soit par régression linéaire, soit par méthode de différence finie [MEM01]. Ces méthodes permettent d'obtenir une estimation plus précise de la normale à condition d'avoir suffisamment d'informations géométriques pour guider l'approximation locale de la surface.

### B.3.2 Evaluation d'un plan tangent

A partir d'une représentation discrète de la surface, la normale peut être estimée en chaque voxel. L. Papier et al. proposent une méthode simple basée sur une moyenne pondérée des normales *triviales* sur un voisinage [PF98]. Une normale triviale correspond à la normale d'une face visible de voxel (figure 1.14.a). En supposant que la surface discrète sépare localement l'espace

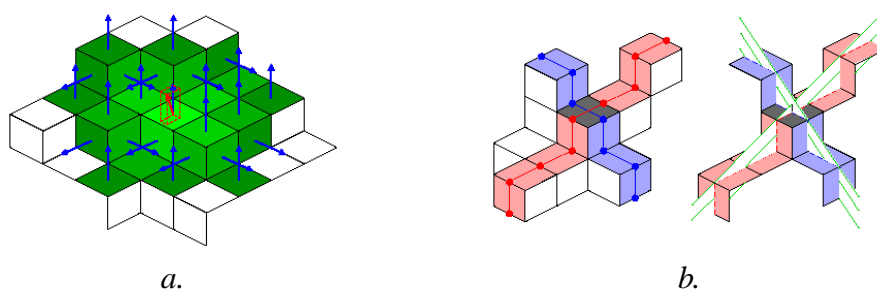


Figure 1.14 – Estimation de la normale d'un voxel ; a. par moyennage des normales triviales des faces de voxels sur un voisinage de faces 4-connexes (sur cet exemple, le voisinage est de distance 3) ; b. par évaluation de deux segments de droites discrètes tangentes au voxel.

en deux sous-ensembles disjoints de voxels, G. Thürmer et al. proposent d'utiliser l'un de ces deux sous-ensembles sur un voisinage autour d'un voxel afin d'évaluer la normale [TW97]. Ces méthodes sont efficaces sur des surfaces continues  $C^1$  mais posent des problèmes pour les voxels représentant des arêtes de la surface (lorsque la surface reste continue  $C^0$ ).

Une évaluation de plan tangent local à chaque voxel permet de s'affranchir des problèmes de discontinuité de la surface. Cette approche évalue un segment de droite 2D (tangent au voxel) à partir des faces 4-connexes de la surface discrète. L'évaluation d'un segment de droite pour deux axes orthogonaux définit un plan tangent au voxel (figure 1.14.b). Le vecteur normal au plan représente ainsi une estimation de la normale locale au voxel [TDR99, LV03].

### B.3.3 Evaluation de la luminance

Nous avons vu à la section B.2.4 que les approches de type *shape from shading* permettent de reconstruire la surface d'un objet par évaluation de la normale en chaque pixel d'une image. L'estimation de la normale en chaque pixel utilise l'information de luminance captée par le pixel pour évaluer l'équation de luminance (section D de l'annexe I) selon le modèle de réflectance lambertien. Par conséquent la surface doit être lambertienne et éclairée par une seule source lumineuse de direction connue. Nous nous étendons pas plus sur ce type de méthodes car elles ne sont pas adaptées à nos besoins : nous disposons de plusieurs images et non d'une seule, nos objets sont éclairés par de multiples sources lumineuses et leur surface n'est généralement pas lambertienne.

## B.4 Calibrage de la caméra

Toutes les méthodes d'acquisition et d'estimation présentées requièrent une évaluation précise des coordonnées spatiales du matériel d'acquisition (par exemple la position et l'orientation de la caméra) afin d'estimer la géométrie de l'objet réel. En effet, sans une connaissance précise des paramètres intrinsèques (longueur focale, coefficient de distorsion, etc) et extrinsèques (position, orientation, etc) de la caméra, il est extrêmement difficile d'obtenir une estimation précise de la géométrie de l'objet acquis.

### B.4.1 Utilisation de matrices de conversion

R. Tsai, travaillant dans le domaine de la robotique, représente les paramètres d'une caméra sous forme matricielle afin de pouvoir passer d'un système de coordonnées locales (coordonnées 2D des pixels d'une image) à un système de coordonnées globales (coordonnées 3D d'un point de la surface de l'objet) par une opération matricielle [Tsa87]. Ce système matriciel intègre la plupart des caractéristiques d'une caméra (6 paramètres extrinsèques et 6 paramètres intrinsèques) : il permet notamment de prendre en considération la déformation sur les bords de l'image due à la lentille de la caméra. C'est la représentation de référence, elle est efficace et très souvent utilisée en vision.

### B.4.2 Utilisation d'un objet de référence

Les systèmes d'acquisition composés de caméras motorisées effectuent généralement l'acquisition d'un objet réel en deux étapes. Dans la première étape, un objet simple dont les caractéristiques géométriques sont bien connues (exemple, un cube de dimension connue) est acquis afin d'extraire les caractéristiques de la caméra (position, orientation, déformation de l'image sur les bords, etc). Dans la seconde étape, l'objet réel est acquis avec les mêmes prises de vues que précédemment. L'objet réel peut donc être reconstruit en utilisant l'estimation préalable des

paramètres de la caméra. I. Sato et al. utilisent cette technique pour l'acquisition de la géométrie et des propriétés de réflectance de leurs objets [SSI99a].

### B.4.3 Utilisation d'une mire

D'autres méthodes utilisent une mire présente lors de l'acquisition des objets pour faciliter la récupération de certains paramètres de la caméra (généralement la position, la direction et l'orientation).

S. Gortler et al. utilisent une mire 3D pour la construction des lumigraphes [GGSC96]. Ils positionnent leur objet devant une mire composée de 3 panneaux orthogonaux représentant les 3 plans de leur repère global et de points/cercles de différentes tailles/couleurs pour différencier chacun des plans (figure 1.15). La mire est partiellement cachée par l'objet à photographier mais

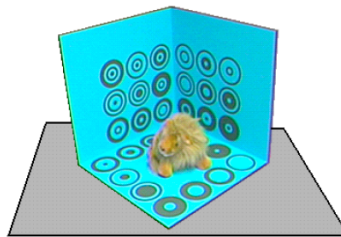


Figure 1.15 – Mire 3D utilisée lors de l'acquisition de lumigraphes (image extraite de [GGSC96]).

l'évaluation de la distorsion des cercles sur chaque image améliore la précision de l'estimation des paramètres de la caméra.

E. Guillou a également utilisé une mire pour retrouver les paramètres de la caméra à partir d'une image [Gui00]. Cependant il utilise une mire 2D (par exemple une feuille de papier) composée d'un ensemble régulier de droites orthogonales (représentant 2 axes du repère global). Le 3<sup>e</sup> axe du repère n'est pas représenté sur la mire, comme dans les travaux de W. Chen [CJ91]. Après la détection de l'ensemble des droites visibles sur la photographie et l'évaluation d'un point de fuite sur chacun des 2 axes du repère, l'utilisation de la géométrie épipolaire permet l'estimation des coordonnées 3D de la caméra dans le repère global.

## B.5 Discussions

La *triangulation* laser ou par lumière structurée nécessite un très fort calibrage de la caméra et du projecteur pour assurer une reconstruction précise. Un système d'acquisition manuel est long et difficile à calibrer tandis qu'un système motorisé est onéreux.

L'approche de type *shape from silhouette* est la seule méthode permettant d'effectuer une reconstruction sans se soucier des propriétés de réflectance de la surface des objets à reconstruire. En revanche, elle ne permet pas de reconstruire les cavités contrairement aux autres méthodes.

L'approche par *photo-cohérence* permet de sculpter l'objet sans avoir à extraire la silhouette de l'objet sur les images. Aucune segmentation n'est donc requise mais la surface doit avoir des propriétés lambertiennes pour que la cohérence des couleurs entre les images soit efficace.

L'approche par *stéréovision* fournit des méthodes très robustes de reconstruction de la géométrie de l'objet mais nécessite des surfaces fortement texturées et diffuses. De plus, l'auto-

occultation de parties de l'objet sur certaines vues entraîne des problèmes de fiabilité de la reconstruction : il est difficile de détecter les occlusions avec une méthode travaillant dans l'espace-image, même si de récents travaux permettent de détecter plus précisément ces occultations.

L'approche *shape from shading* nécessite des surfaces parfaitement diffuses et la connaissance de la direction de la source lumineuse éclairant l'objet pour fournir une reconstruction correcte. C'est la méthode la plus contraignante de toutes.

Malgré la distinction de quatre grandes familles de reconstruction d'objets à partir d'images, il existe beaucoup de travaux combinant plusieurs de ces méthodes afin d'améliorer le résultat final de la reconstruction. Dans [YSCS02], A. Yezzi et al. proposent d'utiliser le volume de photo-cohérence pour guider leur modèle déformable. C. Esteban et F. Schmitt utilisent plutôt l'enveloppe visuelle pour initialiser leur modèle déformable [ES03]. P. Eisert et al. évaluent la photo-cohérence sur chaque voxel de la surface issu d'une reconstruction par enveloppe visuelle afin de creuser les éventuelles cavités de l'objet [ESG99]. Y. Matsumoto et al. s'appuient également sur une reconstruction à partir de la silhouette pour réduire la complexité des algorithmes de corrélation de la stéréovision [MFK99, CZ00]. Enfin, P. Fua et Y.G. Leclerc utilisent les informations de texture et d'ombrage pour reconstruire simultanément des objets texturés ou diffus [FL95].

Même si ces combinaisons produisent des résultats intéressants, elles ne permettent pas de s'affranchir des hypothèses restrictives des grandes familles de méthodes. A titre d'exemple, la combinaison d'une approche *shape from silhouette* avec une méthode de *stéréovision* ne permet pas de reconstruire des objets sans texture. Elle permet simplement d'avoir la qualité des méthodes traditionnelles de *stéréovision* avec une forte optimisation des temps de reconstruction grâce à l'utilisation de l'enveloppe visuelle de l'objet. Elle permet également de régler certains problèmes liés aux occultations.

En règle générale, la combinaison des méthodes est surtout intéressante pour faire converger la forme reconstruite plus rapidement vers sa forme finale au détriment de l'espace-mémoire utilisé (les méthodes combinant l'enveloppe visuelle avec une autre technique de reconstruction utilisent généralement une grille dense de voxels pour fusionner plus facilement les informations reconstruites).

Par conséquent, nous utilisons une approche par enveloppe visuelle afin de garder la liberté d'utiliser des objets de réflectance variée et de n'imposer aucune restriction sur les conditions d'illumination utilisées. Nous avons remarqué l'importance du calibrage de la caméra pour espérer obtenir une reconstruction géométrique précise d'un objet basé-image. Dans le chapitre II, nous proposons une évaluation de la fiabilité de notre système d'acquisition afin d'assurer la construction d'un modèle géométrique précis.

La combinaison des informations provenant du modèle géométrique et du modèle basé-image est intéressante car elle permet de guider la détection des sources lumineuses et l'estimation des propriétés de réflectance.

## C Détection de sources lumineuses

La détection de sources lumineuses à partir d'images est un domaine de recherche récent, peu étudié comparé aux méthodes de reconstruction 3D : cela est certainement dû à la complexité de la distribution d'illumination d'une scène réelle. En effet, les valeurs de luminance reçues en chaque pixel d'une image dépendent essentiellement :

- de la géométrie de la scène,

- des propriétés de réflectance des matériaux,
- et des conditions d'éclairage.

En synthèse d'images, ces trois informations sont utilisées simultanément dans l'équation de luminance afin de générer de nouvelles vues d'une scène pour laquelle toutes les informations sont disponibles (section D de l'annexe I).

L'analyse des valeurs de luminance d'une image permet de retrouver une information manquante à partir des deux autres en réévaluant l'équation de luminance sur chaque pixel. Par exemple, la connaissance de la géométrie d'un objet et de ses paramètres de réflectance permet l'estimation des paramètres d'illumination à partir d'une image. Néanmoins, les scènes réelles sont souvent constituées d'illumination directe et indirecte, rendant difficilement réalisable l'analyse précise de l'ensemble des sources lumineuses.

## C.1 Les types de sources lumineuses

Les conditions d'éclairage des scènes virtuelles sont modélisées par un nombre restreint de sources lumineuses de types différents tout en reproduisant le plus fidèlement possible l'illumination des environnements réels. La figure 1.16 montre un exemple des différents types utilisés en synthèse d'images.

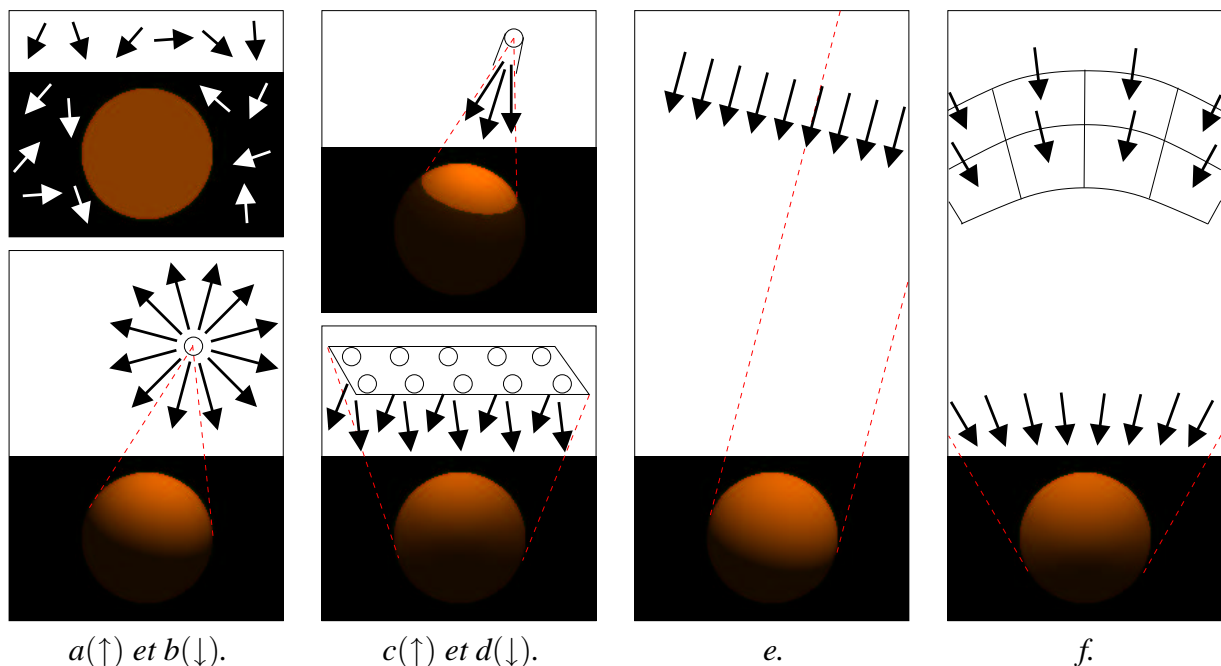


Figure 1.16 – *a*. Interréflexions lumineuses (terme *ambient*) ; *b*. source ponctuelle ; *c*. spot de lumière ; *d*. source surfacique ; *e*. source directionnelle ; *f*. source hémisphérique.

### Source ponctuelle

La source ponctuelle est utilisée pour représenter par exemple une ampoule. Elle émet une luminance constante dans toute direction à partir de son point d'émission (figure 1.16.b). Même si la luminance est constante, la quantité d'énergie atteignant une surface diminue en fonction de

la distance séparant la source de la surface : l'angle solide formé par la surface depuis la source doit être pris en considération.

### **Spot de lumière**

Le spot de lumière est utilisé pour représenter par exemple un projecteur. Il n'émet pas toujours une luminance constante dans toute direction à partir de son cône d'émission. En effet, certains spots sont représentés par une illumination plus faible sur la périphérie du cône (figure [1.16.c](#)).

### **Source surfacique**

La source surfacique correspond à un éclairage depuis une surface d'émission vers une hémisphère. Elle est utilisée pour représenter par exemple des tubes fluorescents au néon. Cette source émet souvent une distribution d'illumination non uniforme : la luminance émise peut varier en fonction de la position et de la direction d'émission. Néanmoins les sources surfaciques sont souvent considérées comme des surfaces émettant une luminance constante ou échantillonnées par une grille régulière de sources ponctuelles identiques entre elles (la figure [1.16.d](#) représente une source surfacique échantillonnée par 10 sources ponctuelles).

### **Source directionnelle**

La source directionnelle correspond à un éclairage unidirectionnel dans une scène. Elle est souvent utilisée pour représenter une source ponctuelle de distance infinie : par exemple le soleil (figure [1.16.e](#)).

### **Source hémisphérique**

La source hémisphérique correspond à un éclairage multi-directionnel dans une scène. Elle est utilisée pour représenter une carte d'illumination naturelle d'un objet assimilé à un point (le centre de l'hémisphère). La figure [1.16.f](#) montre un exemple de source hémisphérique représentée par 8 sources directionnelles.

### **Interréflexions lumineuses**

Un terme ambiant est souvent utilisé en synthèse d'images pour compenser les interréflexions lumineuses non évaluées par une méthode d'illumination locale. Il ne correspond à aucune source physique du monde réel mais peut être assimilé à un éclairage omnidirectionnel dans la scène et de luminance constante quelle que soit la direction (figure [1.16.a](#)).

Plusieurs types de sources lumineuses peuvent être détectés à partir d'images mais les méthodes estiment généralement des sources de type ponctuel ou directionnel. Deux approches différentes permettent de détecter les conditions d'illumination. La première est basée sur l'analyse d'une seule image où les surfaces sont lambertiennes et la géométrie de la scène est connue : cette approche est restreinte à des environnements éclairés par une seule source lumineuse directionnelle. La seconde utilise les informations spéculaires visibles sur une sphère calibrée insérée lors de l'acquisition afin de faciliter la détection de sources lumineuses multiples.



## C.2 Utilisation d'une seule image

### C.2.1 Détection d'une source directionnelle

En 1982, A. Pentland s'intéresse pour la première fois au problème de la détection d'une source lumineuse non directement visible sur une image [Pen82]. Il propose d'estimer la direction d'une source lumineuse à partir d'une seule image et sans la connaissance à priori de la géométrie de la scène. Pour cela, l'environnement doit être éclairé par une seule source lumineuse directionnelle et composé de surfaces lambertiennes. La méthode fait l'hypothèse que les surfaces peuvent être approchées localement par une sphère i.e. sont de continuité  $C1$  (les scènes constituées de surfaces planes avec des arêtes vives ne sont pas adaptées à cette méthode). La direction d'incidence est choisie par évaluation de maximum de vraisemblance et par minimisation d'erreur évaluées sur des régions spécifiques de l'image.

En 1994, Chojnacki et al. apportent quelques améliorations en modifiant la formulation d'un des angles du système de coordonnées polaires représentant la direction d'incidence estimée [CBG94] et en restreignant les informations de luminance aux centres des objets (les bords des objets émettent une luminance peu fiable).

En 1993, P. Sinha et E. Adelson ont proposé une nouvelle approche basée sur des surfaces localement planes à réflectance diffuse [SA93]. La scène est modélisée manuellement et les paires de polygones adjacents dont l'arête commune représente un changement d'illumination entre les deux faces sont utilisées pour restreindre l'intervalle des directions potentielles de la source lumineuse (figure 1.17). L'intersection des directions d'incidence potentielles sur plusieurs paires

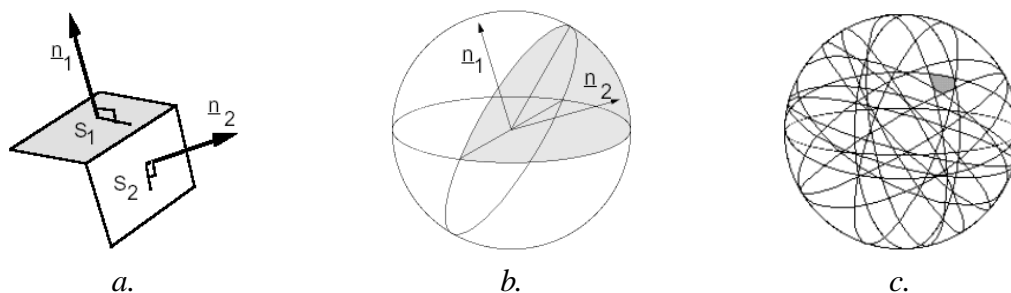


Figure 1.17 – a.  $S_1$  et  $S_2$  sont deux surfaces éclairées telles que la direction d'incidence est plus proche de  $\vec{n}_2$  que de  $\vec{n}_1$  (car la luminance émise par  $S_2$  est plus importante qu'en  $S_1$ ); b. ensemble de directions d'incidence potentiellement valides (intersection de l'hémisphère centrée en  $\vec{n}_1$  par l'hémisphère contenant  $\vec{n}_2$  et dont la base passe par l'arête entre  $S_1$  et  $S_2$  et par le vecteur bissecteur entre  $\vec{n}_1$  et  $\vec{n}_2$ ); c. ensemble de directions d'incidence potentiellement valides après l'utilisation de plusieurs paires de surfaces (images extraites de [SA93]).

de polygones permet de déduire un petit ensemble de directions représentant la direction de la source lumineuse estimée (figure 1.17.c). Cette méthode ne fonctionne que pour des scènes éclairées par une seule source lumineuse directionnelle, sans ombre projetée mais les surfaces peuvent être non parfaitement diffuses.

En 1985, M. Brooks et K. Horn proposent une méthode d'estimation conjointe de la direction de la source et de la géométrie de la scène par calcul variationnel. Une carte de réflectance est calculée afin de représenter l'énergie réfléchiée par la surface lambertienne en fonction de l'énergie reçue pour toutes valeurs de la normale [BH85]. Cette carte de réflectance est modifiée



itérativement jusqu'à trouver une orientation de normale en chaque pixel de l'image qui permette de minimiser l'erreur entre l'image d'origine et l'image recalculée.

En 1999, D. Samaras et D. Metaxas utilisent un modèle géométrique déformable pour retrouver la direction de la source lumineuse et une estimation précise de la surface d'un objet [SM99]. Le modèle déformable est représenté par une surface paramétrique et une contrainte d'illumination de la surface lambertienne est intégrée à l'équation de mouvement de la surface afin qu'un processus itératif évalue successivement les paramètres du modèle déformable et la direction de la source lumineuse jusqu'à convergence. La figure 1.18 montre un exemple de modèle reconstruit. Le modèle obtenu est fiable pour des images synthétiques, même à partir d'une mauvaise

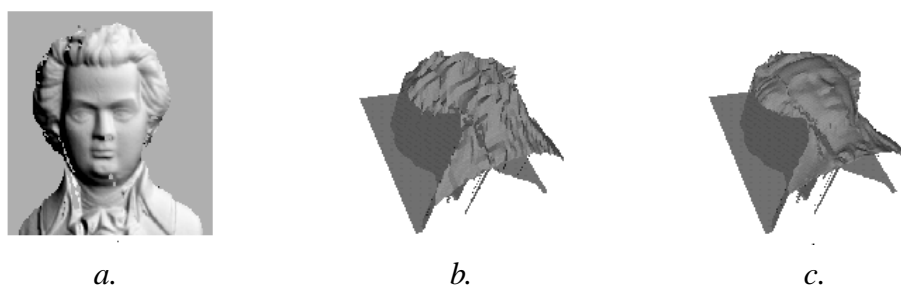


Figure 1.18 – *a. Image originale ; b. modèle reconstruit sans réestimation de la direction de la source lumineuse ; c. modèle final obtenu après plusieurs itérations (images extraites de [SM99]).*

initialisation de la direction de la source lumineuse (l'estimation de moindres carrés non linéaire utilisée est dérivée de la méthode de Levenberg-Marquart [PTVF99]).

Tous ces travaux sont en étroite liaison avec l'approche *shape from shading* développée en section B.2.4 permettant la reconstruction de la géométrie de la scène à partir d'une seule image.

### C.2.2 Détection de plusieurs sources directionnelles

Ces travaux développés par Y. Yang et A. Yuille en 1991 sont basés sur l'extraction des contours des objets occultant une partie de la scène [YY91]. Chaque contour occultant est potentiellement éclairé par la source lumineuse et tous les pixels de ces contours sont des points 3D dont la normale est supposée orthogonale à la direction du point de vue. Une évaluation de la tangente au contour, en chacun de ces points, fournit une estimation précise des coordonnées du vecteur normal en chaque pixel du contour occultant (figure 1.19.a). En faisant l'hypothèse que toutes les surfaces sont lambertiennes, les informations de luminance sont utilisées en chaque pixel du contour afin de déterminer la direction de la source lumineuse éclairant le contour. Ce processus est réitéré sur chaque contour occultant afin d'obtenir une estimation fiable de la direction d'incidence. Si les directions sont significativement différentes (angle supérieur à  $90^\circ$ ), elles peuvent être groupées afin de détecter plusieurs sources lumineuses directionnelles. Plusieurs images de différents points de vue de la scène peuvent être utilisées pour améliorer la précision des résultats.

En 2001, P. Nillius et J. Eklundh ont proposé d'approcher la surface autour du pixel de plus forte luminance d'un contour occultant par une ellipse (figures 1.19.b et 1.19.c). Cette ellipse permet d'évaluer la normale sur un voisinage de pixels et d'estimer la luminance émise par la source lumineuse [NE01] (les valeurs de luminance trop proches du contour occultant ne sont

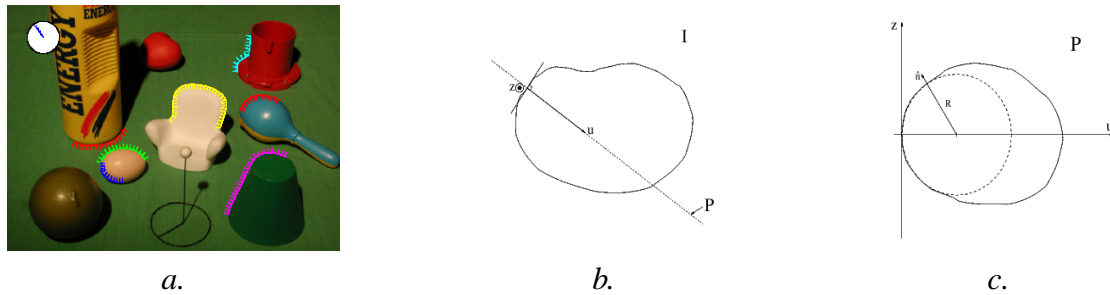


Figure 1.19 – a. Normales définies sur les contours occultants ; b. contour d'un objet extrait de l'image  $I$ ;  $P$  est le plan passant par le centre optique de la caméra, par le pixel de luminance maximale situé sur le contour, et est défini orthogonal à la tangente au pixel de contour ; c. vue de profil, le plan  $P$  est maintenant confondu avec la feuille ; le bord de la surface de l'objet est approché par une ellipse, ce qui donne une estimation de la normale en chaque point de la surface, à proximité du pixel de contour (images extraites de [NE01]).

pas prises en compte car le modèle lambertien n'est pas valide pour des angles complètement rasants).

### C.2.3 Détection de sources surfaciques

Ces approches permettent d'estimer une source lumineuse de type surfacique en utilisant une méthode de radiosité sur une scène représentée par une seule image. Ce sont souvent des approches utilisées en *réalité augmentée* pour insérer un objet virtuel dans un environnement réel avec prise en compte de l'illumination de la scène sur l'objet. Les conditions d'éclairage de l'environnement réel sont extraites de l'image par calcul des transferts d'énergie à l'intérieur de la scène : celle-ci doit donc au préalable être modélisée. Généralement, la reconstruction est faite de façon semi-automatique en traçant les lignes de fuites présentes dans la scène.

A. Fournier propose en 1994 une méthode pour estimer les paramètres des sources lumineuses par des calculs de simulation d'éclairage [Fou94]. La scène est modélisée grossièrement par une centaine de parallélépipèdes représentant des surfaces lambertiennes et chaque pixel est projeté sur la scène comme un élément de surface diffuse. Chaque élément de surface de la scène est considéré comme une source lumineuse avec une émission égale à sa radiosité (déduite des pixels).

La même année, A. Gagalowicz propose une méthode différente permettant l'estimation simultanée de la géométrie de la scène et des paramètres des sources lumineuses par des étapes d'analyse puis de synthèse d'images [Gag94]. Une image de la scène est produite par un algorithme de radiosité puis est comparée à l'image réelle de la scène : les réflectivités des objets de la scène sont modifiés, puis le processus est réitéré jusqu'à l'obtention d'une image proche de l'image réelle.

### C.2.4 Détection de sources hémisphériques

En 1993, D. Hougen et N. Ahuja proposent d'estimer de multiples sources lumineuses en discrétisant la distribution d'illumination sur une hémisphère englobant l'objet [HA93]. Ils partitionnent l'hémisphère par un ensemble de 17 régions d'angle solide équivalent et représentent

la luminance reçue en chaque pixel d'une image par la somme de la luminance reçue depuis chacune des 17 sources en utilisant un modèle de réflectance lambertienne. Cette approche utilise la stéréovision sur deux images pour retrouver la géométrie de la scène et résout un ensemble d'équations linéaires sur la luminance des pixels. Seules les sources lumineuses permettant de minimiser l'erreur entre les luminances des pixels et les luminances calculées sont conservées. Cette méthode fait l'hypothèse qu'aucune ombre n'est visible sur les deux images.

En 1999, Sato et al. ont repris l'idée d'un échantillonnage discret de la distribution d'illumination sur une hémisphère afin d'estimer les sources lumineuses éclairant un objet de géométrie connue posé sur une table [SSI99b]. La luminance reçue par les pixels de l'ombre projetée par l'objet sur la table est évaluée en fonction de l'ensemble des sources présentes sur l'hémisphère. L'ombre projetée n'avait jamais été utilisée auparavant pour retrouver des informations d'illumination dans une scène. La géométrie du contour occultant et de l'ombre projetée permet d'avoir une première estimation des directions des sources lumineuses. Cette première estimation est réutilisée dans [SSI01] pour sur-échantillonner l'hémisphère dans des directions d'illumination privilégiées, notamment pour les zones de pénombre apparaissant dans l'image. L'ensemble des sources lumineuses est estimé soit pour des surfaces non-lambertiennes de réflectance connue (le modèle simplifié de Torrance-Sparrow [TS67] est utilisé avec une mesure des coefficients de réflexion spéculaire et de réflexion diffuse), soit pour des surfaces lambertiennes de réflectance inconnue (une seconde image est utilisée où l'objet aux contours occultants est enlevé). Ils définissent généralement la distribution d'illumination par une centaine de sources lumineuses directionnelles.

### C.2.5 Détection de sources ponctuelles

E. Guillou est le seul à proposer une méthode de détection de multiples sources lumineuses de type ponctuel ou directionnel à partir d'une seule image [Gui00]. Pour faciliter l'estimation, l'environnement doit être composé de surfaces lambertiennes et les cartes de profondeur et de réflectance sont nécessaires. La méthode fait l'hypothèse qu'une source ponctuelle peut être considérée comme directionnelle si elle est utilisée dans l'évaluation de la luminance sur une petite région de la scène. Par conséquent, une source lumineuse directionnelle est estimée par région puis la position d'une source ponctuelle est évaluée comme l'intersection des directions des sources précédemment estimées. L'imprécision sur la position d'une source lumineuse éclairant une pièce de 3m de large est d'environ 50cm. Pour évaluer plusieurs sources ponctuelles, l'algorithme utilise l'ensemble des sources directionnelles évaluées sur un maximum de régions de la scène (généralement définies manuellement pour un maximum de cohérence), puis regroupe les directions d'incidence en fonction d'un critère d'erreur évaluant la pertinence des regroupements éventuels. L'imprécision sur la position des sources lumineuses augmente avec le nombre de sources détectées : elle dépasse 1 mètre lorsque deux sources sont estimées.

## C.3 Utilisation d'une ou plusieurs sphères

### C.3.1 Estimation d'une carte d'environnement

La détection de sources lumineuses étant un problème difficile à partir des seules informations contenues dans une image, P. Debevec propose en 1998 d'utiliser une sphère miroir afin de capter l'éclairage arrivant sur cette sphère : la construction d'une carte d'environnement per-

met l'insertion réaliste d'un objet synthétique dans une scène réelle (objectif premier de la *réalité augmentée*) [Deb98]. La sphère miroir est placée dans la scène près de l'endroit d'insertion de l'objet synthétique et plusieurs photographies de la scène sont prises en faisant varier la vitesse d'obturation de la caméra (entre 1/4 et 1/10000 de seconde). Ainsi, une carte de luminance de la sphère peut être reconstruite avec une forte variation de ses valeurs [DM97] (voir section A de l'annexe II pour plus de détails). Cette méthode souffre de l'apparition de la caméra sur la sphère miroir mais l'utilisation de deux photographies de la sphère, à  $90^\circ$ , permet de supprimer cet artefact.

### C.3.2 Détection de plusieurs sources directionnelles

Depuis, de nombreuses méthodes d'estimation de sources lumineuses utilisent le principe de l'acquisition de la luminance émise par une sphère. Y. Zhang et Y. Yang utilisent une image d'une sphère lambertienne de taille connue pour détecter la direction de multiples sources lumineuses directionnelles [ZY00]. Leur approche est basée sur la détection des contours d'ombre correspondant à la frontière entre la partie éclairée et la partie à l'ombre de chaque source lumineuse (figure 1.20.a). Chaque contour d'ombre est évalué par application d'une transformée de Hough

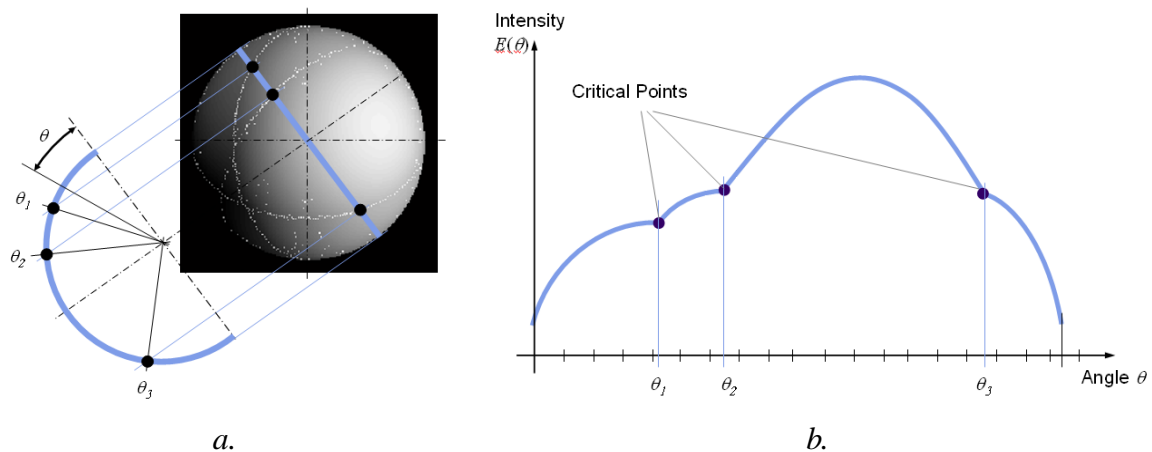


Figure 1.20 – a. Image d'une sphère lambertienne éclairée par trois sources lumineuses : trois contours d'ombre sont détectés par application d'une transformée de Hough sur les points critiques ; b. les points critiques sont détectés sur chaque arc de cercle passant par le centre de la sphère : ils correspondent à des changements brusques d'illumination sur les arcs, produits par un changement du nombre de sources lumineuses visibles (images extraites de [ZY00]).

sur un ensemble de points critiques correspondant à des changements d'illumination sur les arcs de la sphère (figure 1.20 pour plus de détails).

En 2004, C. Bouganis et M. Brookes ont amélioré cette approche en détectant les ambiguïtés (sources parfaitement opposées partageant un même contour d'ombre) et en définissant une nouvelle méthode d'estimation des points critiques [BB04]. Ils déplacent une fenêtre de  $5 \times 5$  pixels sur la sphère afin d'estimer les points critiques : à chaque déplacement, une erreur d'illumination minimale est évaluée sur les 25 pixels en appliquant une source lumineuse virtuelle. Lorsque cette erreur est inférieure à un seuil, le pixel central est marqué comme point critique. Cette technique limite le nombre de points critiques utilisés et améliore ainsi la robustesse de la transformée de Hough.

L'approche originale de Y. Zhang et Y. Yang est également améliorée par Y. Wang et D. Samaras en 2003 [WS03] en la combinant avec l'approche de Sato et al. [SSI01] développée à la section C.2.4. Ils utilisent un objet quelconque de géométrie connue (non obligatoirement une sphère) et définissent une méthode d'estimation de points critiques plus robuste que celle proposée dans [ZY00]. Après avoir obtenu une estimation des directions des sources lumineuses, les auteurs utilisent une approche similaire à [SSI01] pour obtenir une estimation précise de la luminance émise par chaque source à partir des luminances des pixels situés dans les régions d'ombre.

Y. Li et al. combinent plusieurs types d'informations extraites d'une image afin d'obtenir une méthode très robuste, applicable sur un objet texturé [LLLS03]. En effet, les textures présentes dans une scène perturbent généralement les algorithmes d'estimation de l'illumination et des propriétés de réflectance des matériaux. Dans cette nouvelle approche, la consistance physique est vérifiée entre les informations estimées : pour une direction d'illumination donnée, il existe une relation entre le contour d'ombre détecté sur la sphère et l'ombre de cette sphère projetée sur la surface plane (figure 1.21) L'estimation des points critiques introduit généralement

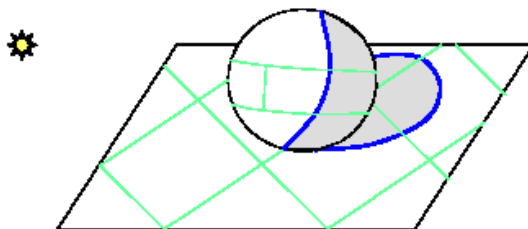


Figure 1.21 – Utilisation de plusieurs critères pour ne sélectionner que les vrais contours définissant la direction de la source lumineuse ; les contours bleus représentent les contours d'ombre et l'ombre projetée : une direction (celle de la source) établit une corrélation entre chaque point du contour d'ombre sur la sphère et chaque point de l'ombre projetée sur la surface plane ; les contours verts représentent les contours de la texture : il n'y a aucune corrélation entre les points verts, ni même entre les points verts et les points bleus (image extraite de [LLLS03]).

du bruit provenant des informations de textures. Par conséquent, les contours sont extraits par une segmentation de l'image puis sont classés en fonction de leur nature : contour d'ombre, ombre projetée ou contour de texture. Pour effectuer cette classification, l'ensemble des sources directionnelles éclairant potentiellement la scène est représentée par la discrétisation d'une hémisphère englobant la scène (comme dans [SSI99b]) et une évaluation hypothétique du contour d'ombre sur la sphère, de l'ombre projetée sur la surface plane et de la position du halo spéculaire sur l'image est effectuée pour chaque échantillon de l'hémisphère (correspondant à une source directionnelle).

En 2002, W. Zhou et al. proposent une autre approche basée sur l'acquisition d'une paire d'images stéréo d'une sphère possédant des propriétés lambertiennes et spéculaires [ZK02] (figure 1.22.a). Cette approche a l'avantage de n'avoir besoin ni de la position de la sphère, ni de son rayon : ces informations sont estimées en appliquant une reconstruction de la sphère par stéréovision (voir section B.2.3 pour plus de détails). Les surfaces lambertiennes ont la propriété d'émettre une luminance constante, quelle que soit leur direction d'émission. Seules les surfaces spéculaires émettent une luminance dépendant de la direction d'émission. Par conséquent, les luminances de la paire d'images peuvent être comparées afin de séparer les luminances prove-

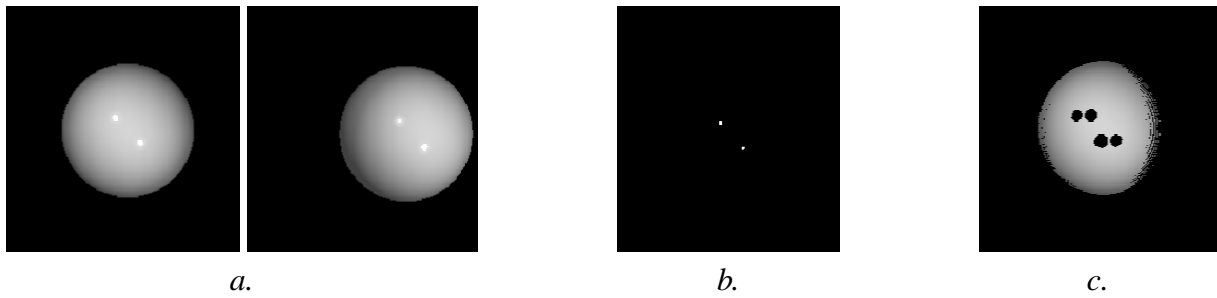


Figure 1.22 – a. Paire d’images stéréo d’une sphère aux propriétés lambertiennes et spéculaires ; b. image laissant apparaître les halos spéculaires après un seuillage sur les fortes différences de luminance entre les deux images présentées en a. ; c. image laissant apparaître les zones homogènes après un seuillage sur les faibles différences de luminance (images extraites de [ZK02]).

nant des directions de réflexion spéculaire des autres luminances variant faiblement entre les deux images. Une image de différence de luminances est évaluée à partir de la première image stéréo en soustrayant, pour chaque pixel, la valeur de luminance reçue sur la seconde image et correspondant au même point 3D sur la sphère : le nombre de sources lumineuses ainsi que leur direction peuvent ainsi être estimés (figure 1.22.b). Un seuillage sur les faibles différences de luminances permet également d’évaluer la luminance émise par chaque source détectée (figure 1.22.c). Avec deux images d’une sphère réelle, cette méthode permet l’estimation de plusieurs directions de sources lumineuses avec une imprécision comprise entre  $3^\circ$  et  $7^\circ$ . Cette imprécision dépend de l’orientation de la source par rapport à la caméra.

### C.3.3 Détection de sources ponctuelles

En 2001, M. Powell et al. ont proposé une méthode pour estimer plusieurs sources lumineuses ponctuelles à partir de l’acquisition d’une paire de sphères spéculaires [PSG01]. Une seule image calibrée est utilisée mais les deux sphères doivent être de géométrie connue et placées à égale distance de la caméra (figure 1.23.a). Après avoir détecté les halos spéculaires visibles sur les deux sphères, la mise en correspondance de chaque halo spéculaire entre les deux sphères permet d’estimer la position de chaque source ponctuelle par une méthode de triangulation (figure 1.23.b). Les coordonnées de chaque source lumineuse sont obtenues avec une imprécision d’environ  $13\text{cm}$  pour une source placée à  $1\text{m}$  des sphères et une distance de  $11\text{cm}$  entre les deux sphères. Cette imprécision s’accroît lorsque les sources lumineuses sont éloignées de l’axe optique de la caméra car la triangulation est effectuée dans des conditions critiques.

En 2004, T. Takai et al. ont proposé un système similaire permettant la détection de multiples sources lumineuses ponctuelles et directionnelles en utilisant deux sphères aux propriétés lambertiennes [TNMM04]. Les deux sphères n’étant pas placées au même endroit, elles sont soumises à des conditions d’éclairage différentes. Par conséquent, elles émettent des luminances différentes bien qu’elles soient toutes les deux lambertiennes et de même albédo (figure 1.24.a). La différence pixel-à-pixel entre les deux sphères permet d’intensifier l’illumination provenant des sources ponctuelles et de supprimer l’illumination provenant des sources directionnelles (figure 1.24.b). L’image des différences est convertie en une image avec représentation par contours, identique à des courbes de niveaux représentant l’altitude sur une carte géographique, afin de faciliter la détection des régions de l’image affectées par une source lumineuse, plusieurs sources



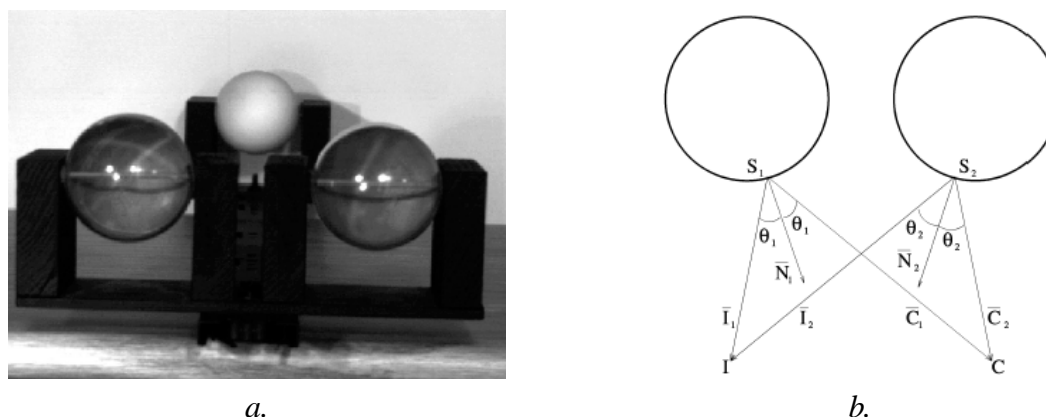


Figure 1.23 – a. Les deux sphères spéculaires, placées au premier plan, sont utilisées pour estimer les sources lumineuses ponctuelles qui apparaissent par des halos spéculaires sur l’image ; la troisième sphère, lambertienne, est utilisée pour estimer l’illumination réelle due aux sources lumineuses ponctuelles sur les surfaces lambertiennes ; b. triangulation entre le centre optique  $C$  et le halo vu sur chacune des sphères (en  $S_1$  et  $S_2$ ) afin d’estimer la position  $I$  de la source lumineuse (images extraites de [PSG01]).

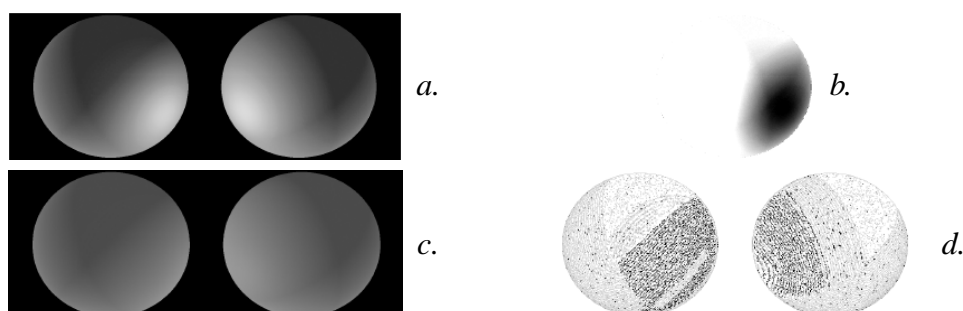


Figure 1.24 – a. Image des deux sphères lambertiennes éclairées par deux sources directionnelles et une source ponctuelle ; b. différence des deux sphères : seule la source ponctuelle apparaît ; c. suppression de l’illumination de la source ponctuelle dans l’image originale a. ; d. représentation par contours de l’image c. (images extraites de [TNMM04]).

lumineuses ou aucune source (figure 1.24.d). En faisant l’hypothèse que les sphères sont lambertiennes, chaque région éclairée par une seule source permet de détecter la position de cette source ainsi que sa luminance. Lorsque les sources ponctuelles sont détectées dans toutes les régions, leur illumination est supprimée de l’image originale afin d’avoir une image de sphères éclairées uniquement par des sources directionnelles (voir figure 1.24.c). Un processus identique est ensuite appliqué à l’image obtenue afin de détecter chacune des sources directionnelles. Pendant tout le processus de détection de sources lumineuses, chaque source estimée est vérifiée en recalculant sa contribution sur l’image originale : si les luminances des pixels de l’image sont plus faibles que les valeurs estimées, alors la source estimée est incorrecte et elle est supprimée de la liste des sources détectées.

## C.4 Discussions

Les méthodes de détection de sources lumineuses existantes sont variées, mais aucune ne permet l'extraction de sources multiples sans hypothèses fortes sur la réflectance de la surface ou la géométrie de la scène. En effet, chaque méthode répond à des besoins spécifiques et impose son lot de contraintes sur le nombre de points de vue, l'utilisation de sphères diffuses ou spéculaires, la connaissance de la géométrie de la scène ou la restriction à des surfaces lambertiennes ou texturées.

Les méthodes basées sur les informations contenues dans une seule image font toutes l'hypothèse que la scène est composée de surfaces lambertiennes. De plus, ces méthodes ne permettent de détecter que des sources lumineuses directionnelles. Seule la méthode proposée par E. Guillou détecte des sources ponctuelles mais elle nécessite l'intervention manuelle pour sélectionner des zones d'importance dans l'image. Nous voulons utiliser des objets de réflectance variée et non nécessairement lambertienne, par conséquent ce groupe de méthodes n'est pas adapté à nos besoins.

L'utilisation d'une ou plusieurs sphères dans l'environnement permet d'obtenir des résultats plus précis sur l'estimation de sources lumineuses multiples mais ce type de méthodes impose l'intrusion d'objets dans l'environnement. Le système de sphères multiples est assez contraignant car les sphères ne doivent pas être trop proches l'une de l'autre afin d'éviter les interrélaxions et doivent être positionnées judicieusement par rapport aux multiples sources lumineuses afin de ne pas projeter l'ombre de l'une sur l'autre. Les méthodes basées sur les calculs de différence de luminance provenant de deux sphères ou de deux points de vues différents sont les méthodes donnant les meilleurs résultats [PSG01, ZK02, TNMM04]. Elles permettent la détection de multiples sources directionnelles avec une imprécision inférieure à  $7^\circ$ .

Des détails sur l'implantation de plusieurs méthodes présentées dans cet état de l'art sont disponibles dans les rapports techniques rédigés par N. Funk [Fun03].

Il n'existe pas de méthode adaptée à nos besoins pour détecter de multiples sources lumineuses, toutes les méthodes présentées imposent des contraintes. Nous disposons d'un nombre conséquent de points de vue mais aucune méthode n'utilise plus de deux points de vue pour améliorer leur estimation. Nous pouvons toutefois utiliser plusieurs sphères spéculaires pour obtenir une estimation précise des sources lumineuses comme dans [PSG01], mais leur insertion provoque des changements importants d'éclairément. Ainsi, nous proposons une nouvelle méthode d'estimation de sources multiples exploitant les informations contenues dans les images sans avoir à introduire de sphères dans notre système d'acquisition.

## D Estimation de BRDFs

L'estimation des propriétés de réflectance des surfaces d'une scène ou d'un objet à partir d'images est généralement guidée par une représentation géométrique de la scène et les caractéristiques des sources lumineuses.

En synthèse d'image, de nombreux modèles de réflectance (ou BRDF) ont été définis pour représenter les interactions entre la lumière et le matériau. Les principales caractéristiques de surface prises en compte sont les réflexions diffuses, les réflexions spéculaires et l'anisotropie. D'autres modèles de BRDFs représentent des phénomènes lumineux plus complexes mais ils ne sont pas présentés ici car aucune méthode d'estimation de BRDFs ne permet de les paramétrer à partir d'une série d'images.



## D.1 Types de surfaces

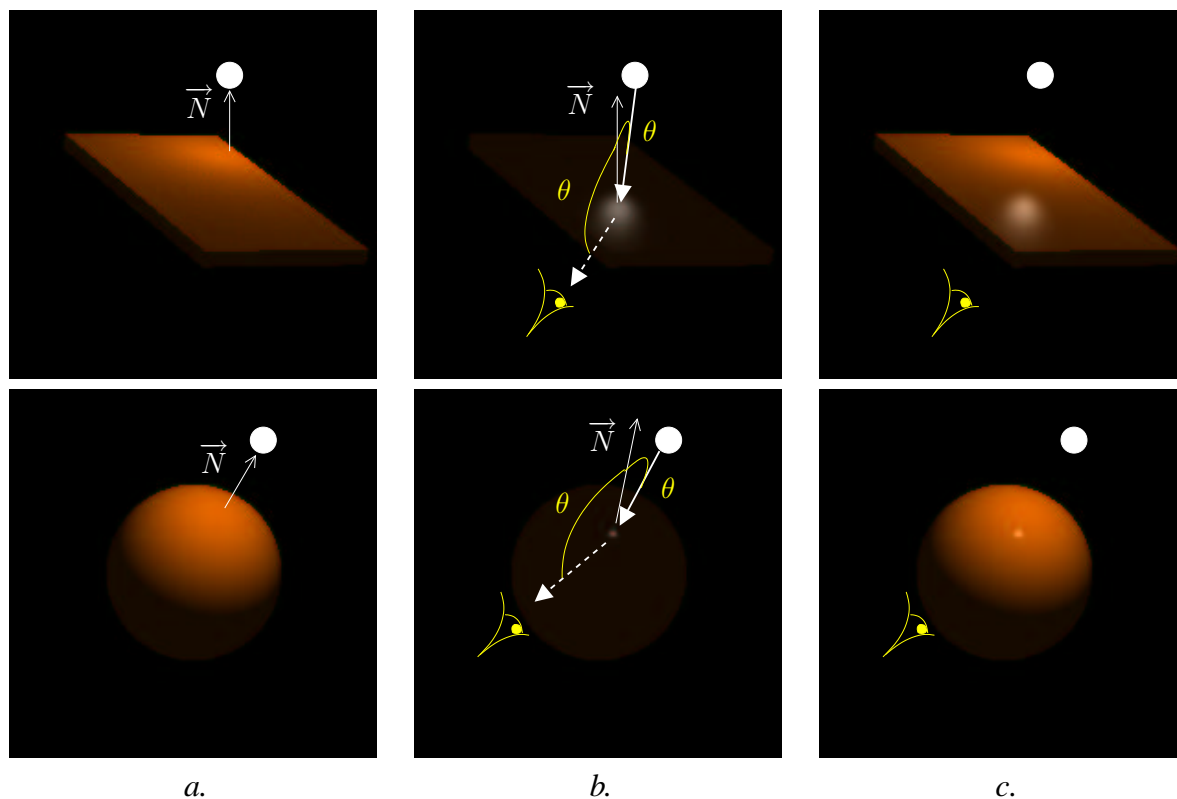


Figure 1.25 – a. Surfaces parfaitement diffuses ; b. surfaces spéculaires ; c. surfaces d’aspect diffus directionnel (surfaces mixtes).

### Surfaces diffuses

Une surface parfaitement diffuse réfléchit en un point une luminance constante, indépendante du point d’observation. La luminance réfléchie est maximale pour les points de la surface dont la normale est confondue avec la direction d’incidence car l’angle solide formé par un élément de surface  $dS$  est maximal dans cette direction (figure 1.25.a). Ce type de surface caractérise les surfaces dites *mates* comme par exemple le papier ou la craie.

### Surfaces spéculaires

Une surface parfaitement spéculaire réfléchit en un point une luminance uniquement dans la direction miroir de la direction d’incidence : la direction miroir est la direction symétrique par rapport à la normale. Sur une surface spéculaire, la luminance réfléchie est maximale pour les points de la surface dont la direction miroir est confondue avec la direction d’observation (figure 1.25.b). Ce type de surface caractérise les miroirs (cas d’une surface parfaitement spéculaire) ou les surfaces dites *brillantes* comme par exemple les surfaces métalliques.

### Surfaces d'aspect diffus directionnel

Une surface d'aspect diffus directionnel, également appelée surface mixte, est une surface possédant à la fois des propriétés diffuses et des propriétés spéculaires. La luminance réfléchie en un point dépend donc du point d'observation. Les halos spéculaires (taches brillantes) apparaissant sur les images sont plus larges sur les surfaces planes que sur les sphères et sont généralement teintées de la couleur de la source lumineuse pour des surfaces d'aspect plastique (figure 1.25.c).

### Surfaces anisotropes

Une surface anisotrope réfléchit la lumière dans des directions privilégiées. La réflexion diffuse réfléchit alors la luminance presque exclusivement dans les directions privilégiées et la réflexion spéculaire produit des halos spéculaires dans des directions proches des directions privilégiées et non proche de la direction de réflexion miroir. Ce type de surface caractérise les matériaux tels que les revêtements des CDs ou les métaux brossés (figure 1.26).

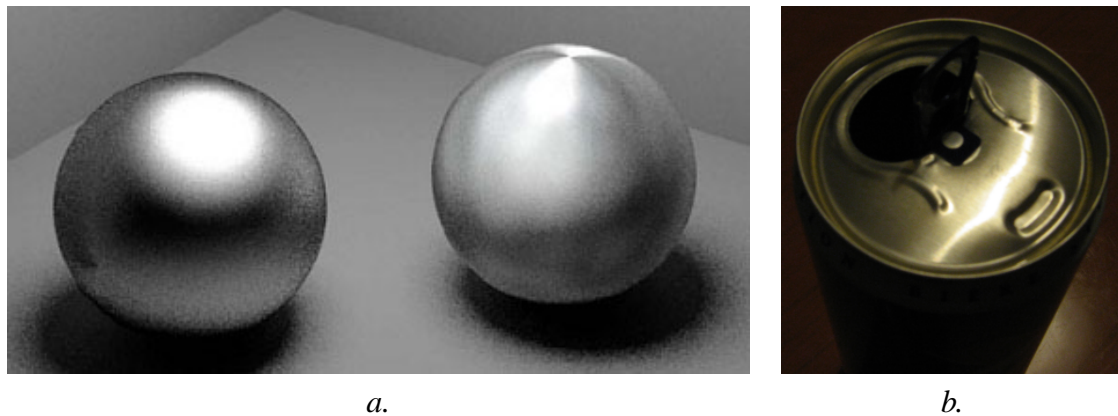


Figure 1.26 – a. Sphères d'aspect diffus directionnel ; la sphère de gauche représente une surface isotrope et celle de droite représente une surface anisotrope ; b. photographie d'une cannette de soda, métal brossé.

## D.2 Modèles de BRDFs

Divers modèles d'illumination (modèles de BRDFs) ont été proposés afin de représenter les propriétés de réflexion des surfaces par une équation de rendu. Ces modèles paramétriques ne sont pas toujours physiquement corrects mais ils sont utilisés en synthèse d'images pour une évaluation rapide de la luminance en un point. Un modèle est physiquement correct doit vérifier la loi de conservation de l'énergie et la propriété de réciprocity de Helmholtz (section C de l'annexe I pour plus de détails). L'équation de réflectance des surfaces, telle qu'elle est définie en section C de l'annexe I, n'est pas utilisable en pratique car l'intégration des luminances incidentes sur l'hémisphère nécessite des calculs complexes en chaque point de la surface : ce sont des calculs d'illumination globale généralement effectués par des méthodes de lancer de rayons récursif ou de radiosité. Les modèles d'illumination présentés dans cette section évaluent plutôt une illumination locale, ils sont basés sur une discrétisation de l'intégrale des luminances incidentes : les sources lumineuses visibles depuis un point de la scène sont les seules directions

utilisées pour calculer l'illumination en ce point. D'autres modèles de BRDFs existent, ils ne sont pas présentés car sont trop complexes pour être utilisés par une méthode d'estimation de la réflectance. Une classification des différents modèles de réflectance est disponible dans [Sch94].

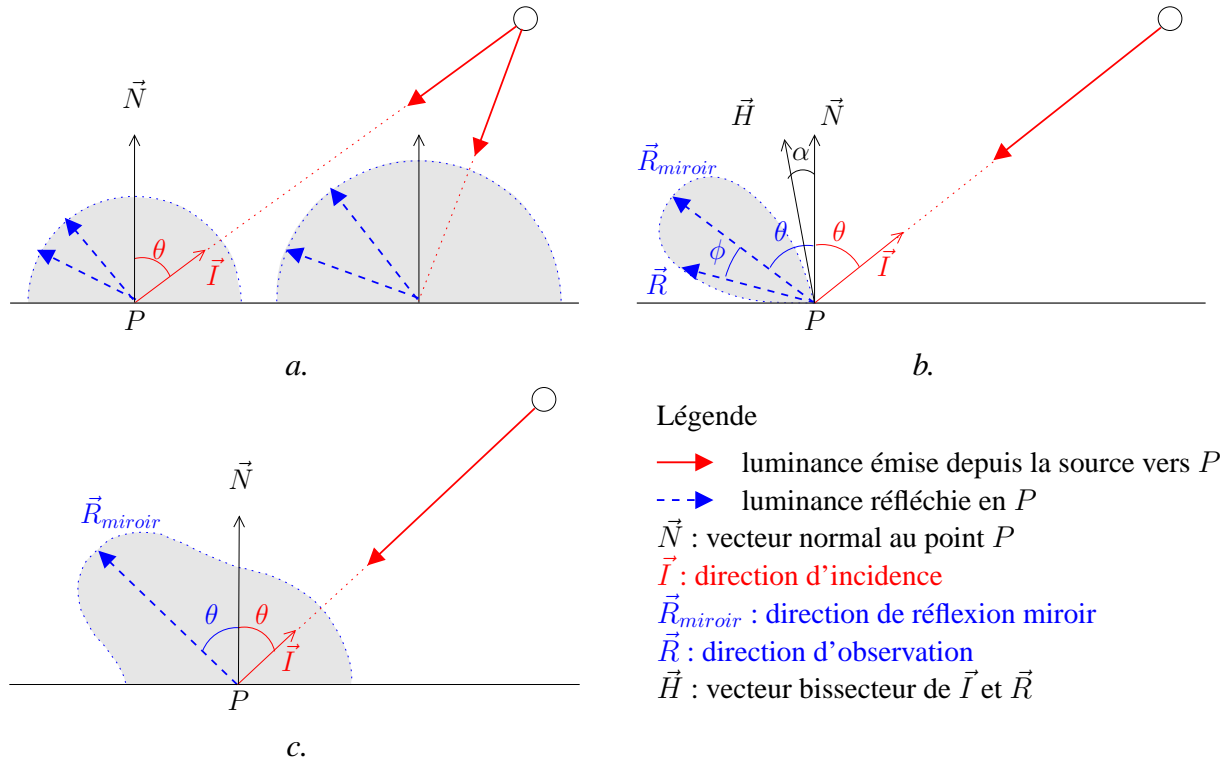


Figure 1.27 – a. Modèle de Lambert ; b. lobe spéculaire ; c. modèle de Phong.

### Modèle de Lambert

Le modèle de Lambert permet de représenter des surfaces parfaitement diffuses (figure 1.27.a). La luminance réfléchiée en tout point suit la loi de Lambert : elle est uniforme en un point de la surface, i.e. constante pour toutes les directions de réflexion [Lam60]. L'équation d'illumination d'un point  $P$  éclairé par une source lumineuse de direction  $\vec{I}$  peut être écrite :

$$L_{r,\lambda}(P) = L_{i,\lambda}(P, \vec{I}) \cdot K_{d,\lambda} \cdot \cos \theta \cdot \frac{1}{r^2} \quad (1.4)$$

avec  $K_d$  le coefficient de réflexion diffuse de la surface ( $K_d \in [0, 1]$ ) ;  $\theta$  l'angle incidence ( $\cos \theta = (\vec{I} \cdot \vec{N})$  borné sur  $[0, 1]$ ) et  $r$  la distance séparant la source du point  $P$ . Le terme  $1/r^2$  provient du calcul de l'angle solide  $d\omega_i$  présent dans l'équation de réflectance et n'existe plus lors d'un éclairage par une source directionnelle (section D de l'annexe I). L'équation doit être évaluée pour chaque longueur d'onde  $\lambda$ . En pratique, elle est généralement calculée sur chaque composante de l'espace colorimétrique utilisé (par exemple sur les 3 composantes de l'espace RGB). Par conséquent le coefficient  $K_d$  peut prendre une valeur différente pour chaque composante couleur.

Si plusieurs sources lumineuses sont visibles depuis  $P$ , la luminance émise par  $P$  correspond à la somme de l'évaluation de l'équation d'illumination sur chacune de ces sources :

$$L_r(P) = L_a \cdot K_a + \sum_{s=1}^{\#Sources\ visibles} \left( L_i(P, \vec{I}_s) \cdot K_d \cdot \cos \theta_s \cdot \frac{1}{r_s^2} \right) \quad (1.5)$$

avec  $L_a \cdot K_a$  un terme ambiant correspondant à aucune source lumineuse physique mais généralement utilisé dans les modèles d'illumination locale pour compenser la perte d'énergie lumineuse provenant des interréflections lumineuses non évaluées par le modèle.  $r_s$  est la distance séparant la source visible  $s$  du point  $P$ .

### Modèle de Phong

B. Phong a amélioré le modèle de Lambert en ajoutant la contribution provenant de la réflexion spéculaire [Pho75]. Il modélise de manière empirique la réflexion spéculaire par des directions de réflexion privilégiées, formant un lobe de directions autour de la direction de réflexion miroir  $\vec{R}_{miroir}$  (figure 1.27.b). L'équation d'illumination d'un point  $P$  éclairé par une source lumineuse de direction  $\vec{I}$  s'exprime en fonction de la direction de réflexion  $\vec{R}$  :

$$L_r(P, \vec{R}) = L_a \cdot K_a + L_i(P, \vec{I}) \cdot (K_d \cdot \cos \theta + K_s \cdot \cos^n \phi) \cdot \frac{1}{r^2} \quad (1.6)$$

avec  $K_s$  le coefficient de réflexion spéculaire de la surface ( $K_s \in [0, 1]$ );  $n$  l'exposant de réflexion spéculaire ( $n \geq 1$ ) et  $\phi$  l'angle de déviation par rapport à la direction spéculaire ( $\cos \phi = (\vec{R} \cdot \vec{R}_{miroir})$  borné sur  $[0, 1]$ ). L'augmentation de la valeur de  $n$  entraîne un affinement du lobe spéculaire, i.e. une diminution de la taille du halo spéculaire produit dans la direction de réflexion miroir. Une valeur infinie de  $n$  correspond à un miroir parfait. Contrairement au coefficient  $K_d$ , le coefficient  $K_s$  est souvent identique pour les trois composantes couleurs. Cette valeur fixe permet au halo spéculaire de réfléchir la couleur de la source lumineuse (représentation des surfaces plastiques). Lorsque les valeurs des coefficients  $K_s$  sont différentes, elles varient proportionnellement aux coefficients  $K_d$  afin que le halo réfléchisse plutôt la couleur de la surface (représentation des surfaces métalliques).

La somme des coefficients de réflexion diffuse et de réflexion spéculaire doit être inférieure à 1 afin de ne pas émettre une plus forte luminance dans une direction que la luminance reçue (figure 1.27.c). Néanmoins ce modèle reste empirique car la loi de conservation de l'énergie énoncée à la section C de l'annexe I ne peut pas être respectée : la luminance totale réfléchie en un point dans toutes les directions de l'hémisphère est souvent supérieure à la luminance incidente. Il existe des variantes de ce modèle pour évaluer plus rapidement la réflexion spéculaire. Par exemple J. Blinn a proposé d'utiliser l'angle bissecteur  $\alpha$  à la place de  $\phi$  ( $\alpha = \phi/2$ , figure 1.27.b).

### Modèle de Phong modifié

En 1994, R. Lewis propose une modification du modèle de Phong afin qu'il devienne physiquement correct (respect de la loi de conservation de l'énergie et du principe de réciprocité de Helmholtz) [Lew94]. Voici l'équation d'illumination de ce nouveau modèle (extraite de [LW94]) :

$$L_r(P, \vec{R}) = L_i(P, \vec{I}) \cdot \left( K_d \cdot \frac{1}{\pi} \cdot \cos \theta + K_s \cdot \frac{n+2}{2\pi} \cdot \cos \theta \cdot \cos^n \phi \right) \cdot \frac{1}{r^2} \quad (1.7)$$

Le modèle est plus proche de l'équation de luminance (voir section D de l'annexe I) car il peut s'exprimer en fonction de la BRDF de la surface par intégration sur l'ensemble discret des sources lumineuses visibles depuis  $P$  :

$$L_r(P, \vec{R}) = \sum_{s=1}^{\#Sources\ visibles} \left( f_r(P, \vec{R}, \vec{I}_s) \cdot L_i(P, \vec{I}_s) \cdot \cos \theta_s \cdot \frac{1}{r_s^2} \right) \quad (1.8)$$

avec  $f_r$  :

$$f_r(P, \vec{R}, \vec{I}_s) = K_d \cdot \frac{1}{\pi} + K_s \cdot \frac{n+2}{2\pi} \cdot \cos^n \phi_s \quad (1.9)$$

### Modèle de Torrance-Sparrow

En 1967, K. Torrance et E. Sparrow ont développé un modèle de BRDF adapté aux surfaces rugueuses [TS67]. Ils définissent une surface rugueuse par de petites surfaces planes et spéculaires appelées microfacettes. Ce modèle physiquement correct est exprimé par une combinaison linéaire d'une partie diffuse et d'une partie spéculaire :

$$L_r(P, \vec{R}) = L_i(P, \vec{I}) \cdot \left( K_d \cdot \cos \theta + K_s \cdot \frac{1}{\cos \theta_r} \cdot \exp\left(-\frac{\alpha^2}{\sigma^2}\right) \cdot G(\theta, \theta_r) \cdot F(\theta, \rho) \right) \cdot \frac{1}{r^2} \quad (1.10)$$

avec  $\theta_r$  l'angle de réflexion ( $\cos \theta_r = (\vec{R} \cdot \vec{N})$  borné sur  $[0, 1]$ );  $\alpha$  l'angle entre le vecteur bissecteur  $\vec{H}$  et le vecteur normal  $\vec{N}$ ;  $\sigma$  le coefficient de rugosité et  $\rho$  l'indice de réfraction de la surface pour une incidence normale à la surface. L'équation est exprimée en fonction de  $G$ , le coefficient d'atténuation géométrique : pourcentage de luminance reçue (non obstruée par une autre facette), il est également appelé le coefficient d'auto-ombrage ( $G \in [0, 1]$ ); et  $F$  le facteur de Fresnel : il exprime la réflexion de la luminance par les microfacettes orientées dans la direction de  $\vec{H}$  ( $F \in [0, 1]$ ). Les coefficients  $K_d$  et  $F$  peuvent être exprimés en fonction de la longueur d'onde  $\lambda$ .

### Modèle de Cook-Torrance

En 1981, R. Cook et K. Torrance proposent une extension du modèle de Torrance-Sparrow représentant plus précisément les surfaces rugueuses en modifiant la fonction de distribution des pentes des microfacettes [CT82] :

$$L_r(P, \vec{R}) = L_i(P, \vec{I}) \cdot \left( K_d \cdot \frac{1}{\pi} \cdot \cos \theta + K_s \cdot \frac{1}{4\pi \cdot \cos \theta_r} \cdot D(\alpha, \omega) \cdot G(\theta, \theta_r) \cdot F(\beta) \right) \cdot \frac{1}{r^2} \quad (1.11)$$

avec  $\omega$  l'angle formé entre le vecteur projeté de  $\vec{H}$  sur la surface et un vecteur de référence de la surface (précédemment défini); et  $\beta$  l'angle bissecteur (angle entre  $\vec{I}$  et  $\vec{H}$ ). L'équation est exprimée en fonction de  $D$ , la fonction de distribution des pentes des microfacettes : elle définit la quantité de facettes orientées dans la direction bissectrice  $\vec{H}$  ( $D \geq 0$ ) et remplace le coefficient  $\sigma$  précédemment utilisé;  $G$  représente toujours le coefficient d'atténuation géométrique; et  $F$ , le facteur de Fresnel, n'utilise plus l'indice de réfraction.

### Modèle de Ward

En 1992, G. Ward a proposé un modèle de BRDF simple et adapté aux surfaces anisotropes [War92]. Ce modèle physiquement correct est exprimé, comme les modèles présentés précédemment, par une combinaison linéaire d'une partie diffuse et d'une partie spéculaire. Un modèle gaussien est utilisé pour la partie spéculaire et un modèle elliptique pour la partie anisotropique :

$$L_r(P, \vec{R}) = L_i(P, \vec{I}) \cdot \left( K_d \cdot \frac{1}{\pi} \cdot \cos \theta + K_s \cdot \frac{\sqrt{\cos \theta}}{4\pi \cdot \sqrt{\cos \theta_r}} \cdot D(\alpha, \omega) \right) \cdot \frac{1}{r^2} \quad (1.12)$$

La principale différence par rapport au modèle de Cook-Torrance est la suppression du coefficient d'atténuation géométrique et du facteur de Fresnel. En revanche,  $D$  la fonction de distribution de pentes de la surface définit une pente sur deux axes orthogonaux tangents à la surface afin de modéliser des surfaces anisotropes.

### Modèle de Lafortune

En 1997, E. Lafortune et al. proposent un modèle basé sur des sommes de lobes en cosinus (le lobe en cosinus est la primitive utilisée dans la composante de réflexion spéculaire du modèle de Phong :  $K_s \cdot \cos^n \phi$ ) afin de représenter les composantes diffuses-directionnelles non paramétrables avec les autres modèles de BRDF mais souvent rencontrées sur des surfaces réelles [LFTG97]. Ce modèle permet ainsi d'obtenir une réflexion importante dans les directions de vue rasantes et dans la direction de rétro-réflexion (direction d'incidence). De plus les surfaces soumises à des directions d'incidence rasantes peuvent émettre des luminances non négligeables. Voici l'équation d'illumination de ce modèle :

$$L_r(P, \vec{R}) = L_i(P, \vec{I}) \cdot \left( \sum_{l=1}^{\#Lobes} K_{s,l} \cdot (C_{x,l} \cdot I_x \cdot R_x + C_{y,l} \cdot I_y \cdot R_y + C_{z,l} \cdot I_z \cdot R_z)^n \right) \cdot \cos \theta \cdot \frac{1}{r^2} \quad (1.13)$$

avec  $K_s$ ,  $C_x$ ,  $C_y$ ,  $C_z$  et  $n$  les coefficients paramétrant un lobe ;  $(I_x, I_y, I_z)$  et  $(R_x, R_y, R_z)$  sont les coordonnées des directions d'incidence  $\vec{I}$  et de réflexion  $\vec{R}$  exprimées dans un repère local au point  $P$  où l'axe  $z$  représente la direction du vecteur normal  $\vec{N}$  ;  $K_s$  est le coefficient de réflexion spéculaire, il peut être intégré aux coefficients  $(C_x, C_y, C_z)$  : seulement quatre coefficients paramètrent réellement un lobe ; pour une surface isotrope  $C_x = C_y$  : seulement trois paramètres sont utilisés dans la plupart des cas ;  $n$  représente l'exposant de spécularité ; généralement  $C_x$  est négatif, une valeur positive de  $C_x$  permet la rétro-réflexivité (réflexion vers la direction d'incidence) ; l'augmentation de  $|C_x|$  par rapport à  $|C_z|$  permet de diriger le pic de spécularité vers les directions rasantes.

Ce modèle est physiquement correct, un lobe  $l$  peut être utilisé pour représenter la composante diffuse du modèle de Phong modifié avec  $n_l = 0$  et  $K_{s,l} = K_d/\pi$ .

### Autres modèles

Le modèle d'Oren-Nayar [ON94] représente les surfaces rugueuses par des microfacettes diffuses. Leur modèle permet de représenter des surfaces telles que les murs crépis ou les poteries.

Un modèle de réflectance très complet a été développé par X. He et al. afin de modéliser la plupart des phénomènes physiques de réflexion de la lumière sur des surfaces rugueuses (polarisation, diffraction, interférences, petite rugosité de la surface pour des angles d'incidence rasants)

[HTSG91]. Le modèle est défini par une combinaison linéaire d'une partie réflexion spéculaire, une partie réflexion diffuse uniforme et une troisième partie réflexion diffuse directionnelle obtenue par intégration de la diffraction et autres interférences. Ce modèle paramétrique n'est pas présenté dans cette étude car il présente beaucoup de paramètres et devient inutilisable dans le cadre de l'estimation des propriétés de réflectance de la surface à partir d'images.

Deux approches différentes permettent d'estimer la BRDF d'une surface : soit la réflectance est acquise précisément avec un matériel spécifique afin de reconstruire un échantillonnage dense de la fonction, soit la réflectance est représentée par un modèle paramétrique : les paramètres du modèle sont estimés à l'aide d'échantillons de réflectance préalablement acquis.

## D.3 Acquisition d'un champ de réflectance

### D.3.1 Réflectance uniforme

Une mesure précise de la BRDF d'une surface peut être enregistrée avec un matériel spécifique appelé gonioréfectomètre. Cependant, ce matériel coûte cher, demande un calibrage très précis et prend beaucoup de temps pour une acquisition complète de la BRDF (l'acquisition est effectuée sur 4 dimensions : 2 angles pour la direction d'incidence, 2 angles pour la direction de vue ; une dimension supplémentaire peut être requise pour la longueur d'ondes) [Foo97]. L'acquisition d'une BRDF est généralement effectuée sur une surface aux propriétés de réflectance uniforme ; dans le cas contraire, l'acquisition doit être effectuée en un point précis de la surface et ne reflète pas la réflectance de la surface complète. Une lumière laser est souvent utilisée afin d'éviter les interférences lumineuses sur la surface : par exemple, la luminance incidente en un point de la surface peut modifier la luminance émise en un autre point de la surface (ces phénomènes sont mesurés dans les représentations par une BSSRDF ou un champ de réflectance complet, voir section A.2.5).

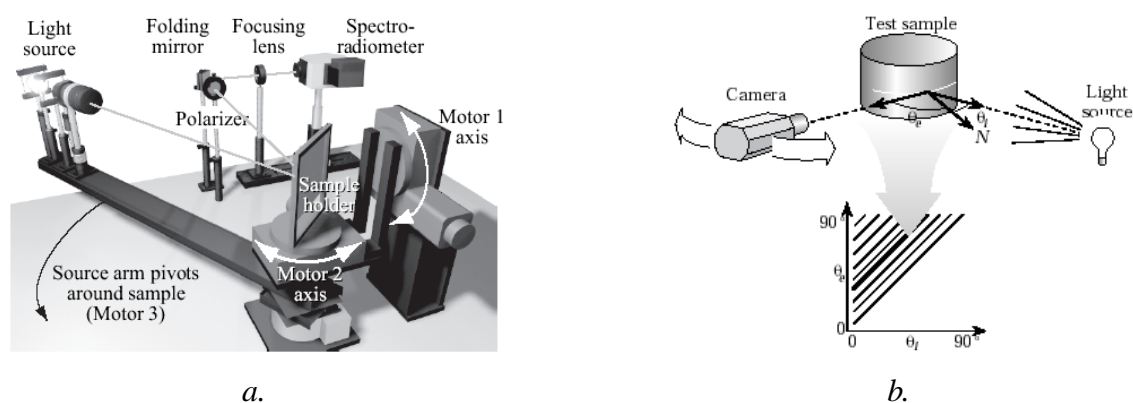


Figure 1.28 – Gonioréfectomètres permettant l'acquisition d'une BRDF isotrope ; a. système précis avec 3 axes de liberté et un pas de rotation de  $0.1^\circ$  (image extraite de [LFTW06]) ; b. système plus modeste avec 2 axes de liberté (image extraite de [MWLT00]).

H. Li et al. proposent un gonioréfectomètre entièrement automatisé, permettant l'acquisition précise de la BRDF d'une surface plane isotrope en faisant varier seulement 3 axes de rotation



(voir figure 1.28.a) [LFTW06]. L'isotropie permet de réduire l'acquisition de la BRDF à 3 degrés de liberté.

S. Marschner et al. propose un système d'acquisition moins onéreux, constitué d'un appareil photographique et d'une source ponctuelle, et permettant également l'acquisition de la BRDF d'une surface isotrope [MWLT00]. En utilisant un objet sphérique ou cylindrique, l'acquisition peut être réduite à seulement deux dimensions : seule la caméra évolue autour de l'objet, une ligne de pixels est utilisée pour l'acquisition de la BRDF et les trois composantes chromatiques ( $R, G, B$ ) échantillonnent l'ensemble des longueurs d'ondes (voir figure 1.28.b). Les interférences lumineuses sont négligées dans ce système d'acquisition.

Les systèmes d'acquisition décrits dans cette section supposent une fonction de réflectance uniforme sur toute la surface acquise. Cependant comme nous ne voulons pas acquérir des objets composés d'une seule réflectance, ces systèmes d'acquisition ne sont pas utilisables.

### D.3.2 Réflectances variées

Pour faire l'acquisition d'une réflectance non uniforme, une caméra fixe est utilisée pour enregistrer la réflectance de tous les points de la surface de l'objet tandis qu'une source lumineuse est déplacée en diverses positions afin que la luminance émise soit enregistrée en fonction de la direction d'incidence.

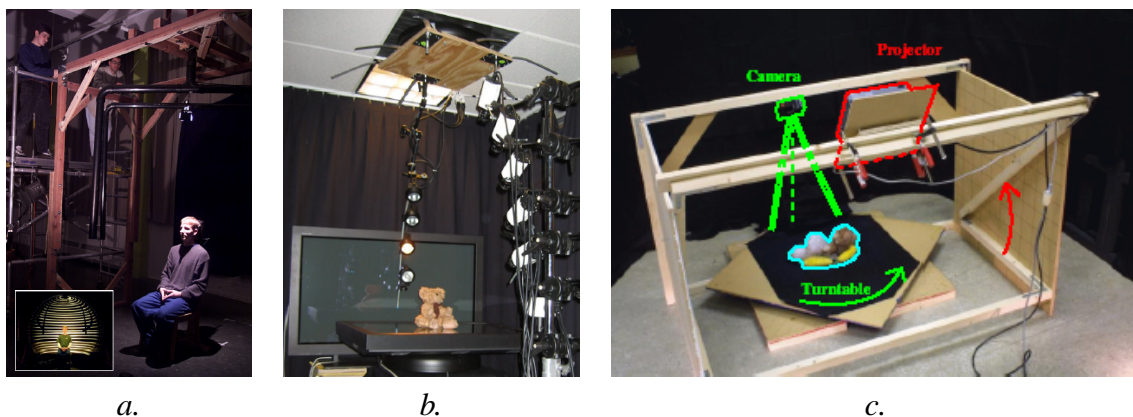


Figure 1.29 – Systèmes d'acquisition permettant l'enregistrement de propriétés de réflectance avec des conditions d'éclairage variant entre chaque image ; a. système automatisé utilisé pour l'acquisition de visages humains (image extraite de [DHT<sup>+</sup>00]) ; b. système automatisé utilisé pour l'acquisition d'objets à géométrie et photométrie complexes comme les fourrures, les peluches, etc (image extraite de [MPN<sup>+</sup>02]) ; c. système manuel utilisé pour des objets à photométrie complexe (image extraite de [MPDW03]).

P. Debevec et al. utilisent un champ de réflectance pour stocker les données acquises (voir section A.2.5 pour plus de détails sur leur système d'acquisition) [DHT<sup>+</sup>00]. Ils effectuent l'acquisition complète de 4 dimensions (2 angles de direction d'incidence et 2 paramètres de position sur la surface de l'objet) puis étendent l'acquisition à 6 dimensions (2 angles de direction de vue) en enregistrant les données depuis deux points de vue différents. Pendant l'acquisition de la luminance émise par toute la surface, une source lumineuse directionnelle illumine la surface en



2048 positions différentes : les interréflexions de la surface sur elle-même (auto-interréflexions) sont donc négligées et 6 dimensions suffisent à décrire le champ de réflectance complet.

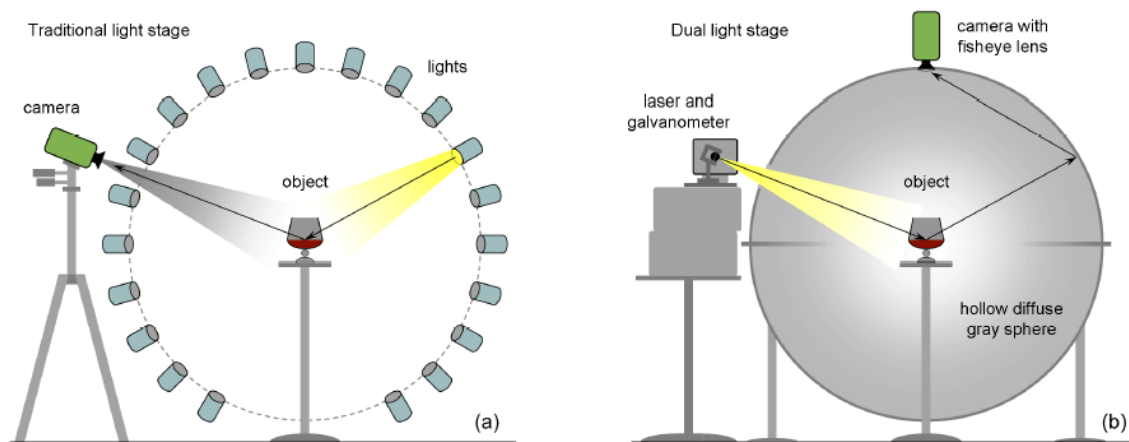


Figure 1.30 – Le système d'acquisition, appelé *dual light stage* est composé d'une sphère diffuse englobant l'objet, d'une caméra avec objectif grand angle et d'une lumière laser, et il permet l'acquisition d'un champ de réflectance 4D. Le principe de réciprocité permet de connaître la quantité de lumière émise par la surface de l'objet vers le point de vue situé à la place de la lumière laser et pour n'importe quelle direction de lumière incidente.

En 2005, T. Hawkins et al. proposent une acquisition simplifiée du champ de réflectance 4D paramétré par 2 angles de direction d'incidence et 2 paramètres de position sur la surface de l'objet [HED05]. Le principe de la réciprocité de Helmholtz est appliqué pour inverser le transport de la lumière sur un système d'acquisition composé d'une sphère diffuse englobant l'objet à acquérir, d'une lumière laser projetant un rayon de lumière sur l'objet et d'une caméra avec un objectif grand angle. La caméra permet d'acquérir le faisceau laser réfléchi par un point de la surface de l'objet vers toutes les directions de la sphère englobante (la figure 1.30 détaille le système d'acquisition). Le principe de réciprocité sur le transport de la lumière permet, à partir d'un enregistrement de la caméra, de connaître la luminance émise par le point de l'objet touché par le rayon laser, vers la lumière laser (c'est là qu'intervient la réciprocité) et pour chaque direction de la luminance incidente (correspondant à des points de la sphère diffuse, i.e. à des pixels sur l'image acquise par la caméra). En balayant le rayon laser sur toute la surface visible de l'objet, il est possible de connaître la réflectance de la surface de l'objet pour le point de vue représenté par la position de la lumière laser. Ce système a l'avantage de ne pas enregistrer les auto-interréflexions lumineuses, contrairement à [DHT<sup>+</sup>00], mais il est limité à l'acquisition de la réflectance pour un seul point de vue.

W. Matusik et al. définissent un champ de réflectance en effectuant l'acquisition complète des 6 dimensions utilisées par P. Debevec et al. : les données sont enregistrées sur 216 positions différentes de la caméra [MPN<sup>+</sup>02]. Une source lumineuse directionnelle est également utilisée, limitant la reconstruction du champ de réflectance à 6 dimensions au lieu de 8, et seulement 60 positions différentes de source lumineuse sont acquises afin de limiter la taille des données stockées : environ 13000 images sont acquises pour construire le champ de réflectance. Deux écrans plasma sont également utilisés pour acquérir une carte de transparence en chaque point de vue : ceci permet un rendu de meilleure qualité pour les surfaces complexes comme les cheveux, les fourrures, etc.

V. Masselus et al. font l'acquisition d'un champ de lumière incident complet (4 dimensions) en déplaçant une source lumineuse ponctuelle en 224 positions différentes proches de l'objet et en prenant des images avec une caméra fixe où seule une petite partie de la surface est directement illuminée [MPDW03]. Le système permet ainsi de réduire les auto-interréflexions lumineuses mais l'acquisition demande plus de temps que les systèmes utilisant une lumière directionnelle : plus de 40 heures pour l'acquisition du champ de réflectance d'un objet en un seul point de vue. Evidemment les déplacements manuels de la source lumineuse et du cache positionné devant le projecteur pour réduire la zone de l'objet illuminée augmentent considérablement le temps d'acquisition : le cache est déplacé en 256 positions différentes pour chaque position de source lumineuse, soit environ 45000 images acquises pour un seul point de vue. Si une telle acquisition était effectuée avec une caméra se déplaçant tout autour de l'objet, la construction d'un champ de réflectance complet sur les 8 dimensions serait envisageable.

L'acquisition d'un champ de réflectance complet est long et fastidieux. De plus, comme le champ de réflectance nécessite une très forte capacité de stockage, nous n'utilisons cette méthode d'acquisition, nous préférons estimer un modèle paramétrique beaucoup moins coûteux.

## D.4 Estimation d'un modèle paramétrique

Les techniques de représentation de la BRDF par un modèle paramétrique réduisent considérablement les coûts en mémoire, comparées aux champs de réflectance. Un modèle d'illumination est généralement représenté par quelques coefficients (entre 1 et 5 suivant les modèles) et permet d'obtenir une représentation complète de la BRDF sur toutes les directions d'incidence et de réflexion.

### D.4.1 Réflectance uniforme

Tous les modèles de BRDF physiquement corrects présentés dans la section D.2 ont été validés par des mesures expérimentales de surfaces réelles. Pour cela, un ensemble d'images représentant une surface plane éclairée par une source lumineuse est acquis pour des positions de caméra et de source variées. Les paramètres de la source et de la caméra sont connus sur chaque image afin de vérifier le comportement du modèle de BRDF lors de l'évaluation de l'équation de luminance utilisant ce modèle. G. Ward a simplifié l'acquisition de surfaces anisotropes (généralement 4 degrés de liberté) en utilisant une caméra munie d'un objectif grand angle ( $180^\circ$ ) et une hémisphère miroir réfléchissant la luminance émise par la surface vers la caméra [War92]. Ainsi les luminances réfléchies par la surface sont toutes capturées par une seule image : la modification de la direction d'incidence suffit à acquérir la BRDF complète de la surface.

Pour représenter la réflectance d'une surface par un modèle paramétrique, celui-ci doit être choisi en fonction du type de surface à modéliser (spéculaire, anisotropique, rugueuse) et les paramètres de ce modèle sont estimés par une méthode de régression non linéaire minimisant l'erreur entre la luminance extraite des images de la surface et la luminance obtenue par évaluation de l'équation de luminance. Le modèle géométrique de l'objet et les conditions d'illumination doivent être connus afin d'obtenir une estimation précise des paramètres de la BRDF. La régression non linéaire est souvent effectuée par application de la méthode de Levenberg - Marquardt [PTVF99].

K. Ikeuchi et K. Sato estiment les propriétés de réflectance d'un objet à géométrie complexe à partir d'une seule image et de sa géométrie (carte de profondeur) en utilisant un modèle simplifié de Torrance-Sparrow [IS91]. Le modèle de réflectance estimé est formé de trois composantes : un lobe lambertien, un lobe spéculaire et un pic spéculaire afin de mieux modéliser les halos spéculaires visibles sur l'image. P. Debevec utilise le modèle de Lafortune pour estimer les propriétés de réflectance de matériaux composés de marbre [Deb04]. Un lobe diffus, un lobe spéculaire et un lobe rétro-réfléctif sont estimés.

T. Yu et al. utilisent un algorithme itératif afin d'estimer conjointement les paramètres de réflectance du modèle de Phong et les paramètres géométriques d'une surface paramétrique [YXA04]. Une première estimation de la surface paramétrique est obtenue par un algorithme de sculpture utilisant la silhouette de l'objet vue sur les images. L'algorithme itératif estime soit les paramètres de réflectance avec une géométrie fixe, soit les paramètres géométriques avec une réflectance fixe, et ceci jusqu'à convergence des deux modèles (modèle de réflectance et modèle géométrique).

En 2005, K. Hara et al. proposent une méthode d'estimation simultanée des propriétés de réflectance de surfaces mixtes et des paramètres de la source ponctuelle éclairant la scène à partir d'une seule image [HNI05]. Une première estimation de la position de la source est obtenue à partir du modèle géométrique et des halos spéculaires visibles sur l'image. Le coefficient de réflexion diffuse est estimé à partir des pixels hors des halos spéculaires. Puis un algorithme itératif estime soit les coefficients de réflexion spéculaire du modèle simplifié de Torrance-Sparrow avec la position de la source fixe, soit la position de la source avec les paramètres de réflectance fixes. Chaque modification de  $K_s$  entraîne plusieurs passes de modification de la valeur  $\sigma$  avant de modifier la position de la source. La valeur de  $\sigma$  est raffinée plus souvent car elle intervient dans la partie exponentielle du modèle de BRDF. Pour conserver les éventuelles textures présentes sur l'image, une estimation finale d'un coefficient  $K_d$  est effectuée localement, sur chaque pixel, à partir des données estimées (position de la source,  $K_s$  et  $\sigma$ ).

### D.4.2 Réflectances variées

Comme l'utilisation d'un modèle de BRDF uniforme sur tout l'objet n'est généralement pas réaliste, plusieurs méthodes sont proposées afin d'estimer les coefficients d'une BRDF paramétrique sur des ensembles de données plus restreints : par exemple sur des régions de la surface, sur des triangles du maillage ou même sur des entités plus petites (des texels d'une texture appliquée sur chaque triangle). La taille des surfaces utilisées pour estimer les coefficients d'un modèle de réflectance dépend de l'échantillonnage à disposition, c'est-à-dire du nombre de points de vue et de la résolution des images. Quatre méthodes, les plus intéressantes, sont présentées en détails dans cette section, chacune utilisant un modèle géométrique et les caractéristiques des sources lumineuses.

En 1997, Y. Sato et al. [SWI97] estiment la réflectance d'un objet composé de plusieurs BRDFs en utilisant une méthode similaire à [IS91]. Ils disposent d'une série de 120 images acquises autour de l'objet avec une source lumineuse directionnelle positionnée près de la caméra et une géométrie reconstruite en un maillage de 500 triangles à partir d'un nuage de points 3D. L'estimation des réflectances est entièrement basée sur le modèle de Torrance-Sparrow après séparation de la partie diffuse et de la partie spéculaire sur chacune des images. Comme dans [IS91], le modèle de Torrance-Sparrow est simplifié en utilisant un coefficient d'atténuation géométrique  $G = 1$  et un facteur de Fresnel  $F = 1$ . Pour obtenir une estimation précise des coeffi-

cients de réflexion, une carte de normale de  $20 \times 20$  valeurs est estimée pour chaque triangle du modèle géométrique à partir du nuage de points 3D. Une carte de  $80 \times 80$  points 3D est également utilisée en chaque triangle afin de représenter, en chaque point, l'ensemble des luminances émises provenant de la projection des pixels des images sur l'objet géométrique. A partir de cette carte de luminance est estimée une carte des coefficients de réflexion diffuse, également de résolution  $80 \times 80$  par triangle, représentant les détails de texture : les coefficients  $K_d$  sont estimés en chaque point de la carte et pour les trois composantes colorimétriques ( $R, G, B$ ). L'ensemble des cartes de coefficients  $K_d$  représente la texture diffuse de la surface, privée de l'illumination, et est souvent appelée la carte d'albédo. Pour estimer les coefficients de réflexion spéculaire, trop peu d'échantillons sont disponibles dans la direction de réflexion spéculaire en chaque point. Par conséquent, les coefficients de réflexion spéculaire sont estimés uniquement sur les halos spéculaires visibles sur les images (une centaine d'échantillons) et une interpolation est ensuite effectuée entre les différentes valeurs estimées afin d'obtenir une carte de  $20 \times 20$  coefficients  $K_s$  et  $\sigma$  par triangle (le coefficient  $K_s$  est identique pour les trois composantes couleurs).

K. Nishino et al. ont également utilisé le modèle simplifié de Torrance-Sparrow pour estimer les propriétés de réflectance d'un objet mais les conditions d'illumination ne sont pas connues [NZI01]. En faisant l'hypothèse que les surfaces de l'objet contiennent une composante de réflexion spéculaire, les sources lumineuses sont détectées à partir des halos spéculaires visibles depuis certains points de vue. Une carte de texture de  $20 \times 20$  pixels est calculée pour chaque triangle du maillage, correspondant à la composante de réflexion diffuse émise par la surface (sans suppression de l'illumination, différente de la carte d'albédo). Elle est obtenue par projection de chaque image sur le modèle géométrique et en conservant, en chaque point de la texture, la plus petite valeur de luminance projetée afin de supprimer la composante spéculaire. Des images *résiduelles* sont calculées en supprimant de chaque image d'origine la composante de réflexion diffuse stockée dans les cartes de textures. Les halos spéculaires, facilement détectés sur les images résiduelles, sont utilisés pour estimer un ensemble de sources hémisphériques (sources lumineuses directionnelles représentées sur une hémisphère englobant l'objet). Les coefficients spéculaires  $K_s$  et  $\sigma$  sont supposés constants sur toute la surface de l'objet.

Y. Yu et al. utilisent des images à fortes variations de luminance pour estimer les propriétés de réflectance d'une scène à partir du modèle anisotrope de Ward [YDMH99]. Les images HDR offrent une plus grande précision des valeurs de luminance captées dans la scène et permettent ainsi de prendre en compte les interréflections lumineuses, de préserver les valeurs correspondant aux halos spéculaires et d'estimer de l'anisotropie. Une technique de *radiosité inverse* est utilisée entre les faces du modèle géométrique afin de retrouver les paramètres du modèle de réflectance sur chaque face ( $K_d$ ,  $K_s$  et le coefficient de rugosité de la face). Un algorithme itératif permet d'évaluer soit la radiosité sur chaque face de la scène avec des paramètres de réflectance fixes, soit les paramètres de réflectance sur chaque face avec des valeurs de radiosité fixes. Pour augmenter la précision des informations de texture, une carte de coefficients de réflexion diffuse ( $K_d$ ) est estimée sur chaque face en calculant, pour chaque point de la carte, une moyenne des informations de luminance provenant de directions hors de la direction de réflexion spéculaire et en réévaluant la valeur de  $K_d$  à partir des conditions d'illumination connues. Les coefficients de réflexion spéculaire restent estimés globalement sur chaque face mais peuvent être à nouveau estimés en utilisant des images résiduelles (comme celles évoquées précédemment dans [NZI01]).

H. Lensch et al. proposent une méthode d'estimation de propriétés de réflectance uniforme par régions où l'utilisateur doit spécifier le nombre de BRDFs différentes sur la surface de l'objet [LKG<sup>+</sup>03]. Le maillage de l'objet est obtenu à partir d'un scanner à lumière structurée et la

source lumineuse ponctuelle est estimée à partir de trois sphères miroir introduites dans le système d'acquisition. Un algorithme itératif sépare la surface en plusieurs régions de réflectance uniforme (algorithme des k-means utilisé) puis estime les coefficients de la BRDF sur chaque région avec le modèle de Lafortune : le processus est réitéré jusqu'à convergence.

### D.5 Discussions

L'acquisition de BRDFs, présentée dans la section D.3, demande un calibrage très précis des paramètres géométriques (position, orientation de la caméra, de la source lumineuse et de la surface à acquérir) et des paramètres radiométriques (luminance constante émise par la source lumineuse, récepteur calibré de la caméra). Les phases d'acquisition sont très longues et peu envisageables sans un équipement motorisé : une quarantaine d'heures pour l'acquisition d'un seul point de vue.

Les BRDFs uniformes acquises sont enregistrées dans une carte de réflectance 4D (2 angles de direction d'incidence, 2 angles de direction de réflexion) afin d'être facilement réutilisables mais ne peuvent être utilisées pour représenter des objets texturés ou à matériaux variés.

L'acquisition de BRDFs variées sur la surface d'un objet utilise généralement un champ de réflectance pour une représentation efficace des données. Les données sont très volumineuses car une BRDF complète est représentée en chaque point de la surface de l'objet. Pour restreindre la quantité de données acquises, l'échantillonnage n'est effectué que sur un seul point de vue : cette limitation n'est pas envisageable si nous voulons visualiser notre objet rééclairé depuis des points de vue différents.

Nous nous sommes plutôt intéressés à l'autre approche, présentée dans la section D.4, consistant à estimer un modèle paramétrique de BRDF afin d'approcher le plus possible la BRDF réelle de la surface à acquérir. En effet, l'utilisation d'un modèle paramétrique est très intéressante car ce modèle permet de réduire considérablement la quantité de données stockées par rapport aux BRDFs acquises. Par exemple, une carte de réflectance 4D échantillonnée sur chaque dimension par 32 valeurs d'angle requiert  $32^4$  soit 1 million d'échantillons alors qu'un modèle paramétrique nécessite au maximum 5 coefficients. Le constat est identique pour un champ de réflectance à 6 dimensions : il peut être vu comme une texture 2D plaquée sur la surface de l'objet où en chaque texel est stocké soit une carte de réflectance 4D, soit quelques coefficients d'un modèle paramétrique.

Comme nous voulons acquérir des objets avec des réflectances variées sur la surface, nous avons regardé en détails les méthodes s'approchant à nos besoins et avons extrait les principales caractéristiques de ce groupe de méthodes.

En général, le modèle géométrique de l'objet et les caractéristiques des sources lumineuses sont utilisés pour estimer les coefficients de réflexions diffuse et spéculaire. Dans tous les cas, de nombreuses images sont nécessaires à l'estimation des paramètres de la BRDF avec quelquefois des images prises dans des positions spécifiques afin d'avoir des halos spéculaires bien visibles sur une série d'images. Les coefficients sont toujours estimés par régression non linéaire en évaluant l'erreur entre les échantillons de luminance provenant des images et la luminance recalculée par le modèle paramétrique.

Beaucoup de méthodes proposent de séparer la composante de réflexion diffuse de la composante de réflexion spéculaire sur chacune des images afin de mieux contrôler les variations de la spécularité sur la surface de l'objet et de générer une texture précise de toute la surface. La texture représente en chaque point de la surface le coefficient de réflexion diffuse de chaque



composante colorimétrique  $(R, G, B)$ . Cette texture est souvent appelée carte d'albedo car elle est indépendante du point de vue et indépendante des conditions d'illumination. Les coefficients de réflexion spéculaire ne sont généralement pas estimés en chaque point de la surface mais en chaque triangle du modèle géométrique car une estimation précise de la specularité nécessite un nombre considérable d'images pour obtenir suffisamment d'informations dans les directions de réflexion spéculaire. Le coefficient de specularité ou exposant de réflexion spéculaire (suivant le modèle utilisé) est difficile à obtenir précisément sans images à fortes variations de luminance où les halos spéculaires apparaissent sans saturation de luminance.

Toutes ces méthodes négligent les interréllections de l'objet sur lui-même malgré la connaissance de la géométrie et des caractéristiques des sources lumineuses. Seuls Y. Yu et al. proposent une méthode d'estimation des propriétés de réflectance avec prise en compte des interréllections en évaluant la radiativité de chaque face. Cette estimation est possible uniquement parce que la géométrie de la scène, les caractéristiques des sources lumineuses et les valeurs de luminance sont connues avec précision (utilisation d'images HDR). Cette méthode nécessite l'acquisition de points de vue spécifiques afin qu'un halo spéculaire soit visible sur chaque face de la scène.

Il existe d'autres méthodes estimant les propriétés de réflexions diffuse et spéculaire par décomposition en harmoniques sphériques [WAT92, RH01]. Ces méthodes donnent des résultats similaires concernant la qualité de l'estimation de la BRDF mais nécessitent souvent plus de coefficients que les modèles paramétriques : 9 coefficients de fonctions sont au minimum requis pour une décomposition en harmonique sphérique.

Nous avons donc porté notre choix sur une estimation de la réflectance par un modèle de BRDF paramétrique en utilisant une méthode de régression non linéaire proche de celle de Levenberg - Marquardt.

## E Rééclairage d'objets réels

Le rééclairage basé-image, appelé en anglais *Image-Based ReLighting (IBRL)*, est une extension récente du rendu basé-image (*Image-Based Rendering, IBR*) au rééclairage. Le rendu basé-image, présenté en section A, est une représentation d'un objet ou d'un environnement par une série d'images de différents points de vue. La méthode de rendu basé-image la plus populaire est le champ de lumière, permettant de synthétiser des images de l'objet depuis n'importe quel point de vue. Comme les images du champ de lumière sont acquises sous des conditions d'illumination fixes, elles ne permettent pas de rééclairer directement les objets. L'augmentation constante des performances de calcul et des capacités de stockage du matériel informatique permet aujourd'hui de proposer des modèles géométriques et photométriques capables d'effectuer un rendu réaliste d'objets depuis n'importe quel point de vue avec variation arbitraire des conditions d'illumination : c'est l'apparition du rééclairage d'objets réels.

### E.1 Utilisation de BRDFs réelles acquises

L'acquisition de BRDF d'objets réels est la méthode la plus efficace pour obtenir un rendu réaliste lors du rééclairage. Mais cette méthode nécessite l'acquisition d'un nombre colossal d'images avec des conditions d'illumination et des points de vue variés afin d'échantillonner le mieux possible les quatre dimensions de la BRDF.

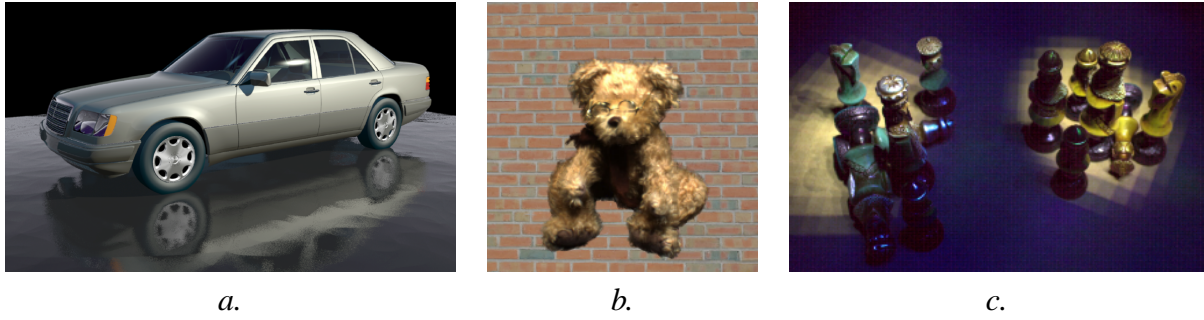


Figure 1.31 – a. Rééclairage d’une voiture à partir d’un modèle géométrique et d’une BRDF uniforme sur la surface (image extraite de [LFTW06]); b. et c. rééclairage d’objets basés-image à partir d’un champ de réflectance (images extraites de [MPN<sup>+</sup>02, MPDW03]).

Les cartes de réflectance 4D couplées avec une représentation géométrique de l’objet permettent un rééclairage sous toute condition d’illumination et depuis n’importe quel point de vue en estimant en chaque pixel la luminance émise par le point  $x$  correspondant sur la surface de l’objet. Cette estimation peut être effectuée par lancer de rayon en évaluant l’équation de réflectance sur l’ensemble des sources lumineuses (définie en section C de l’annexe I) en chaque point  $x$  :

$$L_r(x, \vec{R}) = \sum_{s=1}^{\#Sources\ visibles} \left( f_r(x, \vec{R}, \vec{I}_s) \cdot L_i(x, \vec{I}_s) \cdot \cos \theta_s \cdot \frac{1}{r_s^2} \right) \quad (1.14)$$

avec  $\vec{I}_s$  et  $r_s$  la direction et la distance de la source  $s$  au point  $x$  ;  $\theta_s$  l’angle d’incidence formé avec la source  $s$  au point  $x$  ;  $L_i(x, \vec{I}_s)$  la luminance reçue en  $x$  provenant de la source  $s$  et  $\vec{R}$  la direction du pixel depuis  $x$ .  $f_r(x, \vec{R}, \vec{I}_s)$  est la BRDF évaluée au point  $x$ . Son évaluation est très rapide, il suffit de récupérer la valeur correspondant aux coordonnées polaires de  $\vec{I}_s$  et  $\vec{R}$  dans la carte de réflectance 4D (la normale en  $x$  est requise pour évaluer les angles). Si cette valeur n’existe pas, les coordonnées polaires sont arrondies au plus proche échantillon stocké dans la carte de réflectance.

Ce modèle nécessite le stockage d’une BRDF par région de réflectance uniforme, il est plutôt utilisé pour l’affichage d’objets de réflectance uniforme mais complexe à approcher par un modèle paramétrique : par exemple, sur la figure 1.31.a, la carrosserie de la voiture est rééclairée avec une BRDF de peinture métallisée acquise par un gonioréfectomètre.

Les champs de réflectance sont plus intéressants car ils représentent des BRDFs variant sur la surface des objets et permettent un rééclairage de l’objet sous toute condition d’illumination et depuis n’importe quel point de vue. Dans [MPN<sup>+</sup>02], le champ de réflectance est complété par un champ de transparence permettant d’enregistrer l’opacité d’objets complexes comme les peluches, les fourrures, etc. Cette opacité est obtenue en utilisant des écrans plasma placés derrière l’objet à photographier. Le rééclairage d’un objet avec de nouvelles conditions d’illumination est effectué en deux étapes. Premièrement, un champ de lumière est précalculé à partir du champ de réflectance en sommant, en chaque pixel de chaque point de vue, la luminance émise par l’objet vers ce pixel (valeurs extraites du champ de réflectance pour chaque source lumineuse). Evidemment, les nouvelles sources lumineuses ne sont pas toujours positionnées au même endroit que les sources originales et une pondération est effectuée sur chaque source originale en fonction de l’angle solide couvert : l’objet peut ainsi être rééclairé avec des sources surfaciques.

Deuxièmement, le champ de luminance construit peut être affiché interactivement en n'importe quel point de vue par une technique traditionnelle de rendu basé-image. L'enveloppe visuelle de l'objet est préalablement reconstruite afin d'améliorer l'interpolation entre plusieurs points de vue et l'application du champ de transparence permet un rendu plus réaliste des surfaces complexes. Le faible nombre de positions différentes de source lumineuse (60) prises avec ce système d'acquisition ne permet pas de réafficher précisément les halos spéculaires lors du rééclairage : l'interpolation entre les luminances provenant des différentes sources originales atténue les effets hautement spéculaires (figure 1.31.b). De plus, ce système permet uniquement un rééclairage avec des sources lumineuses directionnelles : sans utilisation de la géométrie de l'objet, l'éclairage ne peut être que directionnel.

Dans [MPDW03], V. Masselus et al. effectuent une acquisition plus précise du champ de réflectance : une seule vue est disponible mais pour chaque pixel de cette vue, un champ de lumière incident (4 dimensions) est acquis en utilisant un faisceau lumineux. Cette acquisition permet le rééclairage d'objet vu depuis un point de vue fixe mais en faisant varier spatialement les conditions d'illumination. En effet, le positionnement de nouvelles sources ponctuelles permet de simuler l'atténuation due à la distance source - objet sans aucune définition de la géométrie de l'objet. Comme précédemment, la nouvelle image est synthétisée par combinaison d'images stockées dans le champ de réflectance. Cependant le stockage d'un champ de lumière incident par pixel entraîne une double somme lors de l'évaluation de la luminance émise par chaque pixel : une somme sur l'ensemble des sources lumineuses originales (224) et pour chaque source, une somme sur l'ensemble des parties de surfaces illuminées correspondant à une direction de source (256). Cette double somme entraîne souvent des artéfacts visibles sur les nouvelles images produites (figure 1.31.c).

Les systèmes d'acquisition actuels permettent rarement l'acquisition d'un champ de réflectance en temps raisonnable. Aucune méthode n'est encore capable d'acquérir les 8 dimensions complètes du champ de réflectance. De plus, le stockage de multiples points de vue est problématique malgré l'application de méthodes de compression : V. Masselus annonce un stockage de 6 giga-octets de données compressées pour un seul point de vue. Par conséquent, les champs de réflectance sont encore limités à l'utilisation de quelques points de vue.

## E.2 Utilisation de modèles de BRDFs paramétriques

L'estimation des BRDFs par un modèle paramétrique est une solution plus raisonnable en terme de coût mémoire (voir section D.5 pour la comparaison) et elle permet un rééclairage réaliste avec des conditions d'illumination variées en tout point de vue.

Pour obtenir des BRDFs variant à la surface de l'objet, un ensemble de coefficients est estimé en chaque point d'un modèle géométrique précis de l'objet : l'évaluation du vecteur normal en chaque point de la surface est indispensable afin d'estimer précisément les coefficients. Généralement une carte d'albédo est stockée sur la surface de l'objet : elle représente les informations de texture de la surface pour chaque triangle du maillage géométrique avec des coefficients de réflexion diffuse dans chaque texel. Suivant le raffinement du modèle géométrique, entre  $20 \times 20$  et  $80 \times 80$  texels sont stockés par triangle [SWI97, YDMH99]. Les coefficients de réflexion spéculaire sont évalués moins précisément avec en moyenne une estimation de coefficients spéculaires par triangle. L'utilisation d'un maillage triangulaire est généralement la forme géométrique utilisée pour permettre un rééclairage efficace en chaque point du modèle. Nous allons voir à présent comment les données peuvent être utilisées de manière efficace pour rééclairer les objets.



Dans [SWI97], Y. Sato et al. estiment également une carte de normale et une carte de coefficients spéculaires par triangle en utilisant l'interpolation de Gouraud/Phong [Gou71] : évaluation de la normale et des coefficients de spécularité en chaque sommet de triangle puis recalcul de ces valeurs en chaque texel par moyenne pondérée entre les trois sommets. La synthèse d'une image avec de nouvelles conditions d'illumination est simplifiée, elle est similaire aux BRDFs stockées dans une carte de réflectance 4D. Un lancer de rayons est utilisé afin d'estimer la luminance en chaque pixel de la nouvelle image : pour cela sont évalués le point d'intersection 3D avec le triangle le plus proche et le texel 2D correspondant dans la carte de texture du triangle intersecté. L'objet peut ainsi être rééclairé avec de nouvelles sources lumineuses ponctuelles et prise en compte de l'auto-ombrage sur sa surface. Le calcul des interrélflexions lumineuses de l'objet sur lui-même peut également être évalué par un lancer de rayon récursif, en calculant en chaque point d'intersection un terme supplémentaire dans la direction miroir du point de vue synthétisé.

Avec un ensemble de cartes représentant les informations de géométrie et de réflectance de chaque triangle, nous pouvons accélérer les calculs sans passer par un traditionnel lancer de rayon. La texture à afficher peut être évaluée pour chaque triangle visible depuis la nouvelle vue et à envoyée directement à la carte graphique sous forme de triangle texturé sans effectuer un calcul pixel à pixel sur l'image finale. Le tampon de profondeur de la carte graphique trie les triangles reçus et affiche les pixels les plus près sur l'image finale, quel que soit l'ordre de réception des triangles par la carte. Pour évaluer rapidement la texture de chaque triangle, un calcul matriciel peut être effectué entre les différentes cartes stockées dans un triangle sans évaluer le point d'intersection 3D de chaque texel sur le modèle géométrique. Ceci suppose que l'objet est illuminé par un ensemble de sources directionnelles. De plus, chaque triangle projeté sur l'image doit être petit comparé à la taille de l'image afin de pouvoir faire l'hypothèse que la direction de vue est constante en tout point d'un triangle : ceci suppose qu'un objet ne peut plus être regardé de trop près. Un dernier problème reste à résoudre, le plus difficile : comment connaître la liste des sources visibles en chaque texel de la surface. C'est une information nécessaire à l'évaluation de l'auto-ombrage sur les surfaces de l'objet.

La méthode développée par P. Sloan et al. dans [SKS02] peut être appliquée pour évaluer l'auto-ombrage. Une carte d'illumination est utilisée afin de stocker la luminance incidente depuis une hémisphère englobant l'objet. De plus, une carte de visibilité est précalculée en chaque texel du modèle géométrique afin de stocker les directions d'auto-ombrage sur une hémisphère : cette carte est coûteuse en mémoire car elle représente une information 4D sur chaque triangle. Lors du calcul de la luminance émise par chaque texel, le produit de la carte d'illumination par la carte de visibilité représente la carte d'illumination de chaque texel, incluant l'auto-ombrage. Comme la carte de visibilité ajoute beaucoup d'informations par texel comparée à la BRDF paramétrique, P. Sloan et al. utilisent une BRDF uniforme sur toute la surface de l'objet. De plus, les cartes d'illumination et de visibilité sont représentées par une décomposition en harmonique sphérique (entre 9 et 25 coefficients sont utilisés par carte). Pour accélérer les calculs, la BRDF uniforme est également représentée par des coefficients d'harmonique sphérique : 9 coefficients pour la composante diffuse et 25 coefficients pour la composante spéculaire. Ainsi le produit des cartes de coefficients est effectué par opérations matricielles sur le matériel graphique en plusieurs passes afin d'obtenir un rendu de plusieurs images par seconde. Le principal défaut des harmoniques sphériques est leur incapacité à représenter des surfaces fortement spéculaires (où le halo spéculaire visible sur l'image finale doit être très étroit). Pour des conditions d'éclairage fixées, P. Sloan et al. proposent également de calculer une carte d'auto-interrélflexion sur chaque

texel, stockant la luminance diffuse émise vers un texel par les autres surfaces de l'objet. La carte d'auto-interréflexion est également encodée par des coefficients d'harmoniques sphériques et est utilisée lors du calcul de la luminance émise par un texel afin de récupérer la luminance incidente depuis la direction miroir du point de vue.

D. McAllister et al. proposent d'effectuer le rendu interactif d'une scène ou d'un objet complexe en stockant, comme dans [SWI97], les coefficients des BRDFs de chaque polygone du modèle géométrique dans une texture de taille minimale [MLH02]. Chaque texel contient l'ensemble des coefficients du modèle de Lafortune, BRDF paramétrique exprimée avec deux ou trois lobes. Plusieurs cartes de texture sont utilisées par polygone car un texel ne peut contenir que 4 coefficients (sur 4 octets) et le modèle de Lafortune nécessite 7 coefficients par lobe ( $K_s$  selon les trois composantes chromatiques,  $C_x$ ,  $C_y$ ,  $C_z$  et  $n$ ). L'ensemble des textures de tous les polygones est appelé une *BRDF Spatiale (SBRDF)* par les auteurs. La luminance émise vers un pixel de la nouvelle image synthétisée est évaluée par le matériel graphique en deux passes pour chaque lobe de Lafortune et chaque source lumineuse présente dans l'environnement (la méthode est similaire à celle de [SKS02]). Pour accélérer les calculs, une texture supplémentaire de  $256 \times 256$  valeurs est utilisée afin de stocker une table de valeurs d'exponentiation précalculées :  $x^n$  avec  $x \in [0, 1]$  et  $n \geq 0$ . Dans le cas de l'utilisation de sources lumineuses directionnelles, une carte d'environnement indexée par la direction du vecteur normal  $\vec{N}$  est également précalculée afin de stocker les valeurs de  $L_i(x, \vec{I}_s) \cdot (\vec{N} \cdot \vec{I}_s)$  pour chaque source  $s$ .

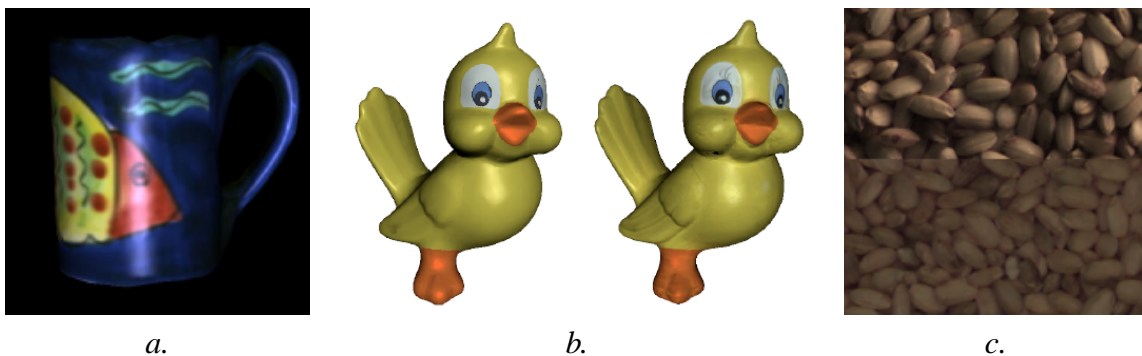


Figure 1.32 – a. La partie spéculaire de la surface rééclairée, évaluée à partir d'une BRDF paramétrique, est plus large que sur la surface réelle (image extraite de [SWI97]); b. à gauche, rééclairage avec une BRDF paramétrique pour chaque surface de réflectance uniforme : la surface paraît trop lisse, peu réaliste ; à droite, rééclairage avec une BRDF paramétrique pour chaque triangle du modèle géométrique : les détails de texture sont conservés (images extraites de [LKG<sup>+</sup>03]); c. textures éclairées avec une source lumineuse rasante ; en bas : texture conventionnelle ; en haut : texture polynomiale (image extraite de [MGW01]).

Même si les BRDFs paramétriques estimées sont très proches des vraies BRDFs des objets, de petits défauts visuels subsistent, notamment sur la précision des halos spéculaires affichés. Le manque de points de vue pour estimer les coefficients des modèles paramétriques entraîne souvent un rééclairage approximatif des surfaces situées dans la direction de réflexion miroir. (figure 1.32.a).

Dans [LKG<sup>+</sup>03], H. Lensch et al. utilisent également le modèle de Lafortune pour estimer la réflectance d'objets réels. Utilisant des objets de réflectance sensiblement uniforme sur de grandes étendues de la surface, la surface de l'objet est subdivisée en plusieurs régions de propriétés photométriques différentes entre elles et les coefficients du modèle de BRDF sont estimés

sur chaque région. Lors du rendu de la surface de l'objet, le faible nombre de BRDFs différentes donne un aspect lisse et artificiel à la surface de l'objet (figure 1.32.b). Pour remédier à cela, la réflectance de chaque région précédemment définie est représentée par une base de BRDFs paramétriques et chaque triangle du modèle géométrique dispose de sa BRDF, formée par pondération des BRDFs de la région dont le triangle appartient.

T. Malzbender et al. ont proposé les *textures polynomiales* afin de représenter des surfaces non parfaitement planes par un ensemble de 6 coefficients par texel et par composante chromatique [MGW01]. Ces coefficients correspondent à une fonction biquadratique polynomiale approchant les propriétés de réflectance de la surface. Cette fonction évalue la luminance émise par chaque texel en fonction de la direction d'incidence. En utilisant une texture stockant la direction d'incidence en chaque texel de la surface, la texture polynomiale est évaluée très rapidement par des opérations de texture programmables sur la carte graphique. Les 6 coefficients sont estimés en chaque texel à partir d'un ensemble de 48 images de la surface illuminée par une source lumineuse prenant des positions variées. Les surfaces rééclairées donnent une impression de relief mais ne permettent pas de générer des textures dépendantes du point de vue. Les textures rééclairées dépendent uniquement des conditions d'éclairage mais elles permettent néanmoins l'auto-ombrage (figure 1.32.c). Elles sont donc utilisées pour générer n'importe quel point de vue de surfaces non planes d'aspect diffus ou pour générer un point de vue fixe de surfaces à composante spéculaire (en utilisant le même point de vue que celui des images ayant servi à paramétrer le modèle).

### E.3 Discussions

La finalité du rééclairage d'un objet réel est son intégration réaliste dans une scène avec prise en compte des conditions d'illumination de cette scène sur l'objet. Ceci suppose donc une modification des conditions d'illumination initialement présentes sur la surface de l'objet réel et une visualisation de l'objet depuis n'importe quelle direction de vue.

Afin que les nouvelles vues synthétisées semblent réalistes, l'intégration suppose la prise en compte précise des échanges lumineux entre l'objet et la scène. L'éclairage de la scène avec ses propres sources lumineuses doit tenir compte des ombres de l'objet réel projetées sur les objets de la scène. Réciproquement, le rééclairage de l'objet réel avec les conditions d'illumination de la scène doit tenir compte des ombres de la scène projetées sur l'objet réel. Une intégration réaliste doit utiliser un algorithme d'illumination globale comme un lancer de rayons récursif ou un calcul de radiosit  afin d'évaluer les interr flexions lumineuses entre la scène et l'objet. Ces algorithmes sont utilisables uniquement lorsque les modèles géométriques de la scène et de l'objet réel sont connus.

Un rééclairage réaliste de l'objet suppose également un calcul de l'auto-ombrage et un calcul des auto-interr flexions lumineuses sur les objets insérés : les sections qui suivent montrent comment les calculs d'ombrage et d'interr flexions peuvent  tre r alis s suivant les modèles de représentation de la réflectance utilis s (cartes de réflectance, champs de réflectance, BRDFs param triques). Nous  tudions  galement la pertinence de l'utilisation d'un mod le g om trique compl mentaire.

### E.3.1 Intégration dans des scènes virtuelles

Une scène virtuelle est peu contraignante pour l'insertion d'objets réels car elle est définie par un modèle géométrique, des propriétés de réflectance connues pour tous les objets qui la composent et des conditions d'illumination précises (nombre fixé de sources ponctuelles, surfaciques, etc).

Lorsque l'objet est représenté par des cartes de réflectance 4D, un modèle géométrique de l'objet est fourni afin de pouvoir évaluer la luminance émise par chaque point du modèle. Par conséquent tous les échanges lumineux entre la scène et l'objet peuvent être évalués mais l'apparence d'une réflectance uniforme sur toute la surface de l'objet réel peut nuire à une intégration réaliste : la surface paraît généralement trop lisse, trop régulière donc peu réaliste. Nous ne voulons pas utiliser ce type de rendu.

Un objet représenté par un champ de réflectance ne dispose pas de modèle géométrique. Par conséquent, son intégration est plus problématique. L'évaluation de l'auto-ombrage est toujours possible puisqu'elle est stockée dans le champ de réflectance pour chaque direction d'incidence. En revanche, l'évaluation des auto-interréflexions n'est pas toujours possible. En effet, si un champ de réflectance complet est utilisé, comme dans [MPDW03] avec plusieurs points de vue acquis, alors l'évaluation d'une auto-interréflexion en un pixel demande l'évaluation de la luminance incidence dans la direction miroir de la direction de vue : or la normale à la surface de l'objet en ce pixel n'est pas connue, ce qui interdit le calcul de la direction miroir. Dans [MPN<sup>+</sup>02], le champ de réflectance n'est défini que sur 6 dimensions en effectuant une acquisition avec une source directionnelle. Par conséquent, les auto-interréflexions sont déjà enregistrées dans le champ de lumière et peuvent être extraites pour tout point de vue. Néanmoins, le rééclairage est limité à l'utilisation de nouvelles sources directionnelles sinon des erreurs d'illumination apparaissent à la surface de l'objet (les auto-interréflexions sont incorrectes).

Les interréflexions diffuses et les ombres projetées de l'objet vers la scène sont évaluées directement à partir du champ de réflectance. En un point de la scène, chaque rayon en direction d'une source lumineuse évalue l'ombre projetée par l'objet réel en traversant le champ de lumière : si le rayon traverse la silhouette de l'objet alors le point de la scène est à l'ombre de l'objet pour la source lumineuse testée. De même, les interréflexions en provenance de l'objet réel depuis un point de la scène sont évaluées en lançant un rayon dans la direction miroir de la direction de vue et en récupérant la valeur de luminance en provenance du champ de réflectance.

En revanche, les interréflexions diffusées de la scène vers l'objet réel ne peuvent pas être prises en considération car il est impossible d'évaluer la position et la direction de réflexion miroir par rapport au point de vue sur l'objet réel sans modèle géométrique : il faudrait ajouter l'information de profondeur et d'orientation de la normale en chaque direction du champ de réflectance. Enfin, concernant les ombres de la scène projetées sur l'objet, elles ne peuvent être intégrées que dans le cas d'un champ de réflectance complet en restreignant l'utilisation du champ de luminance incident aux directions d'incidence non interceptées par les objets de la scène : ce calcul est compliqué, il demande un traitement sur les 4 dimensions du champ de lumière incident.

Les champs de réflectance ne permettent donc pas d'évaluer tous les calculs d'illumination globale malgré un échantillonnage dense sur 8 dimensions. Ces limitations peuvent être supprimées en définissant des champs de réflectance paramétrés sur la surface, comme les champs de lumière paramétrés sur la surface afin de compenser le manque d'information par un modèle géométrique (voir section A.2.3). Ils peuvent également être complétés par des informations de profondeur et de normale en plus des informations de luminance. Le système de représentation

est peu viable actuellement, vu la quantité d'information devant être stockée par objet. Le champ de réflectance est très rapide pour un rééclairage avec un calcul d'illumination locale mais son intégration dans un environnement virtuel avec un calcul d'illumination globale nécessite autant de traitements que l'utilisation d'un modèle de BRDF paramétrique.

Les représentations par un modèle paramétrique semblent un meilleur compromis car les coefficients des modèles paramétriques stockés dans des textures plaquées sur le maillage nécessitent un stockage beaucoup moins volumineux que les champs de réflectance et des temps de calculs d'illumination globale similaires. L'utilisation d'un modèle géométrique permet le calcul de tous les transferts lumineux entre la scène et l'objet. De nombreuses optimisations sont proposées pour accélérer les calculs d'illumination en utilisant notamment la puissance du matériel graphique (voir section E.2 pour plus de détails). Néanmoins, il n'est pas encore possible d'afficher des objets réels rééclairés dynamiquement avec prise en compte de propriétés spéculaires sans passer par des précalculs utilisant des conditions d'illumination préfixées.

### E.3.2 Intégration dans des scènes réelles

La réalité augmentée est un vaste domaine travaillant sur l'insertion d'objets virtuels dans des scènes réelles. Les nombreux travaux ont montré qu'une insertion réaliste d'objets devait utiliser un modèle géométrique de la scène réelle afin d'effectuer les calculs de transfert d'énergie entre la scène et les objets insérés.

Les informations sur les conditions d'illumination doivent également être estimées afin de permettre une insertion des objets réels avec des méthodes similaires à celles proposées dans la section précédente. P. Debevec a proposé une capture d'illumination de l'environnement réel avec l'insertion d'une sphère miroir dans l'environnement près de l'endroit d'insertion d'un objet [Deb98]. La carte d'illumination 2D obtenue est ensuite utilisée pour intégrer des objets virtuels dans l'environnement en prenant en considération les conditions d'éclairage de la scène. La méthode peut être réutilisée pour insérer des objets réels. Toutefois, l'utilisation d'une carte d'illumination 2D oblige un éclairage directionnel sur l'objet à rééclairer : celui-ci doit être de taille négligeable comparée à celle de l'environnement afin que l'insertion semble réaliste.

# NOTRE SYSTÈME DE RECONSTRUCTION GÉOMÉTRIQUE

---

## Sommaire

---

<b>A</b>	<b>Acquisition . . . . .</b>	<b>60</b>
A.1	Choix du matériel . . . . .	60
A.2	Système d'acquisition . . . . .	61
A.3	Construction de lumigraphes synthétiques . . . . .	74
<b>B</b>	<b>Reconstruction de la forme . . . . .</b>	<b>75</b>
B.1	Choix de la méthode . . . . .	76
B.2	Reconstruction basée-voxel . . . . .	77
B.3	Construction de la surface discrète . . . . .	84
B.4	Amincissement de la surface discrète . . . . .	89
B.5	Génération de maillage fermé . . . . .	93
B.6	Utilisation des pixels pour affiner le maillage . . . . .	100
<b>C</b>	<b>Estimation de la normale . . . . .</b>	<b>108</b>
C.1	Choix de la représentation . . . . .	109
C.2	Normales à partir du maillage . . . . .	109
C.3	Normales à partir de la surface discrète . . . . .	111
C.4	Analyse des résultats . . . . .	112
<b>D</b>	<b>Résultats . . . . .</b>	<b>114</b>
D.1	Statistiques sur les rayons projetés . . . . .	114
D.2	Occupation mémoire . . . . .	116
D.3	Complexité des algorithmes . . . . .	117
D.4	Temps de calcul . . . . .	118
D.5	Objets synthétiques reconstruits . . . . .	119
D.6	Objets réels reconstruits . . . . .	119

---



L'état de l'art sur la reconstruction géométrique (section B du chapitre I) montre qu'il existe deux grandes familles de méthodes : (1) les méthodes actives utilisant un scanner 3D pour acquérir directement la géométrie de l'objet et (2) les méthodes passives utilisant les informations contenues dans des images pour reconstruire un modèle géométrique précis de l'objet. Ne disposant pas de matériel d'acquisition spécifique au sein du laboratoire, notre approche est orientée sur la seconde famille de méthodes. Nous définissons un système d'acquisition simple mais robuste car la précision des méthodes d'estimation (de la géométrie, des BRDFs ou des conditions d'illumination) repose toujours sur un fort calibrage de l'ensemble des images. L'estimation du modèle géométrique est effectuée par une méthode utilisant l'enveloppe visuelle car c'est la seule méthode de reconstruction permettant d'utiliser tous types de matériaux, avec des propriétés de réflectances variées et sans la connaissance des conditions d'illumination.

## A Acquisition

La reconstruction géométrique d'un objet représenté uniquement par des images nécessite l'acquisition d'images représentant cet objet sous divers points de vue. De plus, comme les propriétés de réflectance de la surface doivent être estimées à partir des images, celles-ci doivent toutes être acquises avec les mêmes conditions d'illumination. Par conséquent, le protocole d'acquisition doit respecter certaines contraintes :

- plusieurs images acquises sous divers points de vue – l'ensemble des images doit couvrir toute la surface de l'objet pour une reconstruction complète de la géométrie par l'enveloppe visuelle ;
- mêmes conditions d'éclairage sur chacune des images – les sources lumineuses doivent être fixes par rapport à l'objet ;
- peu d'interréflexions lumineuses avec le reste de l'environnement afin de préserver les conditions d'éclairage – attention notamment au déplacement de la caméra : l'ombre de la caméra ne doit pas être projetée sur l'objet à photographier ;
- réduire au maximum les couleurs proches du noir sur la surface de l'objet – ceci facilite l'étape d'extraction de l'objet et de l'arrière-plan.

### A.1 Choix du matériel

Pour acquérir les données représentant un objet réel, nous ne voulons pas utiliser de matériel spécifique comme un scanner laser (acquisition de la géométrie) ou un gonioréfectomètre (acquisition de la BRDF). Nous utilisons uniquement un appareil photographique numérique afin de définir un système d'acquisition rudimentaire et facilement reproductible.

La mise en correspondance de l'ensemble des images nécessite la connaissance des caractéristiques intrinsèques (longueur de la focale, vitesse d'obturation, angle d'ouverture) et extrinsèques (position, direction, orientation) de la caméra pour chaque image acquise. Pour simplifier la mise en œuvre, les caractéristiques intrinsèques de la caméra sont fixées avant l'acquisition. De plus, une grande longueur de la focale est utilisée afin de limiter la déformation sur les bords de l'image (cette déformation est négligée dans la suite des travaux).

L'utilisation d'images HDR (à fortes variations de luminance [Deb98]) peut améliorer l'estimation des propriétés de réflectance (chapitre III) mais oblige à prendre plusieurs photographies avec des vitesses d'obturation différentes pour chaque prise de vue. Ceci ralentit considérablement

ment la phase d'acquisition et prend plus d'espace sur le disque. Par conséquent les images HDR ne sont pas utilisées dans ce travail mais peuvent toujours être intégrés par la suite.

## A.2 Système d'acquisition

L'acquisition des objets réels a été effectuée dans une salle aménagée de notre laboratoire. Les murs sont peints en noir mat afin d'éviter les interrélaxions lumineuses entre l'objet et l'environnement. L'appareil photo numérique est d'utilisation grand public, avec une longueur focale réglée sur  $105\text{mm}$ . Aucun système d'acquisition motorisé n'est disponible dans le laboratoire. Par conséquent l'acquisition est effectuée manuellement et un étalonnage de la caméra permet d'estimer précisément les paramètres extrinsèques lors de chaque prise de vue.

### A.2.1 Utilisation d'une mire

Pour effectuer l'acquisition le plus librement possible, nous avons choisi de reproduire un système d'acquisition similaire à celui des lumigraphes [GGSC96], i.e. fixer les sources lumineuses par rapport à l'objet, prendre une série de photos avec une caméra tenue à la main, retrouver les propriétés extrinsèques de la caméra en utilisant une mire et construire un lumigraphe en projetant les images acquises sur les différents *slabs*. L'acquisition éventuelle d'images supplémentaires permet de compléter le lumigraphe et d'obtenir un échantillonnage dense de l'objet.

La mire 3D utilisée dans [GGSC96] est trop contraignante car son déplacement est inévitable lors de l'acquisition d'images tout autour de l'objet : ceci entraîne une possible perte de précision sur l'estimation des paramètres de la caméra et sur l'acquisition de la réflectance de la surface de l'objet. L'acquisition de photographies supplémentaires demande de repositionner la mire à l'identique de la première acquisition, ce qui ajoute de nouveau du bruit dans le système de calibrage. De plus, le déplacement de la mire 3D crée et modifie les interrélaxions entre la surface de l'objet et la mire (la mire ne peut être noire car elle doit être détectée). Notons que dans [GGSC96], les auteurs utilisent la mire pour l'acquisition d'un seul slab et n'ont effectué aucune acquisition complète autour d'un objet réel.

Nous avons donc opté pour une mire 2D, plus facile à mettre en place, dans le même esprit que celle proposée par E. Guillou dans sa thèse [Gui00]. La mire 2D atténue les interrélaxions lumineuses entre la mire et l'objet, et n'a pas besoin d'être déplacée pendant la phase d'acquisition.

#### Principe

Cette mire est constituée de deux ensembles réguliers de segments orthogonaux représentant deux axes du repère global (figure 2.1.a). Sur la photographie, les segments n'apparaissent généralement pas parallèles entre eux (sauf pour des positions spécifiques de la caméra sur un des deux axes du repère global). En effet, tout ensemble de droites parallèles de l'espace 3D (les segments parallèles de la mire 2D) se projette, suivant un modèle de géométrie projective, en un ensemble de lignes 2D concourantes en un même point d'intersection : le *point de fuite*. Il représente la projection, sur le plan image, du point à l'infini de ces droites.

Nous avons développé un algorithme évaluant les deux points de fuite (coordonnées dans le repère de la caméra) à partir des deux ensembles de droites extraits sur la photographie (figure 2.1.b). Chaque droite est obtenue à partir de la mise en correspondance automatique de deux segments de l'image.



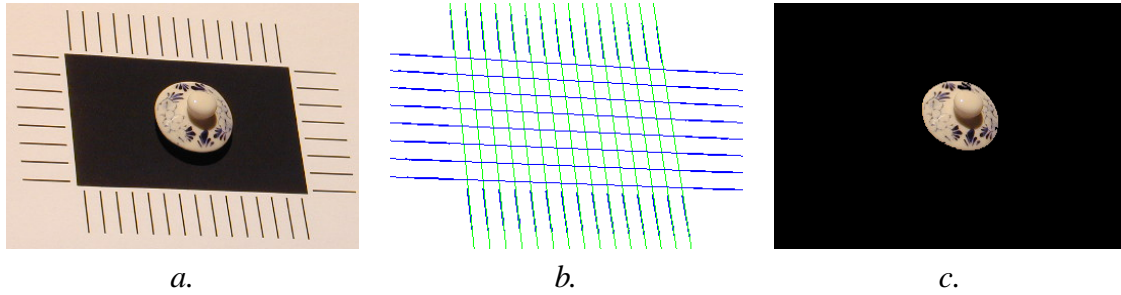


Figure 2.1 – a. Photographie d'un objet avec la mire 2D ; b. puis détection des deux ensembles de droites orthogonales ; c. et extraction de l'objet.

A partir de l'évaluation des deux points de fuite, les propriétés de la géométrie projective sont utilisées pour estimer la focale et les paramètres extrinsèques de la caméra. L'estimation de la valeur de la focale permet de valider la précision des calculs effectués.

### Mise en œuvre

Les segments ont été tracés en noir sur fond blanc pour faciliter leur détection. Un segment est représenté par un ensemble de pixels noirs 8-connexes sur l'image. Les coefficients de l'équation d'une droite sont estimés par un calcul de moindres carrés sur la distance séparant les pixels du segment et la droite. Chaque point de fuite est estimé par un calcul de matrice pseudo-inverse sur un ensemble d'équations de droites.

#### Estimation de la focale

Les deux points de fuite  $F_1$  et  $F_2$  sont situés sur le plan image : voir figure 2.2. Les coordonnées de tous les points situés sur la figure 2.2.b sont connues, exceptées celles du centre optique  $O$  : la valeur de la focale  $f$  est recalculée en supposant que l'image est acquise sur un film de  $24 \times 36mm$ . Les coordonnées des pixels de l'image peuvent donc être traduites en coordonnées métriques : une largeur de  $36mm$  pour l'image donne une mesure métrique précise du pixel.

A partir des deux points de fuite, la focale peut être estimée en utilisant des propriétés de la géométrie projective. Le centre optique  $O$  est projeté orthogonalement en  $O_{\Pi}$  sur le plan image  $\Pi$ ,  $O_{\Pi}$  représentant le centre de l'image. La focale  $f$  correspond à la distance  $OO_{\Pi}$  séparant le centre optique et l'image. Considérons  $O_{\Delta}$  le projeté orthogonal de  $O_{\Pi}$  sur la ligne de fuite  $\Delta$  reliant les deux points de fuite. Ce point  $O_{\Delta}$  est également le projeté orthogonal du centre optique  $O$  sur la droite  $\Delta$ .  $F_1$  et  $F_2$  étant deux points de fuite correspondant à des ensembles de droites orthogonales, ces deux points forment un angle droit avec le centre de projection  $O$  ( $\widehat{F_1OF_2} = \frac{\pi}{2}$ ).

En utilisant les relations sur les angles droits, nous pouvons estimer la distance séparant  $O$  et  $O_{\Delta}$  uniquement en fonction de points situés sur le plan image  $\Pi$  :

$$\begin{aligned}
 & \widehat{F_1OO_{\Delta}} = \widehat{OF_2O_{\Delta}} \\
 \Rightarrow & \tan(\widehat{F_1OO_{\Delta}}) = \tan(\widehat{OF_2O_{\Delta}}) \\
 \Rightarrow & \frac{F_1O_{\Delta}/OO_{\Delta}}{OO_{\Delta}} = \frac{OO_{\Delta}/F_2O_{\Delta}}{OO_{\Delta}} \\
 \Rightarrow & \frac{F_1O_{\Delta}}{OO_{\Delta}^2} = \frac{OO_{\Delta}}{F_2O_{\Delta}}
 \end{aligned} \tag{2.1}$$

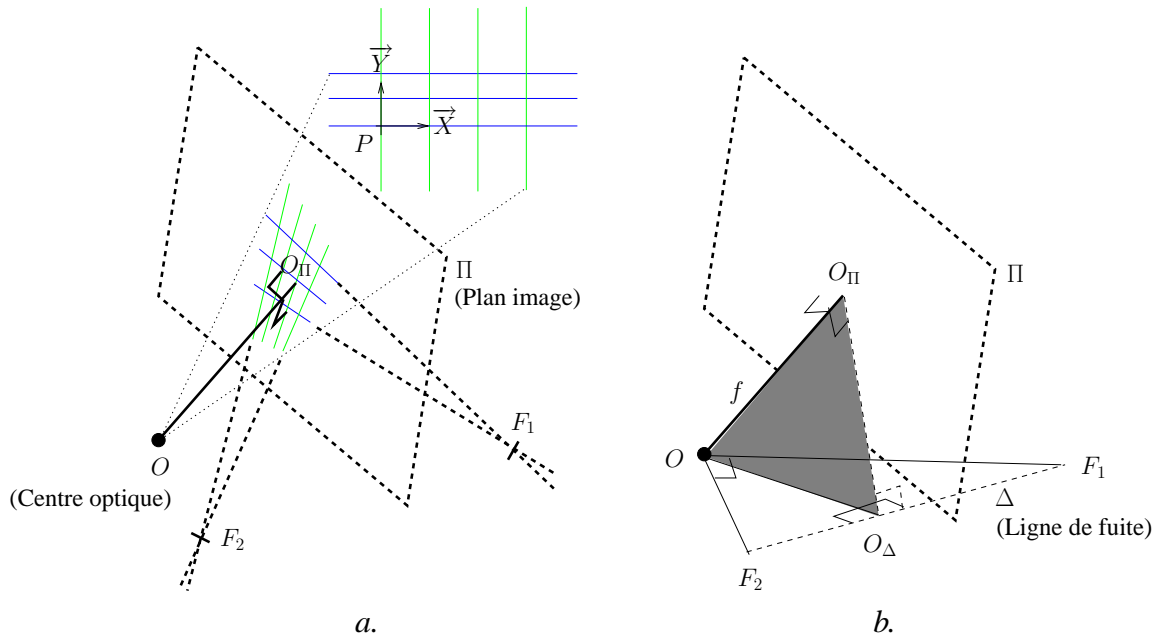


Figure 2.2 – a. Estimation des deux points de fuite  $F_1$  et  $F_2$  sur le plan image  $\Pi$  ; b. calcul de la focale  $f$  en utilisant les propriétés de la géométrie projective.

Une estimation de la focale est ainsi obtenue sans faire intervenir le point  $O$  :

$$\begin{aligned}
 f &= \overline{OO_{\Pi}} \\
 &= \sqrt{\overline{OO_{\Delta}}^2 - \overline{O_{\Pi}O_{\Delta}}^2} \\
 &= \sqrt{\overline{F_1O_{\Delta}} * \overline{F_2O_{\Delta}} - \overline{O_{\Pi}O_{\Delta}}^2}
 \end{aligned} \tag{2.2}$$

Cette estimation de la focale permet d'exprimer pleinement les coordonnées des points  $O$ ,  $O_{\Pi}$ ,  $O_{\Delta}$ ,  $F_1$  et  $F_2$  dans le repère de la caméra.

#### Estimation de la matrice de rotation

Le vecteur  $\vec{X}$  correspond au vecteur  $(1, 0, 0)^t$  du repère global, situé sur une des droites horizontales de la mire (figure 2.2.a). Ce vecteur est donc directeur de la droite  $(O, F_1)$  et peut être exprimé dans le repère de la caméra par :

$$\vec{X}^c = \overline{OF_1}^c / \|\overline{OF_1}^c\| \tag{2.3}$$

Le sens du vecteur doit éventuellement être changé en fonction de la direction du point de fuite. De même, le vecteur  $\vec{Y}$ , situé sur une des droites verticales de la mire, peut être exprimé en fonction de la droite  $(O, F_2)$  dans le repère de la caméra. En suivant l'exemple de la figure 2.2.a, nous pouvons écrire :

$$\vec{Y}^c = -\overline{OF_2}^c / \|\overline{OF_2}^c\| \tag{2.4}$$

Le troisième vecteur de la matrice de rotation est calculé par produit vectoriel entre les deux premiers :

$$\vec{Z}^c = \vec{X}^c \wedge \vec{Y}^c \tag{2.5}$$

La matrice de rotation  $\mathbf{R}^{G \rightarrow C}$ , permettant de transformer un vecteur du repère global au repère de la caméra, s'exprime sous la forme :

$$\mathbf{R}^{G \rightarrow C} = [[\vec{X}^C] [\vec{Y}^C] [\vec{Z}^C]] \quad (2.6)$$

*Estimation du vecteur de translation*

Le repère global est défini par  $(P, \vec{X}, \vec{Y}, \vec{Z})$ . L'origine  $P$  du repère global est situé sur la mire à l'intersection de la première droite horizontale (dans le sens de  $\vec{Y}$ ) et de la première droite verticale (dans le sens de  $\vec{X}$ ) (figure 2.2.a). Le repère de la caméra (repère local) est défini par  $(O, \vec{I}, \vec{J}, \vec{K})$  avec  $\vec{I}$  un vecteur sur l'axe des abscisses de l'image (de la gauche vers la droite), et  $\vec{J}$  un vecteur sur l'axe des ordonnées négatives de l'image (du haut vers le bas).  $\vec{PQ}$  est un

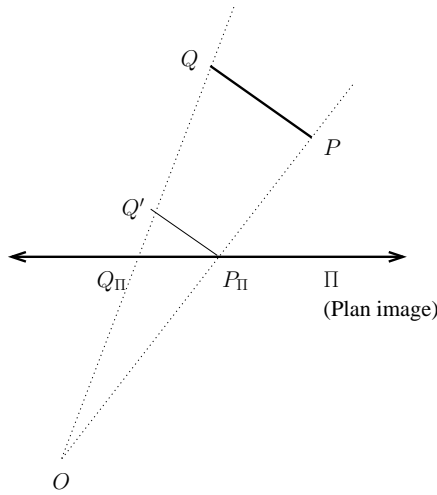


Figure 2.3 – Estimation du vecteur de translation entre les origines des deux repères ( $O$  et  $P$ ).

vecteur collinéaire à  $\vec{X}$  (dans le sens  $-\vec{X}$ ), il correspond à un segment horizontal de la mire. Ce vecteur  $\vec{PQ}$ , aux coordonnées connues (mesurées sur la mire réelle), sert à approcher la distance séparant  $O$  et  $P$  (origines des deux repères). La projection perspective  $P_{II}Q_{II}$  de ce segment sur le plan image  $\Pi$  est également connue (calculée par détection du segment sur l'image).

Le point  $Q'$ , sur la droite  $(OQ)$ , est construit de telle sorte que le vecteur  $\vec{P_{II}Q'}$  soit collinéaire à  $\vec{PQ}$  (figure 2.3). La matrice de rotation précédemment définie permet le changement de repère pour un vecteur. Elle est appliquée au vecteur  $\vec{PQ}$  pour l'exprimer dans le repère de la caméra :

$$\vec{PQ}^C = \mathbf{R}^{G \rightarrow C} \cdot \vec{PQ}^G \quad (2.7)$$

Maintenant, les coordonnées du nouveau point  $Q'$  peuvent être estimées dans le repère de la caméra :

$$Q'^C = P_{II}^C + t \cdot \vec{PQ}^C = O^C + u \cdot \vec{OQ_{II}}^C \quad (2.8)$$

Le théorème de Thalès permet une évaluation de la distance  $OP$  :

$$\Leftrightarrow \frac{\overline{OP} / \overline{OP_{II}}}{\overline{OP}} = \frac{\overline{PQ} / \overline{P_{II}Q'}}{\overline{OP_{II}} * \overline{PQ} / \overline{P_{II}Q'}} \quad (2.9)$$

Le vecteur de translation, permettant de changer l'origine du repère, est exprimé dans le repère de la caméra par :

$$\overrightarrow{OP}^C = \overline{OP} * \frac{\overrightarrow{OP_{II}^C}}{\|\overrightarrow{OP_{II}^C}\|} \quad (2.10)$$

### Analyse des résultats

Pour estimer les deux points de fuite le plus précisément possible, nous avons défini une mire couvrant la totalité de l'image. Evidemment, ceci restreint la zone de l'image consacrée à l'objet : sur l'exemple de la figure 2.1.c, la photographie est de résolution  $640 \times 480$  et l'objet représente  $150 \times 150$  pixels.

### Visualisation de lumigraphes

Pour valider visuellement la cohérence de l'estimation des paramètres de la caméra, l'ensemble des photographies est projeté sur un lumigraphe complet puis est visualisé interactivement en utilisant un outil similaire à celui développé par M. Levoy et P. Hanrahan, i.e. sans aucune information géométrique sur l'objet [LH96]. Voici, sur la figure 2.4, une série de vues sur le plan  $uv$  d'un slab donné ( $u$  varie de 0 à 7, de la gauche vers la droite). Les images apparaissent déformées pour certains points de vue car toutes les images sont projetées sur un même plan  $uv$  (généralement non parallèle au plan image des photographies originales). Sur la première série



Figure 2.4 – Quelques points de vue (l'indice  $u$  varie de 0 à 7, de gauche à droite) pour un slab du lumigraphe. La construction du lumigraphe utilise, pour générer chaque échantillon du plan  $st$ , a. au maximum une photographie ; b. au maximum quatre photographies ; c. au maximum quatre photographies dont la focale a été estimée entre 90mm et 120mm.

de vues, le plan  $st$  de chaque vue est généré par projection de la photographie la plus proche. Le test de recherche de la photographie la plus proche est effectué pour chaque maille du plan  $st$ . Par conséquent, une image du plan  $uv$  peut être construite à partir de plusieurs photographies originales. Lorsque la photographie la plus proche est trop éloignée du point de vue à reconstruire, elle n'est pas utilisée. Ainsi des trous peuvent apparaître dans le lumigraphe final, correspondant à des prises de vues manquantes. Sur la seconde série de vues, les 4 photographies les plus proches ont été utilisées pour générer les nouvelles vues (un angle de déviation maximale est

toujours utilisé pour ne pas prendre en compte les prises de vue trop éloignées). La dernière série de vue est générée comme la seconde à l'exception de certaines photographies non utilisées pour construire le lumigraphe : les images dont la focale n'est pas estimée précisément.

### *Densité de l'échantillonnage*

L'outil de visualisation permet facilement de vérifier la densité de l'échantillonnage et de décider si l'acquisition de points de vue supplémentaires est nécessaire pour combler certains slabs. Pour obtenir un échantillonnage complet de  $8 \times 8$  vues sur le plan  $uv$ , nous avons utilisé 300 photos valides sur la figure 2.4.c, soit environ quatre photographies par nouvelle vue reconstruite.

### *Estimation des paramètres*

L'outil de visualisation permet également de vérifier si certaines photographies ne sont pas projetées au mauvais endroit : c'est le cas de la vue 0.4.3 sur la figure 2.4.a où l'image n'est pas en concordance avec les vues voisines. La vue 0.4.3 est nettement mieux reconstruite après suppression des photographies dont la focale n'est pas estimée correctement (figure 2.4.c). En effet, lorsque la focale n'est pas correctement estimée, les paramètres extrinsèques de la caméra sont également estimés avec une forte imprécision. Cette imprécision découle de l'estimation de points de fuite proches de l'infini (vues 0.4.3 et 0.5.3).

Pour une focale théorique de  $105\text{mm}$ , environ 15% des photographies sont éliminées car leur focale est estimée hors de l'intervalle  $90\text{mm} - 120\text{mm}$ . Ce seuil est fixé de manière empirique, après comparaison visuelle de plusieurs lumigraphes créés en utilisant des seuils différents. Cette suppression de photographies entraîne une réduction forte de la densité d'échantillonnage dans certaines régions. Après suppression de ces photographies, la reconstruction du slab en  $8 \times 8$  vues sur le plan  $uv$  laisse apparaître 3 points de vue sans photographie à projeter mais l'ensemble des vues restantes gardent une cohérence spatiale (figure 2.4.c). L'acquisition de nouvelles photographies corrige difficilement le problème car au moins un point de fuite apparaît à l'infini sur ces photographies. Pour résoudre le problème, il faut déplacer la mire (sans bouger la position et l'orientation de l'objet par rapport aux sources lumineuses) et réaliser une nouvelle acquisition. Le déplacement de la mire entraîne la définition d'un nouveau repère local (lié à la nouvelle position de la mire) et une matrice de changement de repère pour pouvoir projeter toutes les images sur le lumigraphe (repère global). Chaque imprécision sur le calcul du nouveau repère local entraîne une erreur lors de la reprojection des images sur le lumigraphe.

### **Conclusion**

Lors de la construction d'un lumigraphe, chaque vue est généralement produite à partir de plusieurs photographies (même si une seule image est utilisée pour générer chaque échantillon  $st$ ) et des cisaillements apparaissent fréquemment dans la nouvelle vue générée. Ces artefacts sont particulièrement visibles sur la vue 0.0.3 de la figure 2.4.a et sont dus à une légère erreur dans l'estimation des paramètres extrinsèques de la caméra. Ils pourraient également apparaître sur des problèmes de changement de profondeur, mais ce n'est pas le cas sur l'objet de la figure 2.4. Pour pallier ce problème, nous proposons de générer chaque échantillon  $st$  en utilisant plusieurs images originales (comme dans les secondes et troisièmes lignes de la figure) mais cela introduit du flou dans les nouvelles vues générées. En effet, toutes les images de la figure 2.4.c apparaissent moins nettes malgré l'utilisation de photographies étalonnées. La réduction de

l'intervalle d'acceptation des focales augmente de manière considérable le nombre d'images nécessaires sans toutefois éviter l'utilisation de photographies pour lesquelles la distance focale est bien estimée mais les paramètres extrinsèques sont erronés. Le flou apparaissant est évalué à un décalage pouvant atteindre 10 pixels entre les différentes photographies utilisées pour une même vue. Ce décalage peut être réduit en utilisant des méthodes de mise en correspondance de pixels. Cela permet de rectifier l'orientation et la direction de visée de la caméra. Malheureusement la position de la caméra peut être erronée et il est difficile de modifier ce paramètre.

L'accumulation des problèmes de calibrage de la caméra montre que l'utilisation d'une mire 2D n'est pas simple et demande des traitements supplémentaires pour étalonner correctement les différentes images. Des travaux récents de G. Coombe et al. permettent de construire interactivement un champ de lumière paramétré sur la surface [CHLG05]. Deux ordinateurs sont utilisés, l'un pour piloter la caméra effectuant une acquisition à 30 images par seconde, l'autre pour afficher le rendu du champ de lumière en cours de construction. Cette technique permet d'avoir un retour immédiat sur les images acquises et de vérifier qu'aucun point de vue du champ de lumière n'est en sous-échantillonnage. Pour effectuer la construction du champ de lumière paramétré sur la surface, un algorithme incrémental d'analyse en composantes principales est utilisé.

Comme le système utilisant une mire 2D ne nous permet pas d'acquérir des lumigraphes complets, nous choisissons d'utiliser un système d'acquisition sans mire, plus restrictif mais facile à mettre en œuvre et permettant un étalonnage plus précis de la caméra.

## A.2.2 Utilisation d'un plateau tournant

Ce système laisse moins de liberté lors de l'acquisition des images car la position de l'appareil photographique est définie manuellement ; il est également fixé sur un trépied assurant ainsi une meilleure qualité des prises de vue. Nous pouvons opter pour un système d'acquisition similaire à celui des *light fields* [LH96] où l'appareil photographique est déplacé le long d'un plan pour acquérir chaque slab, mais nous préférons faire tourner l'objet devant la caméra en le fixant sur un plateau tournant. Ce choix paraît plus judicieux car il est plus facile de faire pivoter un objet fixé sur un plateau tournant que de déplacer manuellement une caméra le long d'un plan. Le système d'acquisition décrit par un plateau tournant est simple à mettre en place et robuste car il permet d'étalonner la caméra avec fiabilité.

### Principe

Pour réaliser l'acquisition d'un objet avec des conditions d'éclairage constantes, les sources lumineuses sont fixées sur le plateau tournant. De plus, pour espérer obtenir un échantillonnage dense de la fonction plénoptique autour de l'objet à acquérir, nous avons placé les lampes au dessus de l'objet et augmenté au maximum l'écartement des bras du système de fixation (par rapport à l'axe de rotation du plateau). Ceci permet de réduire la partie occultée de l'objet lorsque ces bras sont proches du champ angulaire de l'appareil photographique (la figure 2.5.a représente schématiquement le système d'acquisition et la figure 2.8.a montre le système réel).

L'utilisation d'un trépied permet de régler la vitesse d'obturation sur des valeurs plus lentes et ainsi, de réduire l'éclairement sur l'objet. Cela permet également de réduire l'ouverture de l'obturateur (diaphragme) pour augmenter la profondeur de champ. Tous les paramètres intrinsèques de la caméra sont fixés pour obtenir une série d'images directement exploitables (vitesse d'obturation, ouverture de l'obturateur, focale). Entre chaque prise de vue, le plateau est tourné d'un angle de  $5^\circ$  jusqu'à ce qu'un tour complet soit effectué. Comme la rotation du plateau est



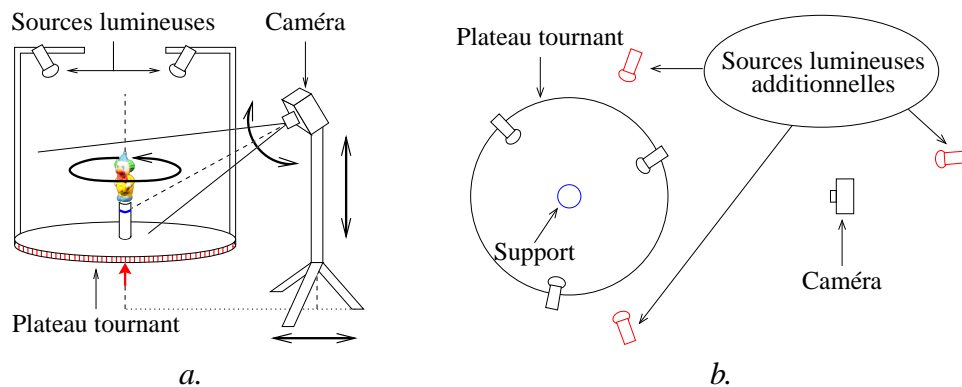


Figure 2.5 – Système d'acquisition utilisé pour photographier des objets réels ; a. acquisition de la luminance émise par la surface de l'objet avec des conditions d'éclairage constantes (vue de profil) ; b. extraction de l'objet avec l'arrière-plan en utilisant des sources lumineuses supplémentaires (vue du dessus).

manuelle, nous avons réduit l'imprécision sur l'angle de rotation en utilisant un plateau de grande taille (60cm de diamètre) gradué sur le bord (tous les 5°). En positionnant l'objet sur le centre de rotation du plateau, la distance séparant la caméra de l'objet reste constante pendant l'acquisition d'une série de photographies. Lors de l'acquisition d'une nouvelle série d'images avec une hauteur de caméra différente, le trépied est déplacé afin que la distance reste constante entre l'objet et la caméra (à quelques centimètres près).

### Mise en œuvre

#### Segmentation du contour de l'objet

Pour faciliter l'extraction de l'objet avec l'arrière-plan, un tissu noir mat est installé sur le système d'acquisition et sur le mur face à la caméra. De plus, des sources lumineuses puissantes fixes par rapport à la caméra sont ajoutées pour éclairer toutes les parties de l'objet visibles depuis la caméra (voir figure 2.5.b). Sans ces sources lumineuses supplémentaires, certaines parties de l'objet sont très sombres et difficilement discernables de l'arrière-plan. Evidemment, les sources lumineuses additionnelles ne doivent pas intervenir dans l'acquisition des propriétés de réflectance de l'objet. Par conséquent, pour chaque position du plateau tournant, deux images de l'objet sont acquises avec des conditions d'éclairage différentes. La première photographie représente l'objet réel, éclairé par un ensemble de sources fixes par rapport à l'objet comme décrit précédemment. La seconde photographie est acquise en présence des sources lumineuses supplémentaires (figures 2.6.a et 2.6.b).

Après avoir détourné l'objet sur la photographie sur-éclairée, l'information de contour de l'objet est enregistrée sur la photographie capturant la réflectance du matériau en encodant les pixels représentant l'arrière-plan comme des pixels parfaitement noirs (leur valeur est mise à (0, 0, 0) dans l'espace de couleur *RGB*). L'utilisation de sources lumineuses additionnelles placées sur les côtés de la caméra permettent de sur-éclairer l'objet sans sur-éclairer l'arrière-plan (tissu mural noir), et ainsi facilitent l'extraction de la silhouette de l'objet. De plus, l'utilisation d'une source lumineuse additionnelle placée derrière la caméra permet la détection d'éventuels trous dans l'objet et la détection de couleurs sombres, proches du noir sur la surface de l'objet.



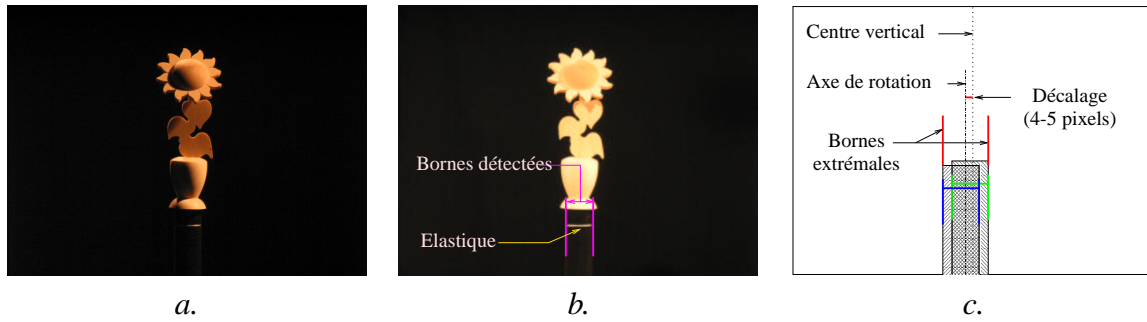


Figure 2.6 – a. Acquisition des propriétés de réflectance de l’objet ; b. photographie surexposée pour séparer l’objet de l’arrière-plan ; l’élastique placé sur le support sert à estimer l’orientation de la caméra ; c. recentrage d’une série de photographies après estimation de l’axe de rotation du plateau.

### Recentrage des images

Pour une série de photographies acquises sans déplacement du trépied (uniquement rotation du plateau tournant), l’axe de rotation du plateau est fixe dans toutes les images acquises. Pour simplifier l’estimation des paramètres extrinsèques de la caméra, l’axe de rotation est détecté à partir d’un élastique placé sur le support de l’objet : toute la série d’images subit un décalage horizontal de quelques pixels (4 ou 5 au maximum) afin de faire coïncider l’axe de rotation du plateau avec le centre des images (voir figure 2.6.c pour des détails sur l’évaluation du décalage à partir de deux images). L’élastique est détecté à une position fixe sur une série de photographies (à 1 ou 2 pixels près) car le support est placé sur l’axe de rotation du plateau. La détection de cet élastique sert également à estimer l’angle d’élévation de la caméra après évaluation de sa position (section suivante).

### Estimation du vecteur de translation

Le repère global est défini par  $(P, \vec{X}, \vec{Y}, \vec{Z})$  avec  $P$  le centre du plateau tournant et le repère local est centré sur la caméra, comme précédemment  $(O, \vec{I}, \vec{J}, \vec{K})$  (figure 2.7.a). La première

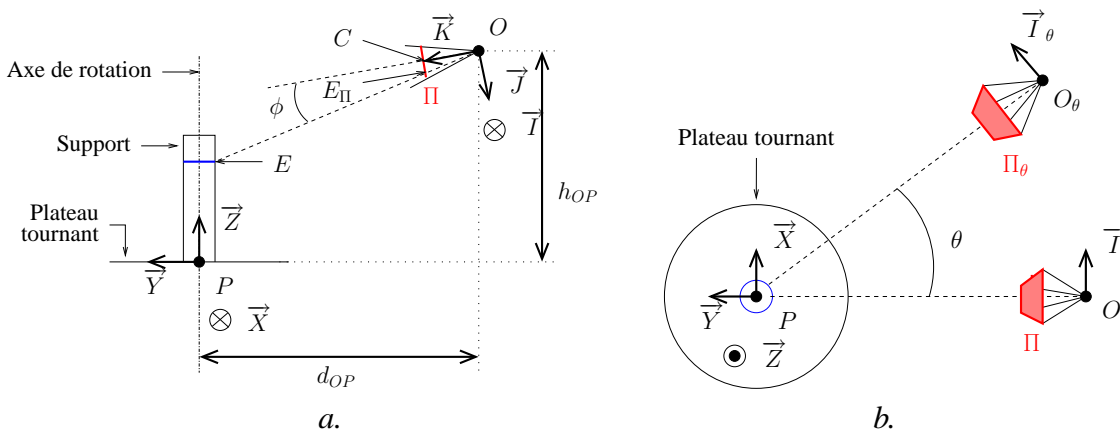


Figure 2.7 – a. Coupe du système d’acquisition selon le plan  $(P, \vec{Y}, \vec{Z})$  ; b. projection orthogonale du système d’acquisition selon l’axe  $\vec{Z}$  ;  $P$ , le centre du plateau, est l’origine du repère global  $(P, \vec{X}, \vec{Y}, \vec{Z})$  ;  $O$ , le centre optique de la caméra, est l’origine du repère local  $(O, \vec{I}, \vec{J}, \vec{K})$ .

image d'une série de photographies est prise par une caméra  $O$  positionnée sur le plan  $(P, \vec{Y}, \vec{Z})$  avec une coordonnée  $y$  négative (voir figure 2.7.b). Comme l'axe de rotation du plateau passe par le centre de l'image, les vecteurs  $\vec{X}$  et  $\vec{I}$  sont égaux. Pour aligner précisément la caméra avec le plan  $(P, \vec{Y}, \vec{Z})$ , la droite d'équation  $(x = 0)$  est tracée sur le sol et un fil à plomb fixé sur le trépied permet de contrôler la direction de son déplacement lors de l'acquisition de nouvelles séries d'images en changeant la hauteur de la caméra.

La position de la caméra est évaluée manuellement lors de la première acquisition en mesurant la distance au sol  $d_{OP}$  séparant la caméra et le centre du plateau, et la hauteur  $h_{OP}$  entre la caméra et le plateau, au centimètre près (voir la figure 2.7.a pour les notations). Ainsi une mesure des coordonnées de la caméra est obtenue dans le repère global, et par conséquent, le vecteur de translation entre les origines des deux repères devient :

$$\vec{OP}^G = (0, d_{OP}, -h_{OP}) \quad (2.11)$$

La rotation physique du plateau tournant d'un angle  $\theta$  est traduite par la rotation virtuelle de la caméra d'un angle  $-\theta$ . La nouvelle position de la caméra  $O_\theta$  est donc déduite de la position initiale  $O$  en appliquant une rotation selon l'axe  $\vec{Z}$  d'un angle  $-\theta$  (voir figure 2.7.b) :

$$\vec{O_\theta P}^G = \mathbf{Rot}_{\vec{Z}}(-\theta) \cdot \vec{OP}^G \quad (2.12)$$

#### Estimation de la matrice de rotation

Pour estimer la matrice de rotation servant à changer de repère, la détection de l'élastique sur la première image de la série de photographies est utilisée pour estimer l'angle d'élévation de la caméra puis les vecteurs de la base du repère de la caméra sont estimés dans le système de coordonnées du repère global. La position de l'élastique fixé sur le support est mesurée manuellement (comme pour le centre optique  $O$  de la caméra) et le vecteur  $\vec{OE}$  est estimé dans le repère global. Ensuite, à partir des coordonnées du centre de l'élastique détecté sur l'image (point  $E_\Pi$ ), des coordonnées du centre de l'image  $C$  et de la focale connue de l'appareil photo  $\vec{OC}$ , l'angle  $\phi$  formé entre la direction de l'élastique (vecteur  $\vec{OE}$ ) et la direction du centre de l'image (vecteur  $\vec{K}$ ) depuis le centre optique est estimé (voir figure 2.7.a pour les notations) :

$$\phi = \arctan(\|\vec{E_\Pi C}\| / \|\vec{OC}\|) \quad (2.13)$$

Le vecteur  $\vec{K}$  peut être estimé dans le repère global par une rotation du vecteur  $\vec{OE}$  d'un angle  $\phi$  autour de l'axe  $\vec{X}$ , puis par une normalisation du vecteur obtenu :

$$\vec{K}^G = \mathbf{Rot}_{\vec{X}}(\phi) \cdot \vec{OE}^G / \|\vec{OE}^G\| \quad (2.14)$$

Le vecteur  $\vec{I}$  peut être exprimé puisqu'il est confondu avec le vecteur  $\vec{X}$  :

$$\vec{I}^G = \vec{X}^G \quad (2.15)$$

Et le vecteur  $\vec{J}$  peut être exprimé par produit vectoriel des deux précédents :

$$\vec{J}^G = \vec{K}^G \wedge \vec{I}^G \quad (2.16)$$

La matrice de rotation  $\mathbf{R}^{C \rightarrow G}$ , permettant de transformer un vecteur du repère de la caméra au repère global, est exprimée sous la forme :

$$\mathbf{R}^{C \rightarrow G} = [[\vec{I}^G] [\vec{J}^G] [\vec{K}^G]] \quad (2.17)$$

Pour les autres images acquises après rotation du plateau tournant d'un angle  $\theta$ , il suffit d'effectuer le produit matriciel entre la matrice de rotation d'angle  $-\theta$  autour de l'axe  $\vec{Z}$  et la matrice (2.17) définie précédemment :

$$\mathbf{R}^{C \rightarrow G} = \mathbf{Rot}_{\vec{Z}}(-\theta) \cdot [[\vec{I}^G] [\vec{J}^G] [\vec{K}^G]] \quad (2.18)$$

### Analyse des résultats

La suppression de la mire permet de réserver une plus grande proportion de l'image à la représentation de l'objet. Seul un élastique est utilisé pour étalonner les caméras mais il est placé sur le support de l'objet, proche de celui-ci mais jamais confondu avec l'objet lors de l'acquisition de multiples points de vue. Ainsi, avec des photographies de résolution  $640 \times 480$ , l'objet est représenté par 250 à 300 pixels de hauteur, soit des acquisitions deux fois plus détaillées qu'avec le système d'acquisition précédemment présenté.

L'estimation précise de la position de la caméra n'est pas très importante car toutes les coordonnées sont calculées relativement à la position estimée et la caméra est fixe pendant l'acquisition d'une série d'images. En revanche entre plusieurs séries d'images la caméra doit être déplacée minutieusement pour assurer un calibrage correct entre les séries d'images : le fil à plomb permet de guider ce déplacement. Par conséquent la robustesse du système d'acquisition est définie par une estimation précise de la direction de visée de la caméra sur chaque point de vue.

### Visualisation de lumigraphes

Lorsque l'acquisition est effectuée sur  $360^\circ$  (un tour complet de plateau tournant), la caméra est déplacée pour éventuellement acquérir de nouvelles images avec le même protocole d'acquisition. Lors de l'élévation de la caméra, le trépied est déplacé afin de garder une distance objet-caméra constante et par conséquent une mise au point fixe de la caméra. Ainsi ce système permet l'acquisition de photographies tout autour de l'objet en faisant varier virtuellement la position de la caméra sur une sphère englobant l'objet : l'angle d'élévation de la caméra varie sur plus de  $60^\circ$ , comme illustré sur la figure 2.8.

La rotation d'angle constant entre les prises de vue successives assure une certaine régularité sur l'ensemble des points de vue acquis et permet la reconstruction des quatre slabs autour de l'objet presque uniquement par projection d'une seule photographie sur chaque point de vue du plan  $uv$  (la figure 2.9 affiche une série de vues sur le plan  $uv$  d'un slab pour divers objets,  $u$  variant de 0 à 7, de la gauche vers la droite). Comme la caméra est déplacée sur une *sphère* englobant l'objet et que la construction du lumigraphe nécessite des caméras positionnées sur un *cube* englobant l'objet, les points de vue du lumigraphe ne peuvent coïncider parfaitement avec les points de vue acquis. Par conséquent chaque vue du lumigraphe est rarement définie par la projection d'une seule photographie : plusieurs images sont généralement nécessaires.

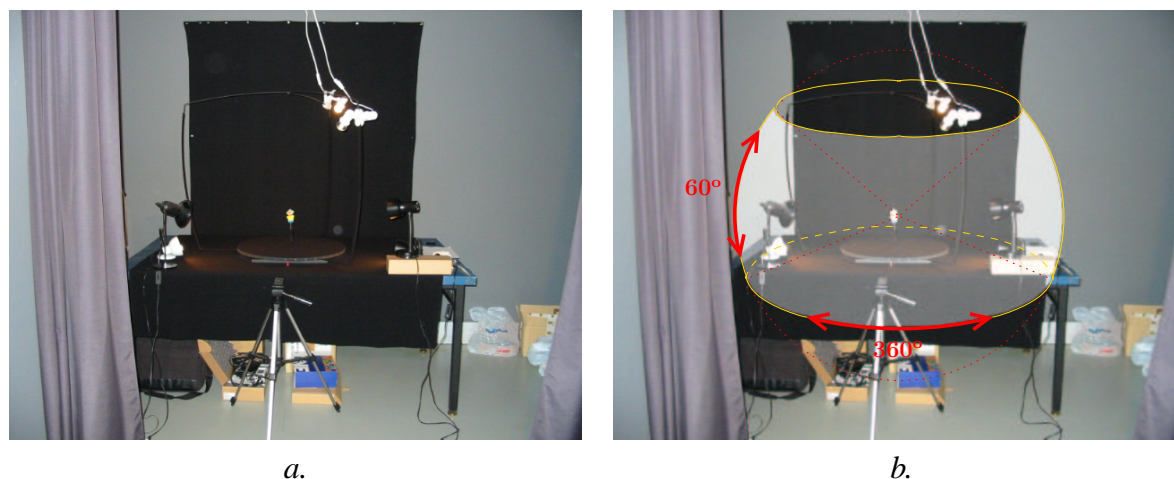


Figure 2.8 – a. Photographie du système d'acquisition ; b. acquisition d'un objet sur 360°.

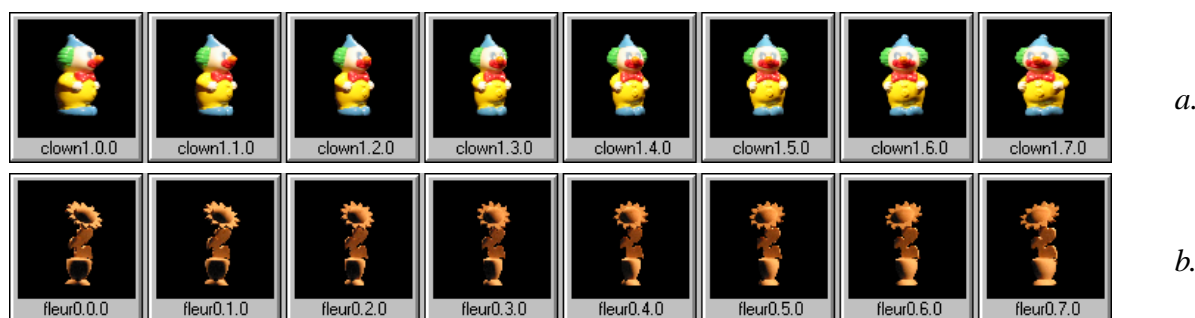


Figure 2.9 – Quelques points de vue (variation de l'indice  $u$ ) pour un slab du lumigraphe ; a. clown, objet spéculaire en plastique ; b. fleur, objet diffus en bois.

### Densité de l'échantillonnage

Pour obtenir un échantillonnage complet de  $8 \times 8$  vues sur les 4 plans  $uv$  autour de l'objet, 15 séries de 72 photographies sont acquises soit un peu plus de 1000 images. Chaque série est acquise après avoir élevé de  $4\text{cm}$  la caméra placée à  $70\text{cm}$  de l'objet. Ceci représente environ quatre photographies disponibles pour générer chaque point de vue du lumigraphe, avec une rotation de la caméra de  $5^\circ$  selon l'axe horizontal et de  $3 - 4^\circ$  selon l'axe vertical.

Cet échantillonnage est donc suffisant mais il est également nécessaire car l'interpolation entre plusieurs photographies pour générer chaque échantillon  $st$  des images du lumigraphe (par exemple, interpolation de valeurs récupérées dans les deux photographies les plus proches du point de vue  $uv$  à générer) conduit à des décalages de plusieurs pixels pour les surfaces de l'objet situées au premier plan ou en arrière plan par rapport au plan  $st$ . Ceci provient de l'algorithme de construction du lumigraphe qui n'utilise aucune information géométrique sur la forme de l'objet (la figure 1.5.b du chapitre I montre le décalage pour une surface située en avant plan : attention, les notations des plans  $uv$  et  $st$  sont inversées). Dans l'exemple du clown, mesurant environ  $8\text{cm}$  de haut, l'interpolation entre deux photographies distantes d'un angle horizontal de  $5^\circ$  par rapport à l'objet entraîne un décalage théorique de  $5 - 6$  pixels sur les parties de l'objet situées au premier plan (à  $2\text{cm}$  du plan  $st$ ). Sur la figure 2.10.a, ce décalage peut effectivement être constaté sur les

zones centrales de l'objet (yeux, bouche, noeud papillon par exemple). Après mesure, le décalage s'avère conforme à l'estimation effectuée, il ne peut pas être réduit sans effectuer l'acquisition de points de vue intermédiaires. Par conséquent, nous n'interpolons pas entre plusieurs images pour générer les échantillons  $st$  du lumigraphe afin de représenter un lumigraphe par une collection d'images non floues.



Figure 2.10 – Images de résolution  $256 \times 256$  extraites du lumigraphe du clown ; a. la vue de gauche est générée par projection d'une seule photographie ; les échantillons  $st$  de la vue de droite (identique à la vue précédente) sont générés par interpolation entre deux photographies ; b. la vue de gauche, correspondant au slab  $v = 2$ , est générée par projection d'une seule photographie ; la vue de droite, correspondant au slab  $v = 3$ , est générée à partir d'une seule photographie par échantillon  $st$  mais deux photographies différentes sont utilisées pour générer l'image complète : une séparation entre les deux photographies projetées apparaît horizontalement au centre de l'image, cisailant la main du clown.

De même, lorsqu'une image  $st$  du lumigraphe est générée par projection de régions provenant de plusieurs photographies, des décalages peuvent survenir sur la jonction entre les différentes photographies projetées. Dans l'exemple du clown, l'utilisation de deux photographies distantes d'un angle vertical de  $3 - 4^\circ$  par rapport à l'objet entraîne un décalage théorique de 4 pixels sur les parties de l'objet situées au premier plan (toujours estimées à  $2\text{cm}$  du plan  $st$ ). Sur la figure 2.10.b, une zone de jonction apparaît sur la main et le bras du clown. Un décalage vertical de 5 pixels peut effectivement être mesuré sur la main du clown (partie de l'objet la plus au premier plan). Néanmoins, le décalage vertical entre les deux photographies projetées tend à être réduit lorsque la surface de l'objet se rapproche du plan  $st$  : aucune zone de jonction sur la région située la plus à droite de l'image (le dos du clown est situé sur le plan  $st$ , passant généralement par le centre de l'objet). Ceci conforte donc la précision de la projection des photographies.

Par conséquent, le système d'acquisition ainsi décrit permet d'acquérir des objets pour les visualiser en tant que lumigraphe, sans même passer par une étape de reconstruction géométrique de l'objet. Avec une acquisition de photographies en moyenne tous les  $4^\circ$ , la déformation atteint tout au plus  $2^\circ$  en choisissant toujours la photographie la plus proche des nouveaux points de vue à générer. Ceci représente un décalage réel de 2 pixels maximum sur des images de lumigraphe de résolution  $256 \times 256$ . C'est une déformation acceptable vu la simplicité de notre système d'acquisition.

### Estimation des paramètres

Les mesures de la figure 2.10 montrent la précision de l'estimation des paramètres de la caméra. En effet, sur les deux images de la figure 2.10.a, l'absence de décalage sur la silhouette de l'objet permet de valider la précision de l'estimation de la direction de visée de la caméra : la



silhouette est utilisée car l'estimation de sa profondeur est proche du plan  $st$  (plan de projection de la photographie). Sur les deux images de la figure 2.10.b, l'absence de décalage horizontal sur l'ombre proche du dos du clown permet de valider la précision de la position des deux caméras utilisées : elles sont bien placées sur un même plan vertical passant par le centre de l'objet.

Quelques imprécisions peuvent subsister, notamment sur la position réelle de la caméra. Le centre de projection n'est peut-être pas évalué au bon endroit, mais l'erreur sur cette position est la même pour toutes les acquisitions et par conséquent, ne nuit pas au fonctionnement global du système. De plus, l'orientation de la caméra entre chaque série de photographies est vérifiée minutieusement : le vecteur  $\vec{I}$  de la caméra doit toujours rester dans le plan horizontal lors d'un changement de la direction de visée. Ceci permet de définir la robustesse de notre système d'acquisition.

### A.2.3 Conclusion

Ce système d'acquisition permet l'acquisition de quatre slabs d'un lumigraphe d'objet réel. L'occultation d'une partie de l'objet par les sources lumineuses empêche la caméra d'être placée au dessus de l'objet. La contre-plongée de la caméra est limitée par le plateau tournant de grande taille, occultant également une partie de l'objet. Toutefois, cet inconvénient est réduit en sur-élevant l'objet avec un support de faible dimension (1cm de diamètre) et l'acquisition de quatre slabs sur six est un bon compromis permettant de visualiser l'objet sur  $360^\circ$ . En effet, il est extrêmement difficile de faire une acquisition complète de l'objet en respectant des conditions d'éclairage fixes.

Les quelques imprécisions sur la génération de lumigraphes d'objets réels (cisaillement de l'objet sur 1 à 2 pixels, voir l'image de droite de la figure 2.10.b) ne dépendent pas de l'acquisition mais de la non prise en compte de la géométrie de l'objet lors de la reprojection des photographies sur les plans  $st$  du lumigraphe. Par conséquent le système d'acquisition est validé tel qu'il est présenté, les séries de photographies acquises sont directement utilisées pour reconstruire la géométrie des objets réels sans passer par la structure de lumigraphe.

## A.3 Construction de lumigraphes synthétiques

La structure de lumigraphe est utilisée pour représenter des objets virtuels : les images sont obtenues par un algorithme de lancer de rayons permettant de contrôler précisément les calculs d'illumination. Le modèle paramétrique de Phong modifié est utilisé pour calculer la réflectance de la surface des objets. Disposant de la géométrie exacte, la structure de lumigraphe est adaptée à la représentation d'objets synthétiques : il n'y a pas de déformation similaire aux lumigraphes d'objets réels (cisaillement montré sur l'image de droite de la figure 2.10.b).

Des lumigraphes synthétiques sont générés avec le logiciel *PovRay* car il permet de générer des objets :

- avec une géométrie par algèbre de volumes (CSG) ;
- composés de textures procédurales (section A) ;
- synthétisés avec des calculs d'illumination plus complexes tels que la perturbation de la normale sur la surface (section A) ou l'utilisation de modèles d'illumination différents de celui de Phong modifié.

Le code source du logiciel a été modifié afin de générer de manière automatique des lumigraphes complets (6 slabs) d'objets virtuels.

L'espace disque nécessaire au stockage des objets basés-image est en général problématique. Nous utilisons des lumigraphes de faible résolution afin de limiter la taille des données. Les lumigraphes synthétiques utilisés sont représentés par des plans  $uv$  de faible résolution ( $8 \times 8$  vues) et des plans images  $st$  de résolution moyenne ( $256 \times 256$  pixels, soit 150 à 200 pixels pour représenter réellement l'objet). Ainsi, la taille des données à charger et à manipuler est restreinte à  $72Mo$  par lumigraphe complet.

L'application de la librairie de compression sans perte *zlib* réduit généralement la taille des lumigraphes synthétiques par un facteur de 15. Ce fort taux de compression est dû à la redondance de l'information stockée dans les images synthétiques : l'algorithme de lancer de rayon est non seulement déterministe, mais il est utilisé pour générer des séries de points de vue échantillonnés sur une grille régulière (le plan  $uv$ ). Les lumigraphes synthétiques sont utilisés sous cette forme compressée lors des étapes d'analyses (reconstruction géométrique, détection de sources lumineuses et estimation de BRDFs).

Ce format de compression ne permet pas toujours d'obtenir des taux de compression très élevés malgré la forte redondance entre les images. Par exemple, pour des lumigraphes d'objets réels, la librairie *zlib* donne un facteur de compression autour de 6.

D'autres méthodes de compression permettent de réduire de façon drastique la taille des lumigraphes. Par exemple, l'utilisation de la quantification vectorielle (méthode de compression avec perte utilisée dans [LH96]) est un exemple classique permettant d'obtenir un facteur de compression autour de 15 sans dégradation visuelle significative. Néanmoins, ces méthodes de compression ne peuvent pas être utilisés dans le cadre de notre travail : les dégradations dans les images (même non visibles) entraînent des pertes d'information et des erreurs dans les algorithmes d'estimation des sources lumineuses et des BRDFs de la surface.

Nous utilisons uniquement ce type de compression lors de la visualisation de lumigraphes en utilisant un modèle de compression basé sur la qualité visuelle des images compressées et non, comme les méthodes traditionnelles, sur le taux de compression. La méthode de compression proposée est présentée dans la section **B** du chapitre IV.

## **B Reconstruction de la forme**

La reconstruction de la forme à partir des images est une première étape indispensable car couplée à l'estimation de la normale de la surface, elles vont contribuer à la détection des sources lumineuses et à l'estimation de la réflectance de la surface. De plus, l'état de l'art sur le rééclairage (section **E** du chapitre I) montre que le modèle géométrique est indispensable pour une insertion réaliste d'un objet réel dans un environnement avec rééclairage (normale indispensable) et prise en compte des ombres projetées (forme nécessaire).

Notre méthode de reconstruction de la géométrie d'un objet basé-image s'appuie uniquement sur une série d'images calibrées représentée :

- soit par un ensemble de photographies avec une matrice de changement de repère pour chacune d'elles (acquisition d'un objet réel) ;
- soit par un lumigraphe complet (acquisition d'un objet virtuel).

Aucune information géométrique n'est utilisée, les coordonnées des différentes prises de vue sont les seules informations supplémentaires utilisées.



### B.1 Choix de la méthode

L'état de l'art sur la reconstruction géométrique à partir d'une série d'images (section B.2 du chapitre I) montre que les méthodes posent de fortes hypothèses :

- soit sur les conditions d'illumination utilisées lors de l'acquisition ;
- soit sur les propriétés de réflectance des matériaux composant l'objet.

Dans le cadre de ce travail, nous souhaitons pouvoir reconstruire et ré-éclairer un large panel d'objets basés-image sans se limiter à des réflectances lambertiennes ou des surfaces texturées. Par exemple, devoir considérer tous les objets réels comme des surfaces lambertiennes est très peu réaliste et nuit à l'étape finale de rééclairage réaliste des objets. De plus, aucune hypothèse n'est posée sur les conditions d'éclairage utilisées lors de l'acquisition des objets, restreignant à nouveau l'éventail des méthodes mises à disposition.

Les surfaces fortement texturées, nécessaires aux méthodes de stéréovision, peuvent être obtenues par projection d'un motif sur l'objet lors de l'acquisition : cette méthode est utilisée par C. Marchessoux lors de l'acquisition 3D de pieds humains [Mar03] mais elle ne permet pas d'acquérir les propriétés de réflectance des matériaux.

Notre choix s'est ainsi automatiquement porté sur l'utilisation d'une méthode de reconstruction de la forme à partir de l'enveloppe visuelle de l'objet. Cette approche, plus connue sous le nom de *shape from silhouette* (section B.2.1 du chapitre I), utilise une série de vues bien calibrées pour représenter la surface d'un objet par des voxels. Aucune contrainte supplémentaire n'est imposée sur les images utilisées, ainsi un ensemble dense de vues disponibles tout autour de l'objet est un atout majeur pour reconstruire précisément la surface.

Néanmoins, cette méthode possède un inconvénient fort : les cavités des objets ne peuvent pas être reconstruites par la seule utilisation des silhouettes vues sur les images. Ce type d'approche est souvent combiné avec des méthodes de mise en correspondance d'images, permettant de creuser les cavités en supprimant les voxels n'appartenant pas à la surface, comme celles proposées à la section B.2.2 de l'étude bibliographique. Malheureusement, ces algorithmes s'appuient sur l'évaluation de la différence d'illumination d'un voxel vu sur deux points de vue proches et nécessitent à nouveau de travailler avec des surfaces purement lambertiennes. Par conséquent, nous faisons l'hypothèse réductrice que les objets utilisés ne possèdent pas de cavité afin de permettre l'utilisation de surfaces spéculaires. Si certains objets sont composés de petites cavités, celles-ci sont ignorées lors de la reconstruction de la forme. Les objets composés de surfaces trop complexes comme les vêtements, les fourrures sont également évités.

Nous construisons une surface discrète 6-connexe afin de pouvoir utiliser l'algorithme des *marching cubes* [LC87] et ainsi produire un maillage polygonal de la surface de l'objet. Cet algorithme est généralement utilisé pour construire un maillage à partir d'un ensemble de voxels [GGSC96, HLGB03]. Nous appliquons quelques modifications à l'algorithme original afin de produire sans ambiguïté possible un maillage fermé représentant la surface de l'objet (section B.5). Une structure d'arbre octal est utilisée afin de pouvoir passer d'une représentation discrète à une représentation par polygones de la géométrie reconstruite. Nous proposons également d'utiliser des informations complémentaires fournies par les pixels des images afin de produire un maillage plus proche de la silhouette de l'objet vue sur chaque image (section B.6).

À présent, nous proposons les détails de notre structure de représentation et la construction de la surface discrète à partir des images.

## B.2 Reconstruction basée-voxel

Une première approche de la forme de l'objet est obtenue par application d'un algorithme de sculpture représentant la surface par un ensemble de voxels. Nous avons développé une approche similaire à celle de R. Szeliski [Sze93] représentant l'objet par un arbre octal<sup>1</sup> sculpté de manière hiérarchique par projections successives sur toutes les vues de l'objet.

### B.2.1 Principe

La forme initiale de l'objet correspond à un voxel stocké sous une représentation en octree et cette structure est raffinée itérativement pendant le processus de reconstruction. Pour chaque vue de l'objet, l'ensemble des voxels obtenus par sculpture (au départ, un seul voxel) sont projetés sur le plan image et comparés avec la silhouette de l'objet (pixels non noirs de l'image). Trois situations différentes peuvent exister pour chaque voxel projeté :

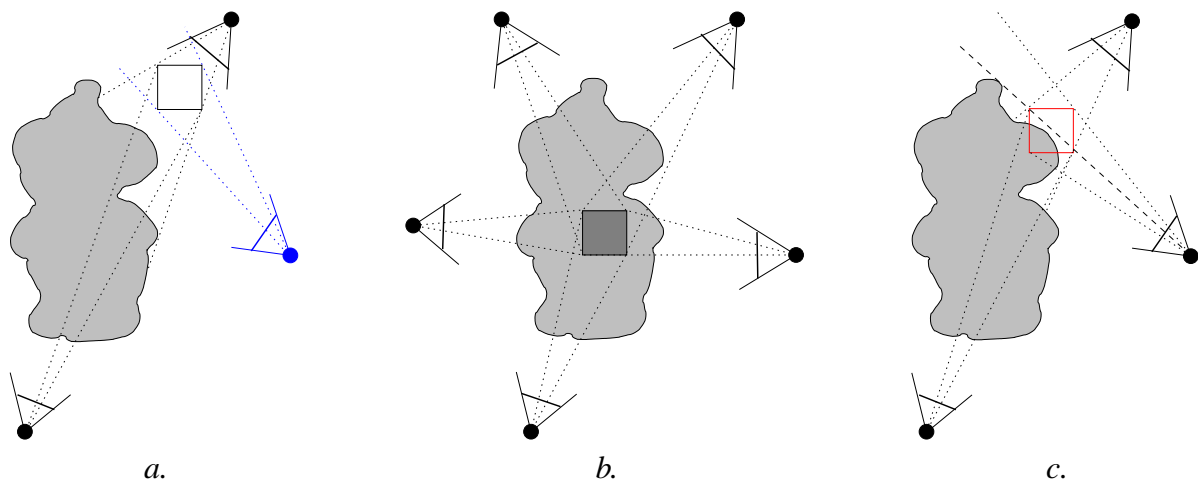


Figure 2.11 – Classification d'un voxel après sa projection sur les différentes vues de l'objet ; a. cas d'un voxel situé à l'extérieur l'objet ; b. cas d'un voxel situé à l'intérieur de l'objet ; c. cas d'un voxel contenant la surface de l'objet.

- soit le voxel apparaît à l'extérieur de l'objet sur *une* des images : ce cas est illustré par la figure 2.11.a, où la silhouette de l'objet n'est pas intersectée par le voxel sur une des vues ;
- soit le voxel apparaît à l'intérieur de l'objet sur *toutes* les images : ce cas est illustré par la figure 2.11.b, où le voxel est entièrement projeté sur la silhouette de l'objet pour chacune des vues ;
- soit le voxel ne vérifie aucune des deux conditions précédemment citées, il correspond probablement à un voxel contenant la surface de l'objet. Un tel voxel est subdivisé en 8 sous-voxels après sa projection sur toutes les images : voir figure 2.11.c.

Après un premier parcours de toutes les images de l'objet, les sous-voxels créés représentent les voxels de la surface obtenus par sculpture et sont les nouveaux voxels à reprojeter sur les images afin de raffiner itérativement la reconstruction. Cet algorithme fournit ainsi une représentation hiérarchique de l'objet sous la forme d'un octree avec différents niveaux de détails de la surface dépendant du nombre d'itérations. L'algorithme est terminé lorsqu'il n'y a plus aucun

<sup>1</sup>Nous désignerons l'arbre octal par *octree* dans la suite de ce manuscrit.

voxel de surface ou lorsque les voxels nouvellement créés ont atteint une taille minimale définie par l'utilisateur : ces voxels deviennent définitivement des voxels contenant la surface de l'objet.

### B.2.2 Mise en œuvre

#### Structure de données

La forme de l'objet est stockée dans un octree, pointeur sur une structure *NOEUD* détaillée dans l'algorithme 2.1.

```
// Marques applicables sur un noeud de l'octree
#Define LEAF_VOXEL 0x01 // Feuille de l'octree
#Define OUT_VOXEL 0x02 // Voxel à l'extérieur de l'objet
#Define IN_VOXEL 0x04 // Voxel à l'intérieur de l'objet
#Define ON_VOXEL 0x08 // Voxel contenant la surface
#Define OUT_RAYS 0x10 // Voxel projeté sur l'arrière-plan
#Define IN_RAYS 0x20 // Voxel projeté sur la silhouette de l'objet
// Remarque 1 : OUT_VOXEL, IN_VOXEL et ON_VOXEL sont les seules marques utilisées en exclusion
mutuelle ; les autres marques sont cumulables

Struct NOEUD {
    Etat : unsigned char ; // stocke les marques appliquées sur le noeud (jusqu'à 8)
    Contenu : void * ; // diffère selon Etat
}
// Remarque 2 : deux marques restent disponibles dans le champ Etat
```

#### Algorithme 2.1 – Structure de l'arbre octal

Cette structure a été optimisée pour stocker un minimum d'informations par noeud (5 octets + allocation du pointeur *Contenu* en fonction du type de noeud). Voici une liste des différents noeuds existants, suivant la valeur du champ *Etat* :

- 0 : noeud interne, *Contenu* est alloué en un tableau de 8 noeuds (*Struct NOEUD[8]*) pour stocker les 8 sous-voxels le constituant ;
- *LEAF\_VOXEL* & *IN\_VOXEL* : voxel classé définitivement à l'intérieur de l'objet ; *Contenu* est inutilisé ;
- *LEAF\_VOXEL* & *ON\_VOXEL* : voxel contenant définitivement la surface de l'objet ; pour stocker des informations sur la surface, *Contenu* est alloué en une structure *FEUILLE* (cette nouvelle structure sera détaillée ultérieurement) ;
- *LEAF\_VOXEL* & *OUT\_VOXEL* : le voxel est classé définitivement à l'extérieur de l'objet ; *Contenu* est inutilisé ;
- *IN\_VOXEL* : voxel en cours de construction, peut devenir *ON\_VOXEL* ou *OUT\_VOXEL* pendant l'algorithme de sculpture ; *Contenu* est inutilisé ;
- *ON\_VOXEL* : voxel en cours de construction, peut devenir *OUT\_VOXEL* pendant l'algorithme de sculpture ; *Contenu* est inutilisé ;
- *OUT\_VOXEL* : voxel en cours de construction, n'appartenant plus à l'objet ; *Contenu* est inutilisé.

### Initialisation

Lors de la première itération de l'algorithme, l'octree est initialisé avec un voxel défini comme la boîte englobante de l'objet. Les coordonnées de ce voxel sont calculées à partir de la silhouette extraite sur deux images prises avec des directions de visée orthogonales. Pour simplifier les calculs de l'octree, nous utilisons des voxels cubiques.

### Niveaux de détails

A chaque itération de l'algorithme, l'octree est stocké dans un fichier : ainsi les différents niveaux de détails géométriques sont disponibles, correspondant aux différentes profondeurs de l'octree. Ces fichiers peuvent également être chargés lors de l'initialisation pour reprendre la reconstruction plus tard si nécessaire.

### Grille régulière 3D

La projection des voxels sur chacune des images est effectuée par projection inverse en traçant des rayons à travers une grille régulière 3D aux dimensions de la boîte englobante de l'objet. Ainsi, nous projetons non pas les voxels sur les images mais l'ensemble des pixels de chaque image sur la grille régulière 3D. Les calculs sont effectués dans ce sens de projection car la grille régulière permet d'optimiser les calculs d'intersection lors du raffinement du maillage (section B.6). Le calcul de projection est effectué par un parcours incrémental de grille, en suivant les rayons passant par le centre optique de l'image et le centre de chaque pixel : voir figure 2.12. Nous récupérons ainsi les coordonnées de grille de l'ensemble des voxels traversés par chacun

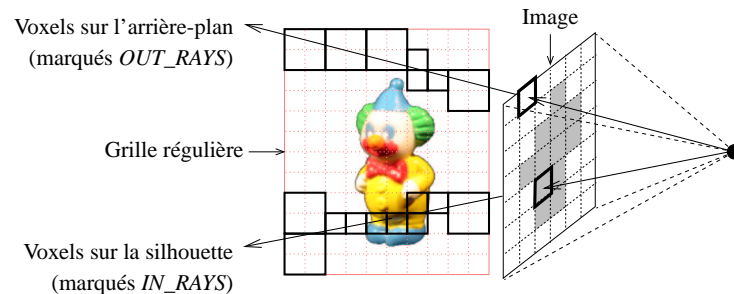


Figure 2.12 – Projection des pixels sur la grille régulière et récupération de la liste des voxels touchés par les rayons

des rayons : le parcours en profondeur de l'octree pour récupérer un voxel est guidé par ses coordonnées de grille.

La grille est échantillonnée en fonction de la profondeur de l'octree en cours de construction. Pour alléger les coûts de stockage, aucune coordonnée n'est stockée dans les voxels, celles-ci sont recalculées à partir de la boîte englobante et du parcours en profondeur dans l'octree. De même, la grille 3D est virtuelle et ne demande aucun espace mémoire.

### Algorithme de sculpture

Le détail de l'algorithme de sculpture utilisé est donné dans l'algorithme [2.2](#).

```
// Initialisation
Calculer la boîte englobante de l'objet et la profondeur de l'octree à construire ;
Initialiser l'octree avec un voxel et le marquer comme ON_VOXEL, ou charger le fichier de reconstruction
le plus complet s'il en existe un ;

// Construction itérative
Tant que la profondeur maximale de l'octree n'est pas atteinte faire
  // Raffinement de la surface
  // Parcours en profondeur de l'octree
  Pour chaque voxel marqué uniquement ON_VOXEL faire
    Subdiviser ce voxel en 8 sous-voxels et les marquer comme IN_VOXEL ;
    // Suppression des marques
    Marquer ce voxel subdivisé comme noeud interne ;
  Fin Pour

  // Parcours de l'ensemble des photographies
  Pour chaque image de l'objet faire
    // Projection de l'image sur l'octree
    Pour chaque pixel de l'image faire
      // Utilisation de la grille régulière 3D
      Récupérer la liste des voxels constituant l'objet traversés par le rayon provenant du pixel
      (voxels marqués LEAF_VOXEL et (IN_VOXEL ou ON_VOXEL)) ;
      Si le rayon traverse la silhouette de l'objet (pixel non noir) alors
        Marquer tous les voxels de la liste comme IN_RAYS ;
      Sinon
        Marquer tous les voxels de la liste comme OUT_RAYS ;
      Fin Si
    Fin Pour

    // Insertion d'un rayon supplémentaire par voxel
    → Détails dans l'algorithme 2.3 ;

  :
```

Algorithme 2.2 – Sculpture itérative de la géométrie basée-voxel

```

:
// Sculpture des voxels de l'objet en fonction des marquages IN_RAYS et OUT_RAYS
// Parcours en profondeur de l'octree
Pour chaque voxel marqué  $\neg$ LEAF_VOXEL et (IN_VOXEL ou ON_VOXEL) faire
    Si le voxel est marqué OUT_RAYS et  $\neg$ IN_RAYS alors
        // Ce voxel est oté de l'objet
        Marquer ce voxel comme OUT_VOXEL ;
    Sinon si le voxel est marqué OUT_RAYS et IN_RAYS alors
        // Ce voxel est placé sur la surface
        Marquer ce voxel comme ON_VOXEL ;
    Fin Si
    Supprimer les éventuelles marques OUT_RAYS et IN_RAYS sur le voxel ;
Fin Pour
Fin Pour

// Maintien de la cohérence de la surface
→ Détails dans l'algorithme 2.4 ;

// Marquage définitif des voxels situés à l'intérieur ou à l'extérieur de l'objet
// Parcours en profondeur de l'octree
Pour chaque voxel marqué uniquement OUT_VOXEL ou IN_VOXEL faire
    Marquer ce voxel comme LEAF_VOXEL ;
Fin Pour

// Sauvegarde sur disque de la reconstruction à chaque itération
Enregistrer la structure de l'octree dans un fichier (ce fichier peut être chargé lors de l'initialisation de
l'octree) ;
Fin Tant que

// Marquage définitif des voxels contenant la surface de l'objet
// Parcours en profondeur de l'octree
Pour chaque voxel marqué uniquement ON_VOXEL faire
    Marquer ce voxel comme LEAF_VOXEL ;
Fin Pour

```

Algorithme 2.2 – Sculpture itérative de la géométrie basée-voxel

Après chaque itération de l'algorithme, les voxels contenant la surface de l'objet sont tous de la même taille. Ces voxels étant raffinés en début de chaque itération, ils ne sont définis qu'au plus bas niveau de l'octree. En revanche, ce n'est pas le cas des voxels classés définitivement à l'intérieur ou à l'extérieur de la surface.

### B.2.3 Analyse des résultats

#### Voxels marqués

Lorsque les voxels sont de trop petite taille, aucun rayon n'est projeté dessus, malgré l'utilisation de l'ensemble des images : un exemple avec la projection d'une rangée de pixels est montré sur la figure 2.13.a. Lorsqu'aucun rayon n'est projeté sur un voxel, celui-ci ne contient ni

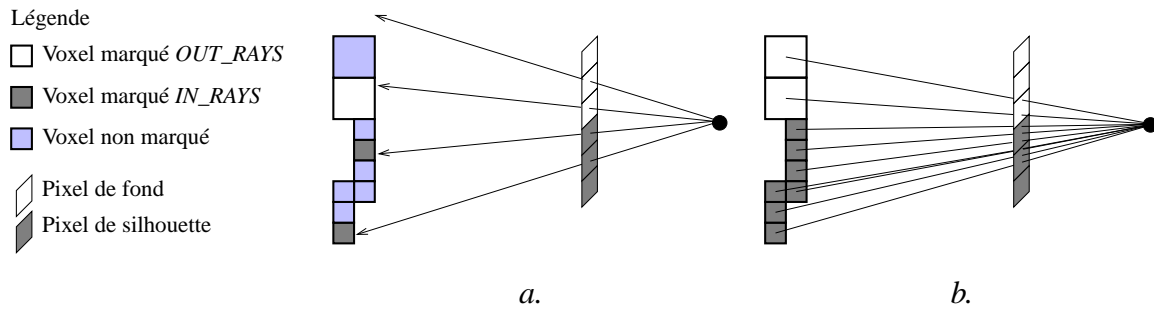


Figure 2.13 – a. Projection des centres des pixels sur l’octree. Les voxels non atteints ne sont pas marqués et deviennent par la suite des voxels marqués *IN\_VOXEL*; b. Projection des centres des voxels sur les images. Ainsi, tous les voxels sont correctement marqués.

marque *IN\_RAYS*, ni marque *OUT\_RAYS*. Par conséquent, le voxel ne change pas de marquage lors de l’étape de *sculpture des voxels* (voir l’algorithme 2.2), il reste dans son état d’origine et devient *IN\_VOXEL* plus tard, i.e. un voxel à l’intérieur de l’objet lors de l’extraction de la surface discrète. Ceci est gênant car le voxel peut être en réalité un voxel situé à l’extérieur de l’objet, auquel cas la surface reconstruite est déviée localement d’un ou plusieurs voxels.

```
// Parcours en profondeur de l’octree
Pour chaque voxel marqué uniquement IN_VOXEL ou ON_VOXEL faire
    // Projection du centre du voxel sur l’image
    Tracer un rayon passant par le centre du voxel et le centre optique de l’image ;
    Si le voxel intersecte un pixel de l’image alors
        Marquer le voxel comme OUT_RAYS / IN_RAYS en fonction de la couleur du pixel intersecté
        (noir / non noir) ;
    Sinon
        // Voxel invisible sur l’image
        Marquer le voxel comme OUT_RAYS ;
    Fin Si
Fin Pour
```

### Algorithme 2.3 – Marquage de chaque voxel

Pour résoudre ce problème, le centre de tous les voxels en cours de construction est projeté sur chaque image afin d’obtenir au minimum une marque (*OUT\_RAYS* / *IN\_RAYS*) par voxel. L’algorithme 2.2 est complété à l’étape d’*insertion d’un rayon supplémentaire par voxel* par l’algorithme 2.3. Ainsi, un voxel disposant d’au moins un marquage par image traitée (voir figure 2.13.b) est correctement sculpté après chaque projection d’image.

### Voxels de l’objet

La géométrie de l’objet s’affine au fur et à mesure des itérations de l’algorithme de sculpture. Un exemple de reconstruction basée-voxel d’un objet réel (le clown) est montré sur la figure 2.14 à partir d’une série d’images de résolution  $256 \times 256$ . Il est inutile de construire un modèle géométrique trop précis où les voxels de la surface sont plus petits que les pixels des images.



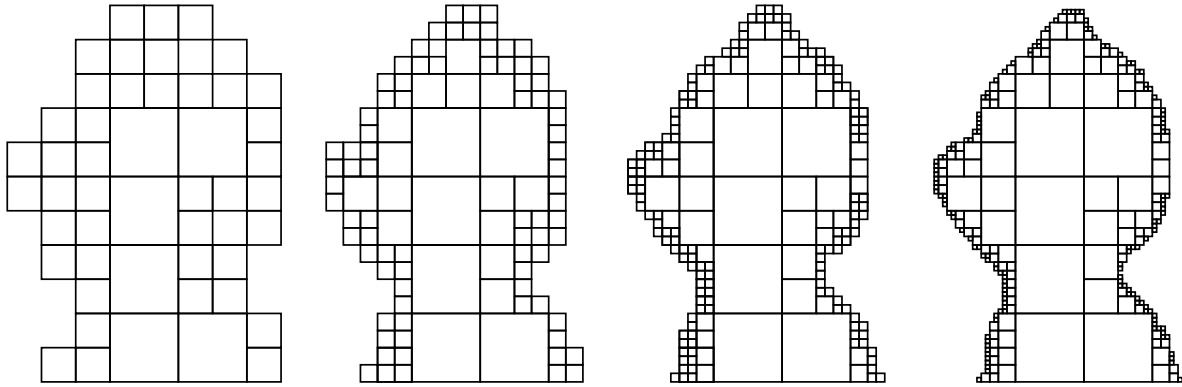


Figure 2.14 – Coupe 2D de l'octree représentant la géométrie d'un objet réel (tête d'un clown) avec une profondeur d'octree variant, de gauche à droite, entre 5 et 8. Les voxels situés à l'extérieur de l'objet ne sont pas représentés pour plus de clarté.

Ainsi, pour des images de résolution  $256 \times 256$ , la construction de l'octree est limitée à une profondeur de 8 ( $256 = 2^8$ ).

En effet, lorsque les voxels sont suffisamment petits, ils sont automatiquement classés en voxel à l'intérieur ou à l'extérieur de l'objet, et n'ont plus besoin d'être raffinés. Ceci s'explique par le faible nombre de rayons projetés sur chaque voxel. Comme nous l'avons vu précédemment, en dessous d'une certaine taille, les voxels ne disposent que d'un seul marquage par image :

- s'il est *OUT\_RAYS* sur une image alors le voxel est définitivement classé à l'extérieur de l'objet ;
- sinon il est *IN\_RAYS* sur toutes les images et le voxel est définitivement classé à l'intérieur de l'objet.

La figure 2.15 est une visualisation 3D de l'exemple de reconstruction présenté sur la figure 2.14. La silhouette de l'objet devient de plus en plus précise. Au niveau de profondeur 8, chaque voxel reçoit environ 1 pixel de chaque image puisque la hauteur de la silhouette de l'objet est représentée par environ 250 pixels.

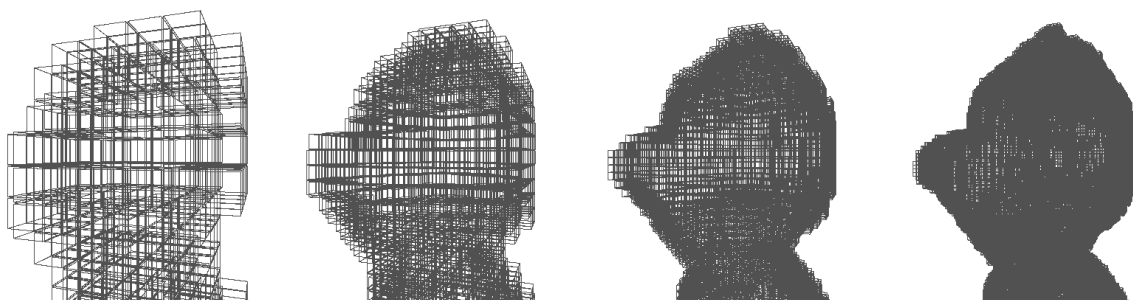


Figure 2.15 – Vue 3D de l'octree représentant la géométrie d'un objet réel (tête d'un clown) avec une profondeur d'octree variant, de gauche à droite, entre 5 et 8.

### Voxels de la surface

Les voxels correspondant à la surface de l'objet (marqués *ON\_VOXEL*) disparaissent à partir d'une certaine profondeur de l'octree. Nous avons vu précédemment que ce phénomène est dû au faible nombre de marques *IN\_RAYS* / *OUT\_RAYS* inscrites dans les voxels et que l'insertion d'un rayon supplémentaire par voxel permet de reconstruire une géométrie cohérente de l'objet.

La surface actuelle de l'objet n'est pas représentée uniquement par des voxels 3D, elle est également définie par des faces communes à deux voxels : polygones 2D entre l'intérieur et l'extérieur de l'objet, appelés surfels. La figure 2.16 montre que la surface est définie différemment d'une profondeur d'octree à une autre et que le raffinement de la géométrie basée-voxels tend vers une surface représentée uniquement par des voxels (figure c). Or, nous ne voulons généralement

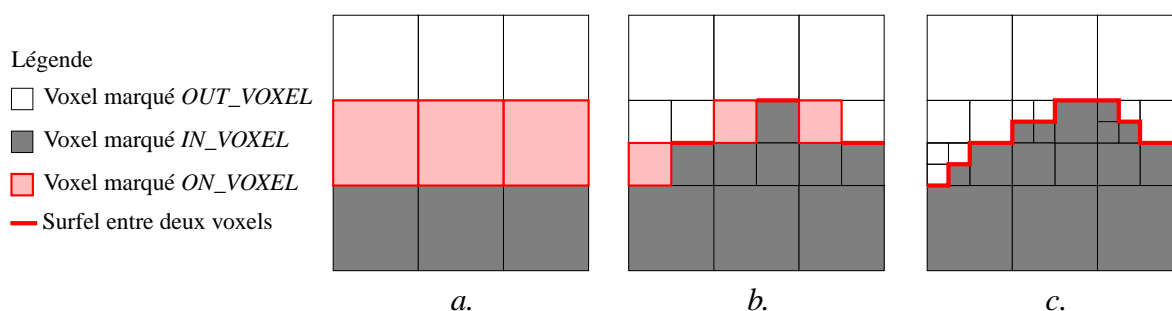


Figure 2.16 – Exemple de reconstruction basée-voxel dépendant de la profondeur de l'octree. La surface est représentée par ; a. uniquement des voxels ; b. des voxels et des faces inter-voxels (surfels) ; c. uniquement des surfels.

pas reconstruire un octree très précis afin de limiter le coût mémoire de la reconstruction géométrique. Par conséquent, nous proposons des solutions dans la section suivante afin de construire une surface discrète homogène, basée sur des voxels et qui soit également 6-connexe afin de pouvoir appliquer l'algorithme des *marching cubes*.

## B.3 Construction de la surface discrète

### B.3.1 Choix de la représentation

Comme la surface discrète est définie par des éléments de différentes dimensions (surfels et voxels), le calcul d'intersection d'un rayon avec la surface est complexe dans la structure en octree. Ce problème est amplifié par un raffinement des voxels qui s'effectue uniquement sur les voxels classés *ON\_VOXEL*. Par conséquent, les voxels classés *IN\_VOXEL* ou *OUT\_VOXEL* sont de tailles variées et perturbent la représentation de la surface entre deux voxels. Par exemple sur la figure 2.16.b, un voxel marqué *IN\_VOXEL* possède une face adjacente à un voxel de plus grande taille, marqué *OUT\_VOXEL*. Lors du calcul d'intersection, l'utilisation de la face du voxel touché ne suffit plus à obtenir une représentation de la surface, le voxel adjacent doit également être utilisé pour connaître la représentation exacte de la surface (dans cet exemple, la surface de l'objet est plus petite que la face du voxel marqué *OUT\_VOXEL*). Une autre solution est de stocker la surface explicitement dans le voxel mais dans les deux cas, la représentation de la surface devient coûteuse en espace et en temps.

Par conséquent, une représentation homogène de la surface au sein des voxels est préférable. Le raffinement des voxels de la surface jusqu'à leur disparition définitive entraîne de multiples itérations de l'algorithme de sculpture et un coût de stockage important dans l'octree. Nous abandonnons donc cette hypothèse au profit d'une représentation de la surface par des voxels de plus grande taille. Nous améliorons ensuite l'algorithme des *marching cubes* [LC87] afin de générer un maillage polygonal fermé de la surface de l'objet qui soit aussi fidèle que possible à la surface réelle de l'objet.

### B.3.2 Principe

L'ensemble des voxels marqués *ON\_VOXEL*, représentant la surface discrète, a besoin d'être 6-connexe pour faciliter la construction d'un maillage de la surface. En effet, l'algorithme des *marching cubes* exige la 6-connexité des voxels pour produire un maillage *fermé* (i.e. un maillage sans bord). Si l'ensemble de voxels est constitué de plusieurs composantes 6-connexes (exemple de l'acquisition d'un objet non connexe), l'algorithme de construction de maillage peut être appliqué sur chaque composante 6-connexe : ainsi, un maillage fermé est généré pour chaque composante.

L'algorithme de sculpture 2.2 est modifié afin de produire une surface discrète 6-connexe. Afin d'éviter les aberrations lors de la reconstruction géométrique et de rester le plus fidèle possible à l'objet réel, la 6-connexité de la surface est vérifiée et maintenue à chaque itération de l'algorithme de sculpture : modification à l'étape de *maintien de la cohérence de la surface* (algorithme 2.2).

### B.3.3 Mise en œuvre

Même si la surface discrète est 6-connexe au début de chaque itération, la subdivision de chaque voxel de la surface en 8 voxels marqués *IN\_VOXEL* sculptés ultérieurement peut ôter la propriété de *fermeture* de la surface discrète. En effet, les bords de la surface apparaissent sur la figure 2.16.b après subdivision des voxels marqués *ON\_VOXEL* sur la figure 2.16.a.

La surface 6-connexe doit vérifier la propriété des *surfaces de Jordan* [Her92] en 26-connexité, i.e. tout chemin 26-connexe reliant un voxel marqué *IN\_VOXEL* à un voxel marqué *OUT\_VOXEL* doit passer par un voxel marqué *ON\_VOXEL*. En d'autres termes, ceci signifie qu'il ne peut pas y avoir deux voxels 26-adjacents (adjacents en 26-connexité) dont l'un est marqué *IN\_VOXEL* et l'autre *OUT\_VOXEL*. Cette propriété est vérifiée dans notre approche à chaque itération de l'algorithme de sculpture.

#### Voxels adjacents de même taille

Lorsque deux voxels 26-adjacents sont de taille égale (dont l'un est marqué *IN\_VOXEL*, l'autre *OUT\_VOXEL*), le voxel marqué *IN\_VOXEL* est changé en *ON\_VOXEL*. Nous choisissons de modifier le voxel situé à l'intérieur de l'objet car nous ne sommes pas certain de l'exactitude de ce marquage. En effet, l'algorithme initial sculpte les voxels en projetant la silhouette de l'objet sur chaque voxel : un voxel vu à l'extérieur de l'objet est alors marqué *OUT\_VOXEL* avec certitude. Lorsqu'aucune silhouette ne permet de modifier le marquage d'un voxel, celui-ci reste inchangé (marqué initialement *IN\_VOXEL* lors du raffinement). Les deux voxels 26-adjacents correspondent toujours à des voxels situés au plus bas niveau de l'octree car ils sont vérifiés à

chaque itération. La figure 2.17 montre un exemple de modification de la surface discrète pour des voxels adjacents de même taille.

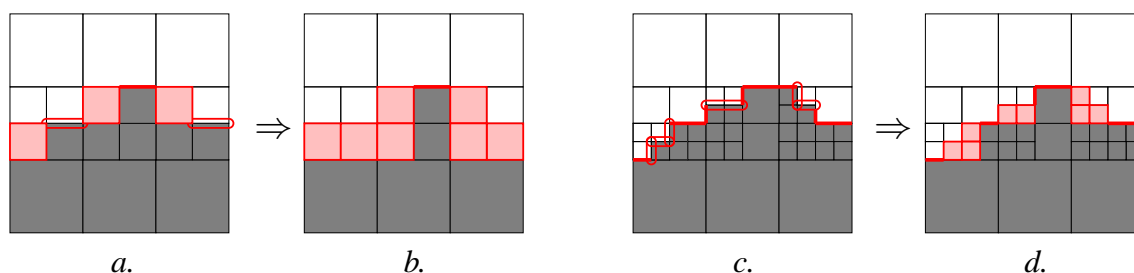


Figure 2.17 – Modification du marquage du voxel *IN\_VOXEL* en *ON\_VOXEL* lorsqu'il est 26-adjacent à un voxel *OUT\_VOXEL* de même taille ; a. et c. détection des régions concernées ; b. et d. modification du marquage.

### Voxels adjacents de taille différente

Lorsque deux voxels 26-adjacents sont de taille différente (dont l'un est marqué *IN\_VOXEL*, l'autre *OUT\_VOXEL*), le voxel le plus petit est changé en *ON\_VOXEL*. Nous choisissons le voxel le plus petit car il provient (à une ou plusieurs itérations précédentes) d'un voxel marqué *ON\_VOXEL* de la même taille que le plus grand des deux. Nous privilégions ainsi la cohérence entre les différents niveaux de profondeur de l'octree. De plus, ceci évite de faire des subdivisions successives sur des éléments de l'octree déjà traités. En effet, le voxel de plus grande taille a déjà été classé définitivement comme une feuille de l'octree. Le changement de marquage se fait à nouveau dans un voxel du plus bas niveau de l'octree (sinon la modification aurait été faite lors d'une des itérations précédentes de l'algorithme de sculpture). La figure 2.18 montre la fermeture sur un morceau de surface discrète.

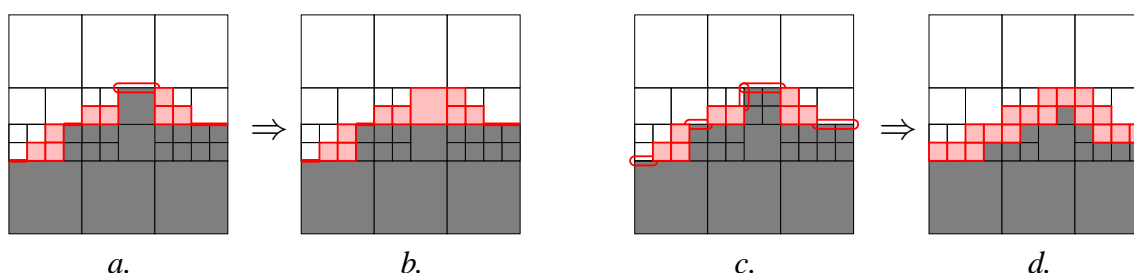


Figure 2.18 – Modification du marquage du voxel le plus petit en *ON\_VOXEL* lorsque un voxel *IN\_VOXEL* et un voxel *OUT\_VOXEL* sont 26-adjacents mais de taille différente ; a. détection de la région en même temps que celles de la figure 2.17.a ; c. détection des régions à modifier en même temps que celles de la figure 2.17.c ; b. et d. modification du marquage et obtention d'une surface 6-connexe en d (avec un octree partiellement représenté).

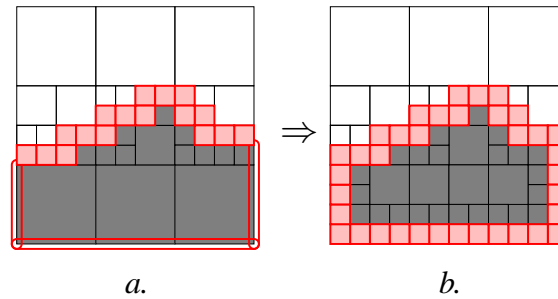


Figure 2.19 – Sur cet exemple, l’octree est entièrement représenté même si le nombre de subdivisions n’est pas une puissance de 2 afin de pouvoir modifier le marquage des voxels *IN\_VOXEL* situés sur le bord de l’octree ; a. détection des régions concernées ; b. modification du marquage et obtention d’une surface fermée après subdivisions successives.

### Voxels sur le bord de l’octree

Tous les voxels de l’objet (marqués *IN\_VOXEL*) situés sur le bord de l’octree sont transformés en voxels contenant la surface de l’objet (marqués *ON\_VOXEL*). Ceci permet la fermeture de la surface discrète dans toutes les situations : voir figure 2.19.

### Algorithme de construction de la surface discrète

L’algorithme 2.4 donne le détail de la construction de la surface discrète 6-connexe. Il est inséré dans l’algorithme de sculpture 2.2 et ne traite que les voxels situés au plus bas niveau de l’octree.

La reconstruction de la surface discrète peut être effectuée par un seul parcours de l’octree. Cependant nous préférons effectuer deux parcours successifs afin de modifier en priorité les voxels situés à l’extérieur de l’objet. Ce processus évite d’épaissir inutilement la surface discrète de l’objet tout en préservant sa 6-connexité (voir la section B.3.4 sur les résultats pour plus de détails).

## B.3.4 Analyse des résultats

### Choix de l’algorithme de construction de la surface

Si la reconstruction de la surface discrète était effectuée, dans l’algorithme 2.4, par un seul parcours de l’octree, alors le choix de parcours des voxels 26-adjacents aurait été fait en fonction de la construction de l’octree et aurait pu conduire à la construction d’une surface anormalement épaissie. La modification des voxels *OUT\_VOXEL* avant les voxels *IN\_VOXEL* permet de réduire l’épaisseur de la surface 6-connexe construite : la figure 2.20 montre le problème survenant lors d’un seul parcours de l’octree.

### Contrôle de l’épaisseur de la surface

En dépit de toutes les précautions prises lors de la construction de la surface discrète, celle-ci peut toutefois apparaître trop épaisse et perturber l’algorithme de construction du maillage (section B.5). En effet, sans réduction de l’épaisseur de la surface discrète, certains voxels de

```
// Modification des voxels situés à l'extérieur de l'objet
// Parcours en profondeur de l'octree
Pour chaque voxel V marqué uniquement OUT_VOXEL faire
    // Utilisation de la grille régulière 3D
    Récupérer la liste des voxels 26-adjacents à ce voxel ;
    Si au moins un voxel de la liste est marqué LEAF_VOXEL et IN_VOXEL alors
        | Marquer le voxel V comme ON_VOXEL ;
    Fin Si
Fin Pour

// Modification des voxels situés à l'intérieur de l'objet
// Parcours en profondeur de l'octree
Pour chaque voxel V marqué uniquement IN_VOXEL faire
    // Utilisation de la grille régulière 3D
    Si le voxel est situé sur le bord de l'octree alors
        | Marquer le voxel V comme ON_VOXEL ;
    Sinon
        | Récupérer la liste des voxels 26-adjacents à ce voxel ;
        | Si au moins un voxel de la liste est marqué OUT_VOXEL alors
            | Marquer le voxel V comme ON_VOXEL ;
        | Fin Si
    Fin Si
Fin Pour
```

Algorithme 2.4 – Construction d'une surface 6-connecte

la surface ne sont pas utilisés dans la construction du maillage et la surface polygonale produite peut avoir des propriétés très différentes de la surface discrète (changement du nombre de composantes connexes, etc).

Le maintien de la 6-connectivité de la surface discrète, détaillé à la section [B.3.3](#), est effectué par transformation de voxels situés à l'intérieur de l'objet en voxels contenant la surface. Ceci permet de ne pas toucher au marquage des voxels situés à l'extérieur de l'objet et de respecter le plus possible la forme de l'objet définie par l'algorithme de sculpture.

Le même raisonnement peut être appliqué afin de réduire l'épaisseur de la surface discrète : lorsque la surface est trop épaisse, les voxels de la surface discrète définis comme *internes* à l'objet peuvent être transformés en voxels à l'intérieur de l'objet, i.e. n'appartenant plus à la surface (voir figure [2.21](#)). Ainsi, la forme extérieure de l'objet est maintenue inchangée. Cette nouvelle modification du marquage est applicable après l'algorithme de sculpture itérative afin de ne pas créer d'interférence pendant le raffinement de l'octree. La section suivante ([B.4](#)) détaille l'algorithme d'amincissement et expose les conséquences de telles modifications sur les cas extrêmes (surfaces étroites, objets quasi-déconnectés, etc).

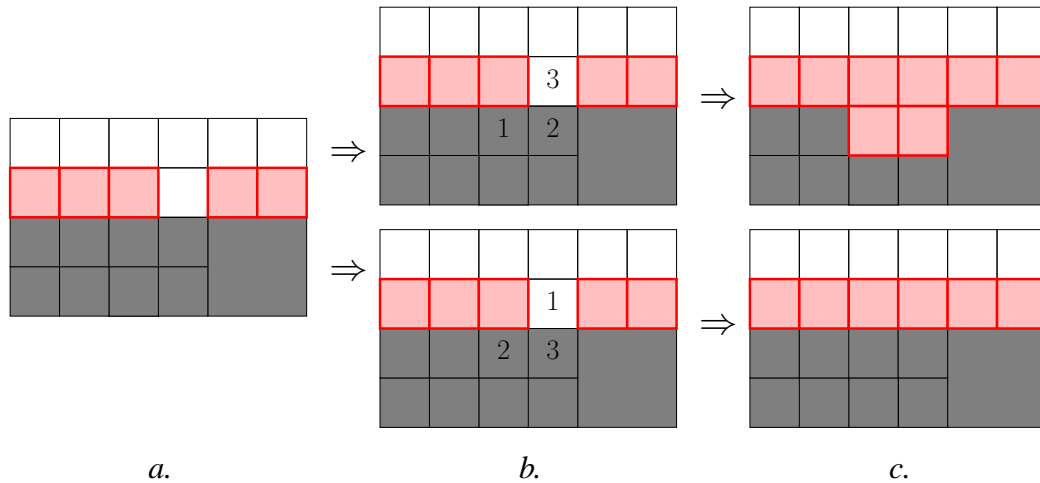


Figure 2.20 – Suivant l’ordre de parcours des voxels, la reconstruction d’une surface 6-connecte par un seul parcours de l’octree peut produire des résultats différents ; a. surface avant l’utilisation de l’algorithme de construction d’une surface 6-connecte ; b. deux ordres différents de parcours des voxels de l’octree ; c. en haut, les voxels 1 et 2 deviennent *ON\_VOXEL* (26-adjacents au voxel 3) puis le voxel 3 devient *ON\_VOXEL* (26-adjacent à un voxel *IN\_VOXEL* de plus grande taille) afin de rendre la surface 6-connecte ; en bas, le voxel 1 devient *ON\_VOXEL* (26-adjacent à un voxel *IN\_VOXEL* de plus grande taille) et la surface est déjà 6-connecte.

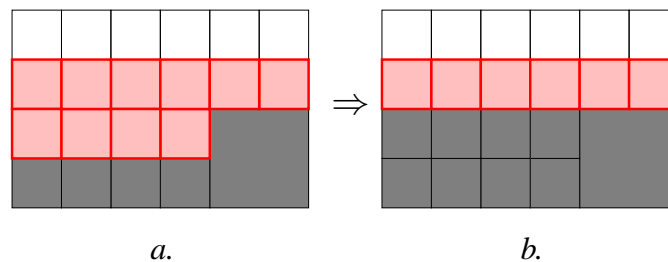


Figure 2.21 – Amincissement de la surface discrète tout en maintenant la 6-connectivité ; a. avant amincissement ; b. après amincissement, suppression des voxels de la surface définis comme internes à l’objet.

## B.4 Amincissement de la surface discrète

L’amincissement de la surface discrète est une optimisation permettant à l’algorithme des *marching cubes* de générer un maillage continu entre les voxels 6-adjacents : la 6-connectivité de la surface doit donc être préservée.

### B.4.1 Principe

Le traitement est effectué uniquement sur les voxels marqués *ON\_VOXEL*, (représentant la surface discrète) et utilise une représentation de chaque voxel par ses 8 sommets. Les sommets sont séparés en deux classes : ceux représentant des points 3D situés à l’extérieur de l’objet et ceux représentant des points 3D à l’intérieur. Ainsi le maillage reconstruit à la section B.5 est défini comme une séparation des deux classes de sommets. L’amincissement de la surface discrète



a pour but de supprimer les voxels inutilisés dans la génération du maillage, et prioritairement les voxels les plus internes à l'objet. Il suffit donc de regarder la classification des 8 sommets de chaque voxel et de modifier les voxels dont tous les sommets appartiennent à la même classe.

### B.4.2 Mise en œuvre

#### Classification des sommets

Les sommets des voxels de la surface sont classés en fonction de leur 26-voisinage. Comme la surface est une surface de Jordan, il n'existe pas de chemin 26-connexe permettant de relier un voxel *IN\_VOXEL* et un voxel *OUT\_VOXEL* sans passer par un voxel de la surface (*ON\_VOXEL*). Ceci signifie en particulier qu'un sommet d'un voxel de la surface ne peut pas appartenir à la fois à un voxel marqué *IN\_VOXEL* et un voxel marqué *OUT\_VOXEL* (ces deux derniers voxels ne peuvent être adjacents). L'étape de classification est ainsi simplifiée et le nombre de possibilités pour chaque sommet est réduit à :

- soit le sommet est partagé par au moins un voxel marqué *OUT\_VOXEL* et est considéré comme un sommet à l'extérieur de la surface ;
- soit le sommet est partagé par au moins un voxel marqué *IN\_VOXEL* et est considéré comme un sommet à l'intérieur de la surface ;
- soit le sommet n'est partagé que par des voxels marqués *ON\_VOXEL*.

Un sommet peut n'être partagé que par des voxels de la surface dans le cas de surface localement trop épaisse. Pour préserver la forme discrète de l'objet, un tel sommet est considéré à l'intérieur de l'objet.

#### Algorithme d'amincissement

L'amincissement de la surface n'est effectué qu'en fin de construction de la surface discrète, comme prétraitement indispensable à la construction d'un maillage et à l'estimation d'un vecteur normal par voxel de la surface (voir algorithme 2.5).

Un seul parcours de l'octree est nécessaire pour amincir la surface. En effet, même si la surface est très épaisse, tous les voxels inutiles à la construction du maillage sont éliminés en un parcours. Il existe deux cas de suppression de voxels de la surface :

- soit tous les sommets du voxel *V* sont classés à l'extérieur de la surface ; auquel cas, chaque sommet appartient à au moins un voxel classé à l'extérieur de l'objet et la modification du voxel *V* en voxel à l'extérieur de l'objet ne change pas la classification des sommets ;
- soit aucun sommet du voxel *V* n'est classé à l'extérieur de la surface ; auquel cas, chaque sommet n'appartient à aucun voxel classé à l'extérieur de l'objet et la modification du voxel *V* en voxel à l'intérieur de l'objet ne change pas la classification des sommets.

Dans les deux cas, la modification locale d'un voxel de la surface n'a aucune influence sur une éventuelle modification des voxels adjacents (la classification des sommets reste inchangée).

La 6-connexité de la surface est maintenue après application de l'algorithme d'amincissement. En effet, la section B.3 montre la construction d'une surface 6-connexe, quel que soit le niveau de raffinement des voxels de l'octree. Ensuite, l'algorithme d'amincissement de cette surface fournit une classification des sommets sans aucune ambiguïté puisque deux voxels dont l'un est marqué *IN\_VOXEL* et l'autre *OUT\_VOXEL* ne peuvent jamais être 26-adjacents. Comme la modification du marquage de certains voxels de la surface ne modifie jamais la classification des sommets, il n'existe pas de voxels 26-adjacents dont l'un soit marqué *IN\_VOXEL* et l'autre

```

// Modification des voxels de la surface situés trop à l'intérieur ou à l'extérieur de l'objet
// Parcours en profondeur de l'octree
Pour chaque voxel V marqué ON_VOXEL faire
    // Utilisation de la grille régulière 3D
    Récupérer la liste des voxels 26-adjacents à ce voxel ;
    // Classification des sommets du voxel V
    Pour chaque sommet du voxel V faire
        Si le sommet est situé sur le bord de l'octree alors
            | Classer le sommet à l'extérieur de la surface ;
        Sinon
            | Récupérer la sous liste des voxels contenant le sommet ;
            | Si au moins un voxel de la sous-liste est marqué OUT_VOXEL alors
            | | Classer le sommet à l'extérieur de la surface ;
            | Sinon
            | | Classer le sommet à l'intérieur de la surface ;
            | Fin Si
        Fin Si
    Fin Pour
    // Modification du marquage du voxel
    Si tous les sommets du voxel V sont classés à l'extérieur de la surface alors
        | Marquer le voxel V comme OUT_VOXEL ;
    Sinon si tous les sommets du voxel V sont classés à l'intérieur de la surface alors
        | Marquer le voxel V comme IN_VOXEL ;
    Sinon
        // Allocation mémoire
        | Allouer une structure FEUILLE pour le stockage d'informations sur la surface locale à V ;
        | Stocker la classification des sommets dans cette structure ;
    Fin Si
Fin Pour

```

Algorithme 2.5 – Amincissement de la surface discrète

*OUT\_VOXEL*. Par conséquent, la surface discrète reste obligatoirement 6-connexe après amincissement.

### B.4.3 Analyse des résultats

#### Composantes connexes

Les petits objets d'un diamètre (épaisseur) inférieur au voxel sont supprimés (voir figure 2.22.a). A partir de deux voxels de diamètre, les objets sont conservés, même s'ils ne sont pas composés de voxels marqués *IN\_VOXEL* (voir figure 2.22.b)

Un objet de faible épaisseur peut également être déconnecté (voir figure 2.23.a) et deux composantes de l'objet proches l'une de l'autre peuvent être fusionnées en une seule (voir figure 2.23.b).

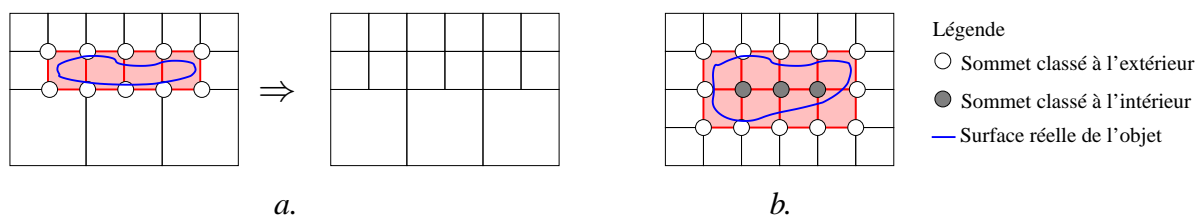


Figure 2.22 – Amincissement de la surface discrète tout en maintenant la 6-connectivité ; a. suppression d'une composante de l'objet ; b. objet inchangé.

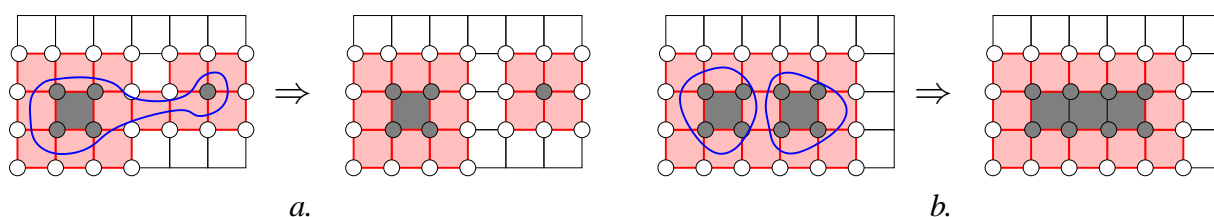


Figure 2.23 – Amincissement de la surface discrète tout en maintenant la 6-connectivité ; a. déconnexion de l'objet en deux composantes 6-connexes ; b. fusion de deux composantes connexes.

### Trous

De manière similaire, un trou ne contenant aucun voxel marqué *OUT\_VOXEL* est supprimé de l'objet (voir figure 2.24.a), un trou de faible diamètre peut être déconnecté et deux trous dans l'objet, proches l'un de l'autre, peuvent être fusionnés en un seul (voir figure 2.24.b).

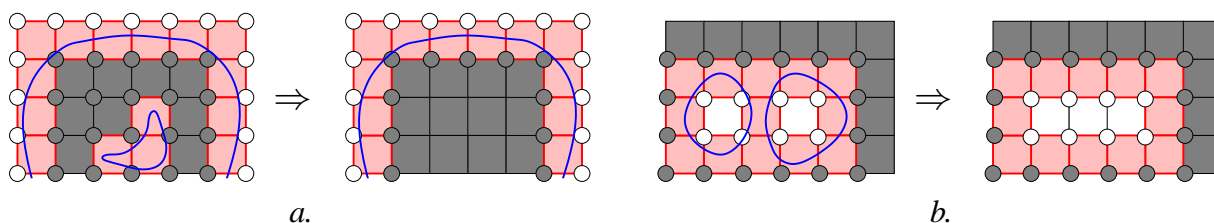


Figure 2.24 – Amincissement de la surface discrète tout en maintenant la 6-connectivité ; a. suppression d'un trou dans l'objet ; b. fusion de deux trous.

Toutes les modifications des propriétés de l'objet (nombre de composantes connexes, nombre de trous) sont dues à des erreurs d'imprécision de l'ordre du voxel. Par conséquent, un raffinement de la surface discrète suffit à supprimer ces cas problématiques.

### Bosses

Les bosses définies à la surface de l'objet sont conservées lors de l'amincissement de la surface, même si elles épaississent la surface d'un voxel supplémentaire (voir figure 2.25). Uniquement les petites bosses représentées par un seul voxel sont supprimées de la surface discrète.



Figure 2.25 – Amincissement de la surface discrète tout en maintenant la 6-connexité ; a. les petites bosses de la surface sont conservées ; b. en revanche, les micro-bosses d'un seul voxel sont supprimées.

### Cavités

En revanche, les cavités d'un diamètre inférieur à deux voxels sont supprimées de la surface discrète puisqu'elles correspondent à des voxels disposant uniquement de sommets classés à l'intérieur de l'objet (voir figure 2.26).

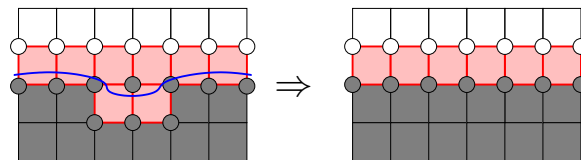


Figure 2.26 – Amincissement de la surface discrète tout en maintenant la 6-connexité ; les petites cavités de la surface sont supprimées.

L'algorithme de sculpture utilisé précédemment reconstruit les bosses existantes à la surface de l'objet mais n'est pas adapté à la reconstruction des cavités. Par conséquent lors de l'amincissement de la surface, nous avons privilégié le classement des sommets ambigus en sommets situés à l'intérieur de l'objet afin de favoriser le maintien des bosses de la surface au détriment des cavités. Certaines cavités peuvent toutefois apparaître, correspondant essentiellement à des rainures sculptées par des pixels de fond des images. Pour conserver ces petites cavités, il suffit à nouveau de poursuivre l'algorithme de sculpture et ainsi de raffiner la surface discrète.

## B.5 Génération de maillage fermé

L'algorithme des *marching cubes* [LC87], appliqué sur un ensemble de voxels 6-connexes, est une méthode classiquement utilisée en imagerie médicale pour reconstruire une surface entre deux milieux. Par exemple, une série d'images issues d'un tomographe peut être utilisée (coupes 2D d'un volume 3D) pour régénérer la surface des organes et ainsi modéliser les organes internes d'un corps humain. Cette méthode est particulièrement bien adaptée à notre approche puisque nous disposons d'une surface discrète représentée par des voxels 6-connexes.

### B.5.1 Principe

Les sommets des voxels contenant la surface sont classés en fonction de leur appartenance ou non à l'objet. Ensuite, une surface est construite localement dans chaque voxel en répartissant les sommets de part et d'autre de cette surface, suivant leur classification. La surface coupe

toutes les arêtes composées d'un sommet à l'extérieur et d'un autre à l'intérieur de l'objet. Il existe 256 configurations différentes pour la classification de 8 sommets (2 choix par sommet, 8 sommets par voxel). Ce nombre de configurations a été réduit à 15 dans [LC87] en utilisant les propriétés de symétrie sur un voxel (configurations identiques après rotation du voxel et/ou utilisation de propriétés duales entre les sommets classés à l'intérieur / à l'extérieur de l'objet). Ces 15 configurations sont illustrées sur la figure 2.27 : la surface de l'objet est construite localement dans chaque voxel en utilisant la configuration adéquate. Cet algorithme génère ainsi entre 0 et 4 triangles par voxel, indépendamment des voxels adjacents.

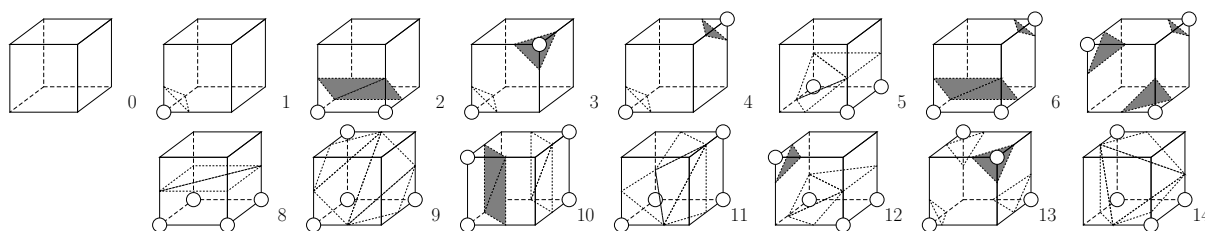


Figure 2.27 – Les 15 configurations de voxel utilisés par W. Lorensen et H. Cline dans l'algorithme des *marching cubes* [LC87].

Néanmoins, une pondération est généralement définie sur les sommets afin d'avoir une estimation de la distance séparant chaque sommet de la surface réelle. Cette pondération est évaluée à partir des pixels des coupes 2D, correspondant à des valeurs de densité de l'intérieur des volumes à modéliser. Elle guide l'algorithme des *marching cubes* afin de produire un maillage de triangles proche de la surface réelle de l'objet.

Lors de l'utilisation de photographies et non de coupes 2D de l'objet, aucune information de densité de volume ne peut être extraite depuis une photographie. Seulement des informations géométriques sur la surface sont extraites à partir de la silhouette de l'objet. Quant aux autres pixels de l'objet, ils fournissent des informations sur la réflectance de la surface. Lorsqu'aucune information ne permet de définir précisément les points d'intersection entre les triangles produits et les arêtes des voxels, le maillage généré passe par le milieu des arêtes. Cette solution est souvent envisagée lorsque l'objet est défini par une série de photographies. Par exemple, J-M. Hasenfratz et al. utilisent une telle méthode de reconstruction pour intégrer interactivement une personne réelle dans un environnement virtuel [HLGB03].

Nous proposons également un maillage passant par le centre des arêtes pour poursuivre cette étude sur la construction d'un maillage fermé mais cette pondération sur les arêtes est provisoire, elle est modifiée dans la section B.6 sans altération de la propriété de fermeture du maillage généré. En effet, la section B.6 décrit notre méthode de raffinement de maillage utilisant les pixels des images comme information complémentaire.

### B.5.2 Mise en œuvre

#### Structure de données

Pour générer le maillage polygonal avec l'algorithme des *marching cubes*, des informations géométriques doivent être stockées dans chaque feuille de l'octree.

L'algorithme 2.6 répertorie les informations utiles au stockage du maillage. Pour réduire les coûts, les coordonnées des triangles ne sont pas enregistrées, elles sont recalculées à partir de

```

// Marques applicables sur un sommet d'un voxel de la surface
#Define OUT_VERTEX 1 // Sommet à l'extérieur de l'objet
#Define IN_VERTEX 0 // Sommet à l'intérieur de l'objet

Struct FEUILLE {
    Classification : unsigned char ; // stocke les marques appliquées sur les 8 sommets
    Ponderation : float [12] ; // stocke le point d'intersection avec le maillage pour chaque arête
    ...
}
// Remarque : le nombre de valeurs de ponderation réellement stocké dépend de la classification

```

Algorithme 2.6 – Structure d'un voxel de la surface, marqué ON\_VOXEL.

l'algorithme des *marching cubes*. Ainsi, moins de 50 octets sont nécessaires par voxel contenant la surface contre 144 octets pour le stockage de 4 triangles (une bande de 4 triangles peut être représentée par 6 sommets avec 3 coordonnées exprimées comme *double*) :

- 1 octet pour stocker la classification des sommets du voxel (8 booléens) ;
- 48 octets pour stocker la pondération de chaque sommet. La pondération est stockée pour chaque arête du voxel avec une valeur comprise entre 0.0 et 1.0 ; comme chaque arête d'un voxel est commune à quatre voxels de la surface discrète, une même pondération est stockée 4 fois dans l'octree (en fait, 12 octets suffisent). Nous préférons tout de même cette redondance car elle permet d'une part d'accélérer les calculs de reconstruction de maillage et d'autre part de simplifier le raffinement du maillage effectué dans la section B.6 ;
- d'autres champs sont définis ultérieurement, notamment pour stocker la normale et la réflectance en chaque voxel.

L'allocation de la structure *FEUILLE*, le calcul et le stockage de la classification des sommets est déjà expliqué dans l'algorithme 2.5. Le stockage de la pondération n'est pas nécessaire a priori puisque la pondération est identique sur chaque sommet. Ce stockage rend cependant la génération du maillage plus évolutive : un raffinement du maillage est proposé dans la section B.6, avec utilisation de pondérations différentes entre les sommets d'un même voxel.

### Algorithme de génération de triangles

Des tables sont construites pour accélérer la génération du maillage. Une première table, indexée sur 14 des 15 configurations de voxel présentées sur la figure 2.27, permet d'évaluer les coordonnées des sommets des triangles en utilisant les coordonnées du voxel et les valeurs de pondération stockées pour chaque arête : une addition et une multiplication suffisent pour calculer chaque sommet de triangle (les sommets des triangles sont étiquetés par les arêtes du voxel). La configuration 0 n'est pas utilisée car aucun voxel de la surface discrète n'est défini avec les 8 sommets classés *IN\_VERTEX* ou les 8 sommets classés *OUT\_VERTEX* : cette configuration représente des voxels n'appartenant pas à la surface. Une seconde table, indexée sur les 256 configurations de sommets par voxel permet de récupérer l'une des 14 configurations de voxel à partir de la classification des sommets. Cette table est directement utilisée pour construire la liste de triangles par voxel.

L'algorithme 2.7 présente les détails de la génération de triangles.

```

// Parcours en profondeur de l'octree
Pour chaque voxel marqué ON_VOXEL faire
    // Evaluation incrémentale
    Calculer les coordonnées des sommets du voxel ;
    // Utilisation d'une table indexée sur les 256 configurations de sommets
    Récupérer le nombre de triangles à générer ;
    // Génération du maillage traversant le voxel
    Pour chaque triangle à générer faire
        Pour chaque sommet du triangle faire
            // Utilisation d'une table indexée sur les 256 configurations de sommets
            Calculer les coordonnées du sommet ;
        Fin Pour
    Fin Pour
Fin Pour
    
```

Algorithme 2.7 – Génération du maillage, algorithme basé sur les *marching cubes*.

### B.5.3 Analyse des résultats

L'algorithme des *marching cubes*, appliqué à une surface discrète 6-connexe et utilisant le milieu des arêtes pour générer les triangles dans chaque voxel, assure la construction d'un maillage fermé (maillage sans bord) mis à part les quelques discontinuités évoquées ci-après puis résolues par la suite.

#### Discontinuité du maillage

Néanmoins, des problèmes de jointure entre les triangles peuvent apparaître sur certaines faces des voxels. En effet une face de voxel est considérée ambiguë lorsque ses sommets sont classés deux par deux selon les diagonales : la surface générée diffère suivant l'utilisation des sommets classés *IN\_VERTEX* ou *OUT\_VERTEX* (voir figure 2.28). La surface reconstruite par

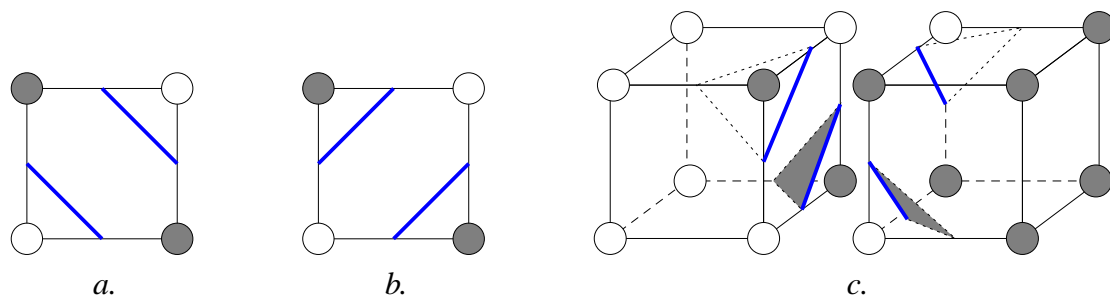


Figure 2.28 – a. Utilisation des sommets classés *OUT\_VERTEX* pour générer le maillage ; b. utilisation des sommets classés *IN\_VERTEX*, le maillage généré est différent du a ; c. discontinuité du maillage entre deux voxels 6-adjacents : sur le voxel de gauche, le maillage est généré en utilisant la configuration 3 sur les sommets classés *IN\_VERTEX* alors que sur le voxel de droite, le maillage est généré en utilisant la configuration 3 sur les sommets classés *OUT\_VERTEX*.



l'algorithme des *marching cubes* sépare toujours les deux sommets utilisés dans l'évaluation de chaque face ambiguë des configurations 1, 6, 7, 10, 12 et 13 proposées par W. Lorensen et H. Cline (voir figure 2.27).

### Topologie de la surface réelle

La séparation systématique des deux sommets de chaque face ambiguë ne permet pas au maillage reconstruit de préserver la topologie de la surface réelle : des trous apparaissent parfois dans le maillage (voir figure 2.29.a). Dans la littérature, diverses méthodes sont proposées pour

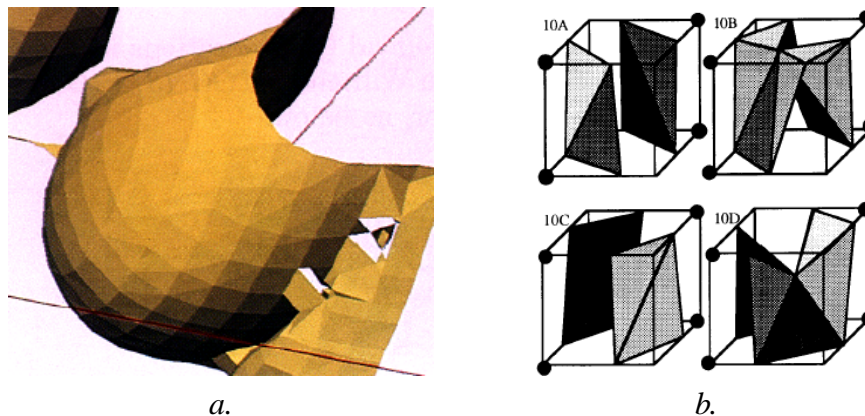


Figure 2.29 – a. Des trous apparaissent dans le maillage reconstruit avec l'algorithme original des *marching cubes* (Image extraite de [GW94]); b. la configuration 10 est subdivisée en 4 configurations pour représenter tous les cas (Image extraite de [NH91]); la topologie de la surface réelle, locale au voxel, est analysée pendant la reconstruction afin de choisir le cas le plus adéquat.

résoudre ces conflits : elles utilisent les valeurs de densité extraites des coupes 2D afin de respecter la topologie des volumes à modéliser. J. Wilhelms et A. Gelder résolvent cette ambiguïté par estimation de gradients obtenus en paramétrisant une fonction polynômiale de faible degré [WG90, GW94]. G. Nielson et B. Hamann utilisent une approche plus simple basée sur le raffinement du nombre de configurations proposées dans l'algorithme original des *marching cubes* [NH91].

L'approche de G. Nielson et B. Hamann est intéressante car elle permet de conserver l'algorithme des *marching cubes* pour reconstruire le maillage. La liste exhaustive des configurations à ajouter aux configurations déjà existantes dans [LC87] montre que seules les configurations de voxels ayant des faces ambiguës posent des problèmes lors de la reconstruction du maillage. Chaque face ambiguë double le nombre de configuration. Ainsi,

- la configuration 3 de la figure 2.27 est représentée par 2 configurations ;
- la configuration 6 est également représentée par 2 configurations ;
- la configuration 10 est représentée par 4 configurations (2 faces ambiguës, voir figure 2.29.b) ;
- la configuration 12 est également représentée par 4 configurations (2 faces ambiguës) ;
- la configuration 7 est représentée par 4 configurations malgré les 3 faces ambiguës du voxel (réduction du nombre de configurations par rotation et symétrie) ;
- la configuration 13 est représentée par 10 configurations (malgré les 6 faces ambiguës du voxel).

Le nombre total de configurations passe ainsi de 15 à 35. Cette nouvelle représentation comporte tout de même quelques inconvénients :

- la définition de plusieurs configurations pour un même marquage des sommets de certains voxels (jusqu'à 10 configurations différentes pour un voxel de la configuration 13) ;
- l'insertion d'un point supplémentaire à l'intérieur du voxel pour générer le maillage dans certaines configurations (exemple : configurations 10B et 10D de la figure 2.29.b)

La classification des sommets n'est donc plus suffisante pour pouvoir reconstruire la surface, la méthode nécessite de récupérer des informations supplémentaires sur la topologie de la surface et de stocker cette information dans les voxels pour éviter de la recalculer. De même, un nouveau point doit pouvoir être stocké dans chaque voxel. Notons que ces nouveaux ajouts d'information dans la structure de représentation des voxels sont très peu utilisés car les 6 configurations possédant des faces ambiguës représentent au total moins de 3% des cas (statistiques détaillées dans [GW94]).

### Topologie de la surface discrète

Ne disposant pas de valeurs de densité, la démarche proposée par G. Nielson et B. Hamann dans [NH91] n'est pas applicable telle qu'elle est décrite. Nous utilisons plutôt les informations topologiques fournies par la surface discrète amincie (définie à la section B.4) pour construire un maillage fermé de la surface qui respecte les propriétés topologiques de la surface discrète (nombre de composantes connexes, nombre de trous). Ainsi, les propriétés topologiques d'un morceau de surface maillée peuvent être évaluées sans avoir à manipuler la liste des triangles (les calculs peuvent se faire sur la surface discrète).

Pour conserver les propriétés topologiques de la surface, il suffit de toujours relier les deux sommets classés *IN\_VERTEX* et de séparer les deux sommets classés *OUT\_VERTEX* de chaque face ambiguë (voir figure 2.30 pour les détails).

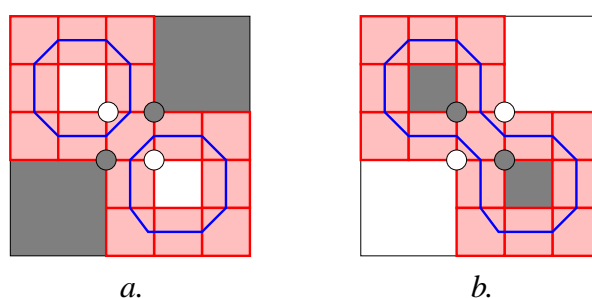


Figure 2.30 – Propriétés topologiques identiques entre la surface discrète et la surface maillée ; a. la surface discrète et le maillage comportent deux trous. ; b. la surface discrète et le maillage ne forment qu'une seule composante connexe.

### Modification de l'algorithme

Dans cette section, nous expliquons comment nous avons modifié l'algorithme original des *marching cubes* [LC87] afin de produire un maillage fermé. Nous nous sommes inspirés des travaux de G. Nielson et B. Hamann en limitant le nombre de configurations à 17 au lieu de 35

et en définissant une règle de construction toujours applicable à partir de la configuration des sommets. Seules les configurations contenant des faces ambiguës dans l’algorithme original sont modifiées, i.e. les configurations 3, 6, 7, 10, 12 et 13.

Ces six configurations, définies dans la figure 2.27, sont à présent utilisées uniquement pour générer les triangles à partir des sommets classés *OUT\_VERTEX* car elles séparent déjà les sommets classés *OUT\_VERTEX* sur toutes leurs faces ambiguës (voir figure 2.27).

En revanche, ces six configurations ne peuvent être utilisées pour générer les triangles à partir des sommets classés *IN\_VERTEX*. Nous utilisons trois nouvelles configurations définies dans la figure 2.31 afin de générer les triangles uniquement avec les sommets classés *IN\_VERTEX*. Ces nouvelles configurations sont duales (par une inversion des sommets classés *OUT\_VERTEX* et *IN\_VERTEX*) des configurations 3, 6 et 7 (ces anciennes configurations sont toujours utilisées mais uniquement avec les sommets classés *OUT\_VERTEX*).

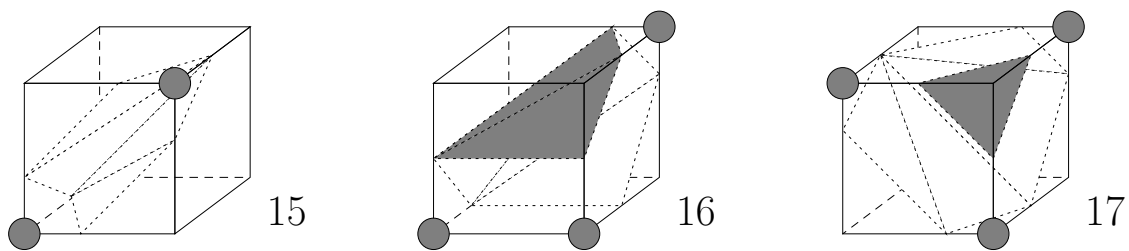


Figure 2.31 – Trois nouvelles configurations utilisées uniquement avec les sommets classés *IN\_VERTEX*. La configuration 15 est duale de la configuration 3, la 16<sup>e</sup> est duale de la 6<sup>e</sup> et la 17<sup>e</sup> est duale de la 7<sup>e</sup>.

Les configurations 10, 12 et 13 contiennent chacune 4 sommets *OUT\_VERTEX* et 4 sommets *IN\_VERTEX*. La configuration duale de chacune de ces configurations est identique à la configuration d’origine par rotation et/ou symétrie. Par conséquent, l’utilisation des sommets classés *IN\_VERTEX* pour générer les triangles d’une de ces configurations est inutile, les sommets *OUT\_VERTEX* sont déjà appropriés (voir figure 2.32 pour plus de détails).

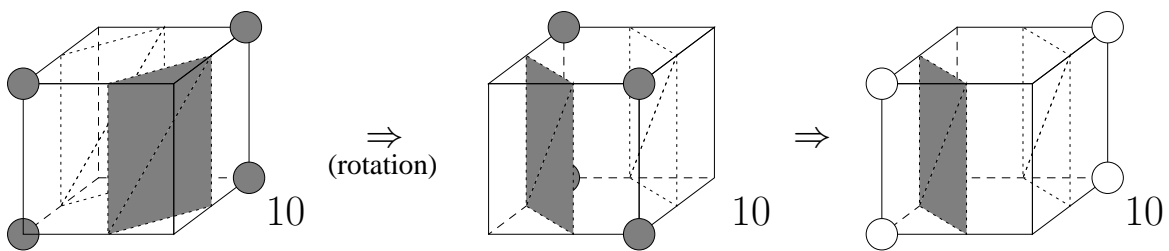


Figure 2.32 – La génération des triangles de la configuration 10 selon les sommets *IN\_VERTEX* s’effectue en réalité en utilisant les sommets *OUT\_VERTEX* de cette même configuration. La configuration de droite est identique à la configuration 10 de la figure 2.27.

Les tables accélératrices utilisées par l’algorithme 2.7 sont modifiées en fonction des nouvelles règles citées ci-dessus. La première table passe de 14 à 17 configurations de voxels et la seconde table est modifiée afin d’utiliser prioritairement les sommets classés *OUT\_VERTEX* pour les configurations 3, 6, 7, 10, 12, 13 et d’utiliser les nouvelles configurations 15, 16 et 17 sur les sommets classés *IN\_VERTEX*.

L'algorithme 2.7 ainsi modifié produit à présent un maillage fermé, sans discontinuité, conservant les mêmes propriétés topologiques que la surface discrète sans avoir besoin d'insérer de données supplémentaires dans la structure de représentation d'un voxel. L'algorithme peut ainsi générer jusqu'à 5 triangles par voxel.

### B.6 Utilisation des pixels pour affiner le maillage

En l'absence de coupes 2D de l'objet, la pondération des sommets des triangles ne peut être évaluée à partir des valeurs de densité de l'objet. Les seules informations disponibles sont les photographies de l'objet : nous utilisons de nouveau la silhouette de l'objet présente sur chaque image pour raffiner la construction du maillage polygonal.

#### B.6.1 Principe

Les pixels correspondant à l'arrière-plan sont projetés sur les voxels afin de dessiner la silhouette de l'objet sur l'ensemble des voxels de la surface. La projection d'un pixel est effectuée

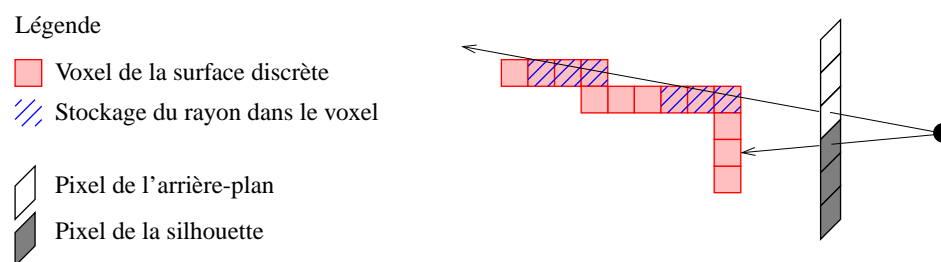


Figure 2.33 – Projection des pixels de l'arrière-plan sur les voxels de la surface discrète.

par un tracé de rayon à travers la grille régulière 3D englobant l'octree (projection identique à celle présentée à la section B.2.2) puis un stockage du rayon dans chaque voxel traversé (voir figure 2.33). Aucun de ces rayons ne traverse la surface réelle de l'objet, mais les rayons proches de la silhouette traversent fréquemment les voxels de la surface discrète. Les pixels de l'arrière-plan sont choisis pour la projection (et non les pixels de l'objet situés sur la silhouette) car chaque pixel peut être projeté sur plusieurs voxels de la surface et ainsi fournir davantage d'informations pour la surface à reconstruire.

La liste des rayons stockés dans chaque voxel est ensuite utilisée pour calculer les points d'intersection entre les rayons et les faces de voxel (voir figure 2.34). Ces points sont utilisés pour évaluer avec précision la surface de l'objet traversant chaque voxel. Des valeurs de pondération sont estimées localement sur chaque arête de voxel contenant un sommet de triangle à générer puis ajustées en fonction du voisinage de voxels afin de générer un maillage continu, proche de la surface réelle.

#### Droites séparatrices

Le maillage est défini par une surface construite à l'intérieur des voxels marqués *ON\_VOXEL* en séparant d'un côté les sommets classés *IN\_VERTEX* et de l'autre côté les sommets classés

Légende :

- rayon de la silhouette
- rayon de l'arrière-plan
- intersection avec une face
- surface réelle de l'objet

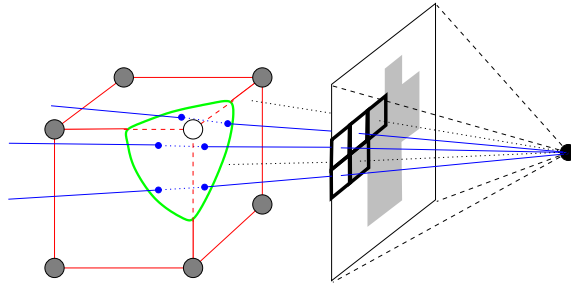


Figure 2.34 – Projection des pixels de l'arrière-plan sur un voxel de la surface discrète et calcul des points d'intersection sur chaque face traversée par la surface réelle.

*OUT\_VERTEX*. Cette surface est représentée sur chaque face de voxel par une droite séparant les deux types de sommets. L'ensemble des droites construites sur un voxel représente la surface reconstruite. Un maillage plus représentatif de la surface réelle de l'objet est ensuite obtenu par ajout de contraintes lors de la construction des droites séparatrices. En effet, les points d'intersection entre les rayons d'arrière-plan (stockés dans la structure *FEUILLE* du voxel) et chaque face sont utilisés pour affiner la construction de la droite (figure 2.35).

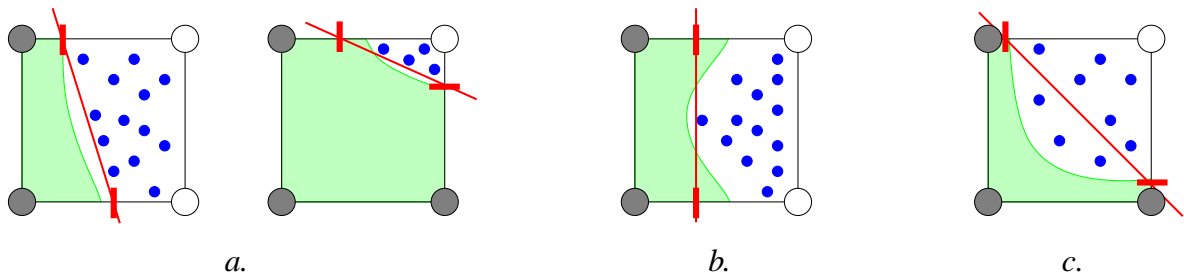


Figure 2.35 – La droite sépare les sommets classés *IN\_VERTEX* des sommets classés *OUT\_VERTEX* et des points d'intersection issus de la projection des pixels d'arrière-plan ; a. la droite séparatrice est définie au plus près des points d'intersection en passant par deux de ces points ; b. et c. la droite ne peut passer par deux points d'intersection tout en respectant les contraintes de séparation des points ; elle est définie soit en passant par un seul point d'intersection (b.), soit comme la plus proche des sommets classés *IN\_VERTEX* (c.) afin de limiter le nombre de points d'intersection situés du mauvais côté de la droite (c. est le pire des cas, il apparaît lorsque l'octree est peu profond i.e. lorsque la surface discrète est très approximative).

La droite sépare les points situés à l'intérieur de l'objet (les sommets classés *IN\_VERTEX*) des points situés à l'extérieur de l'objet (les sommets classés *OUT\_VERTEX* ainsi que l'ensemble des points d'intersection avec la face). De plus, la droite est définie aussi près que possible de l'ensemble des points situés à l'extérieur de l'objet afin de limiter la réduction du volume global de l'objet. Dans le cas général, la droite est confondue avec un segment de l'enveloppe convexe des points situés à l'extérieur de l'objet. Ceci revient à trouver le segment  $S_i$  qui minimise la fonction *dist* de calcul de distance suivante :

$$dist(S_i) = \sum_j distance(S_i, P_j)^2$$

avec  $P$  l'ensemble des points situés à l'extérieur de l'objet et  $S$  l'ensemble des segments formant l'enveloppe convexe de  $P$ .

Dans certains cas, la droite séparatrice ne peut pas être calculée en utilisant un segment de l'enveloppe convexe. Aucun segment ne permet de séparer correctement les points situés à l'intérieur de l'objet des points situés à l'extérieur. Par conséquent, la droite séparatrice est construite parallèlement au segment initial relié au milieu des arêtes concernées en respectant si possible la contrainte de séparation des points. Lorsque la séparation est possible, la droite obtenue passe par un seul point situé à l'extérieur de l'objet (voir figure 2.35.b). Sinon, la droite est construite aussi près que possible de l'ensemble des points situés à l'intérieur de l'objet afin de limiter le nombre de points d'intersection situés du mauvais côté de la droite (voir figure 2.35.c). Les points d'intersection sont des valeurs fiables extraites des images, ils doivent donc être utilisés prioritairement. Les quelques points d'intersection posant un problème sont tout de même écartés afin de ne pas remettre en cause la classification des sommets définie à la section B.4.2. Pour supprimer ces quelques imprécisions géométriques, il suffit souvent de raffiner la surface discrète à cet endroit (subdiviser le voxel à traiter).

Les faces ambiguës, dont la configuration des sommets est en opposition, demandent un traitement supplémentaire car deux droites doivent être définies pour chaque face (voir figure 2.36). La question de la construction d'une surface dans les voxels contenant des faces ambiguës a déjà été traitée à la section B.5.3 : les sommets classés *OUT\_VERTEX* sont systématiquement séparés l'un de l'autre. Cette ambiguïté sur la construction de la surface aurait pu être levée en fonction de la présence ou non de points d'intersection au centre de la face mais ceci aurait rendu difficile le maintien de la cohérence topologique sur l'ensemble de la surface de l'objet (la surface pourrait perdre sa continuité entre deux voxels 6-adjacents). Pour construire les deux

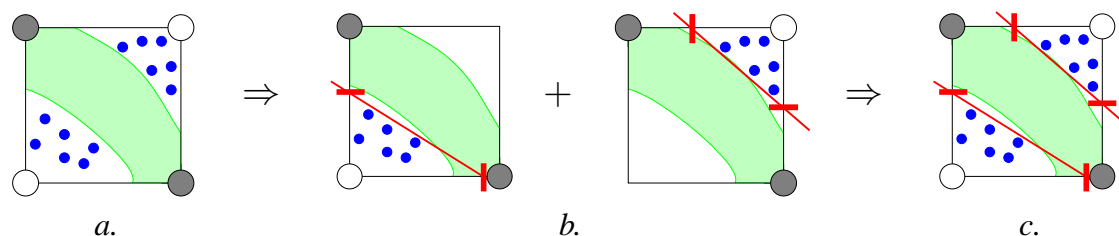


Figure 2.36 – Construction de deux droites séparatrices pour les faces ambiguës. La figure b. montre la construction de chacune des droites : la liste des points situés à l'intérieur de l'objet est commune, seule la liste des points situés à l'extérieur de l'objet diffère.

droites, les deux sommets classés *OUT\_VERTEX* sont séparés en deux listes différentes de points situés à l'extérieur de l'objet et chaque point d'intersection est inséré dans l'une des deux listes en fonction de sa distance avec les sommets. Ensuite, chaque liste est utilisée avec la liste des sommets situés à l'intérieur de l'objet afin de construire une droite respectant les contraintes définies précédemment (voir figure 2.36.b pour les détails).

Afin de supprimer l'apparition de triangles dégénérés lors de l'utilisation de l'algorithme basé sur les *marching cubes*, les droites construites ne passent ni par des points d'intersection, ni par des sommets de la face, elles sont légèrement déplacées (le déplacement est proportionnel à la taille d'un voxel de la surface, fixé empiriquement à  $\frac{1}{20}$ ).



### Continuité intra-voxel

Les intersections entre les droites séparatrices et les arêtes des voxels représentent les sommets des triangles à générer. Dans un voxel, chaque arête reliant un sommet *IN\_VERTEX* à un sommet *OUT\_VERTEX* appartient à deux faces du voxel pour lesquelles une droite séparatrice des sommets est construite. Par conséquent, deux sommets de triangle sont estimés sur cette arête (un pour chaque droite). Les résultats des calculs d'intersection entre les droites séparatrices et les arêtes doivent être harmonisés afin de n'obtenir qu'un seul sommet de triangle par arête : l'intersection la plus proche du sommet *IN\_VERTEX* relié à l'arête est désignée comme le sommet du triangle à construire (figure 2.37) et est stockée dans l'attribut *Ponderation* correspondant à l'arête du voxel (algorithme 2.9). Ce sens de déplacement des sommets de triangle est choisi afin

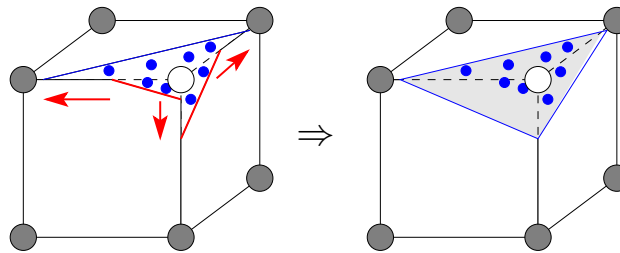


Figure 2.37 – Pour assurer la continuité de la surface créée à l'intérieur d'un voxel, un seul sommet de triangle doit être défini par arête. Lorsque plusieurs intersections existent entre les droites séparatrices et une arête, alors l'intersection la plus proche du sommet *IN\_VERTEX* relié à l'arête est utilisée.

de laisser les points d'intersection entre les rayons d'arrière-plan et les faces de voxel à l'extérieur de la surface. L'algorithme basé sur les *marching cubes* peut ainsi être utilisé pour générer un maillage de triangles par voxel en utilisant les valeurs de pondération stockées dans les voxels.

### Continuité inter-voxels

Pour obtenir un maillage fermé de la surface, les pondérations fixées sur les arêtes doivent être identiques d'un voxel à l'autre. Comme l'évaluation des pondérations est effectuée localement sur chaque voxel, il n'y a pas nécessairement de continuité de surface entre deux voxels adjacents (figure 2.38). Chaque arête utilisée dans la construction de la surface est partagée par quatre

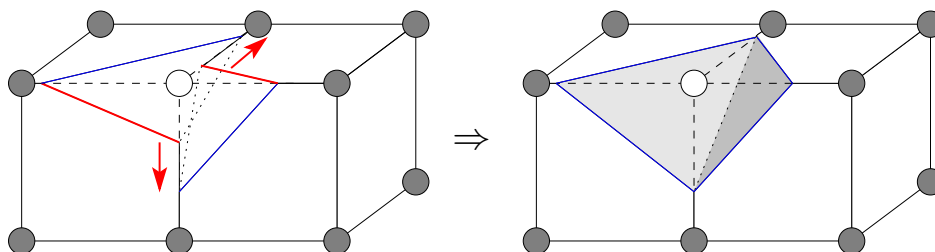


Figure 2.38 – Pour assurer la continuité de la surface créée entre plusieurs voxels adjacents, un seul sommet de triangle ne doit être défini par arête mixte commune aux voxels adjacents. L'intersection la plus proche du sommet *IN\_VERTEX* relié à l'arête est conservée, comme dans la figure 2.37.

voxels de la surface. Par conséquent, quatre valeurs de pondération peuvent exister pour chaque



arête et une harmonisation identique à celle présentée précédemment est utilisée pour ne garder qu'une seule valeur par arête (la figure 2.38 montre un exemple avec seulement deux voxels mais la surface est également définie par des voxels situés au dessus de ceux présentés sur la figure et par des voxels situés au premier plan). Ainsi, un seul sommet de triangle est défini pour chaque arête de la surface discrète et l'application de l'algorithme de génération de triangles construit un maillage continu de la surface avec une topologie identique à celle de la surface discrète.

### B.6.2 Mise en œuvre

#### Structure de données

La structure *FEUILLE* des voxels de la surface est modifiée afin de pouvoir stocker jusqu'à 255 rayons par voxel (voir algorithme 2.8).

```
Struct FEUILLE {  
    ...  
    NombreDeRayons : unsigned char ; // limité à 255 rayons par voxel  
    Rayons : unsigned short [255][3] ; // stocke la liste des rayons sous la forme  
    ...                               // (numéro de l'image, coordonnées 2D du pixel associé)  
}
```

Algorithme 2.8 – Structure de stockage de rayons dans un voxel de la surface, marqué *ON\_VOXEL*.

Ce nombre de rayons est suffisant pour des voxels de petite taille (jusqu'à quatre pixels de chaque image peuvent être projetés sur un même voxel). Pour une utilisation avec des voxels de grande taille, il suffit d'implanter un algorithme de stockage de rayons ne gardant que les rayons les plus pertinents (rayons les plus proches de la surface à reconstruire). Chaque rayon est représenté, non pas par un point plus un vecteur directeur (6 coordonnées en *double*, soit 48 octets), mais par le numéro de l'image et les coordonnées 2D du pixel correspondant au rayon, soit seulement 6 octets. Les propriétés du rayon sont recalculées à partir de l'image et des coordonnées du pixel en utilisant les matrices de changement de repère de l'image.

Cette structure est réutilisée dans le chapitre III lors de la détection des sources lumineuses afin de stocker les pixels de l'objet (et non les pixels de l'arrière-plan). L'ensemble des rayons stockés dans un voxel représente alors un ensemble de luminances émises par la surface locale au voxel.

#### Algorithme d'évaluation des pondérations

L'algorithme 2.9 présente l'évaluation des pondérations sur les arêtes des voxels. Ces pondérations sont ensuite utilisées par l'algorithme 2.7 pour générer, dans chaque voxel, une surface composée de triangles. Chaque sommet de triangle est représenté dans l'arête correspondante par une valeur de pondération entre les deux sommets de l'arête (valeur comprise entre 0 et 1). Ainsi, le stockage requiert peu d'espace mémoire et les coordonnées d'un sommet de triangle sont calculables rapidement : à partir des coordonnées du premier sommet de l'arête, ajout de la valeur de *ponderation* multipliée par la taille du voxel à la coordonnée devant être modifiée (seulement un produit et une somme pour calculer les trois coordonnées).

```

// Parcours en profondeur de l'octree
Pour chaque voxel marqué ON_VOXEL faire
    // Evaluation incrémentale
    Calculer les coordonnées des sommets du voxel ;
    // Parcours des faces traversées par le maillage à reconstruire
    Pour chaque face dont les sommets sont classés IN_VERTEX et OUT_VERTEX faire
        // Utilisation des coordonnées du voxel et de la structure FEUILLE
        Calculer la liste Lo des points d'intersection entre les rayons de l'arrière-plan et la face ;
        Ajouter dans cette liste les sommets de la face classés OUT_VERTEX ;
        Calculer la liste Li constituée des sommets de la face classés IN_VERTEX ;
        Calculer la droite séparatrice des listes Lo et Li ;
        Calculer les points d'intersection entre la droite séparatrice et les arêtes de la face ;
        Evaluer les pondérations appliquées sur les arêtes ;
        // Si la face est ambiguë, la liste Lo est dédoublée et deux droites séparatrices sont calculées
    Fin Pour
    // Parcours des arêtes mixtes (traversées par le maillage à reconstruire)
    Pour chaque arête A dont un sommet est classé IN_VERTEX et l'autre OUT_VERTEX faire
        // Harmonisation des pondérations évaluées pour les deux faces contenant cette arête
        Mettre à jour Ponderation[A] afin de construire un maillage continu à l'intérieur du voxel ;
    Fin Pour
Fin Pour
// Nouveau parcours en profondeur de l'octree
Pour chaque voxel marqué ON_VOXEL faire
    // Parcours des arêtes mixtes
    Pour chaque arête A dont un sommet est classé IN_VERTEX et l'autre OUT_VERTEX faire
        Récupérer les trois autres voxels contenant l'arête A ;
        // Harmonisation des pondérations évaluées pour les quatres voxels contenant cette arête
        Mettre à jour Ponderation[A] afin de construire un maillage continu entre les voxels adjacents ;
    Fin Pour
Fin Pour

```

Algorithme 2.9 – Evaluation des pondérations sur les arêtes.

### B.6.3 Analyse des résultats

#### Critères d'évaluation

Lorsque la géométrie de l'objet est connue (exemple d'une sphère, d'un cube), il est possible d'estimer la précision et la robustesse de notre méthode de reconstruction en estimant une erreur sur le maillage construit. Cette erreur correspond à la distance moyenne  $\bar{d}$  séparant chaque sommet de triangle généré par rapport à la surface réelle de l'objet (figure 2.39). L'écart type  $\sigma(d)$  sur

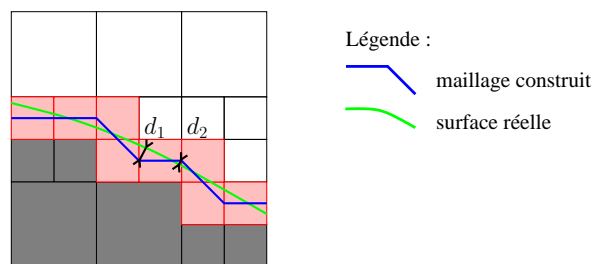


Figure 2.39 – Estimation de l'erreur sur la surface reconstruite.

cette distance  $d$  est également évalué pour vérifier la dispersion ou non des sommets construits par rapport à la surface réelle (voir équation 2.20).

$$\bar{d} = \frac{1}{\#Sommetts} \sum_{i=1}^{\#Sommetts} d_i \quad (2.19)$$

$$\sigma(d) = \sqrt{\frac{1}{\#Sommetts} \sum_{i=1}^{\#Sommetts} (d_i - \bar{d})^2} \quad (2.20)$$

### Comparaison des différents algorithmes

Le critère d'évaluation est utilisé afin de vérifier la qualité du maillage produit par rapport au maillage passant par le centre des arêtes. Nous avons évalué l'erreur uniquement sur des objets synthétiques de forme simple afin de pouvoir utiliser la géométrie réelle de l'objet.

Le tableau 2.1 montre que l'utilisation des rayons d'arrière-plan pour raffiner le maillage améliore considérablement la précision de la surface reconstruite dans le cas de la sphère. La distance séparant la surface reconstruite de la surface réelle est d'environ  $2mm$  lorsque les rayons d'arrière-plan sont utilisés contre  $6mm$  pour la méthode traditionnelle utilisant le centre des arêtes des voxels. Lorsque le nombre de rayons d'arrière-plan par voxel devient trop faible, la nouvelle méthode proposée perd de son intérêt jusqu'à fournir des résultats similaires à la première méthode. En effet, la méthode basée sur l'utilisation des rayons d'arrière-plan montre qu'elle est particulièrement efficace sur des objets dont la surface discrète est faiblement détaillée : les valeurs de distance moyenne et d'écart type sont constantes pour la sphère, quelle que soit la profondeur de l'octree, et ceci malgré un faible pourcentage de rayons d'arrière-plan stockés dans les voxels. Ces résultats montrent qu'il n'est pas nécessaire d'implanter un algorithme de stockage des rayons les plus pertinents pour chaque voxel.

Y. Yemez et F. Schmitt ont publié en 2004 une méthode similaire à celle que nous proposons, évaluant les pondérations sur les arêtes reliant un sommet marqué *OUT\_VERTEX* à un sommet marqué *IN\_VERTEX* par projection de ces arêtes sur l'ensemble des images, calcul d'intersection entre les arêtes et les silhouettes, et récupération de l'intersection la plus proche du sommet marqué *IN\_VERTEX* pour chaque arête [YS04]. Mais leur méthode fournit de moins bons résultats dans le cas de surface discrète faiblement détaillée : ils ne prennent pas en compte l'évolution de la surface réelle à l'intérieur de chaque voxel, contrairement à notre approche stockant des rayons d'arrière-plan dans chaque voxel.

Le cube, tel qu'il est présenté dans le tableau 2.1, est le pire exemple d'objet convexe à reconstruire en utilisant la silhouette de l'objet car il comporte de larges faces planes et aucune

Sphère d'un mètre de diamètre					
Profondeur de l'octree		5	6	7	8
<i>centre des arêtes utilisé</i>	$\bar{d}$	7.2 mm	5.8 mm	5.4 mm	5.5 mm
	$\sigma(d)$	5.1 mm	3.3 mm	1.9 mm	1.2 mm
<i>rayons d'arrière-plan utilisés</i>	$\bar{d}$	2.1 mm	2.0 mm	2.4 mm	3.7 mm
	$\sigma(d)$	1.5 mm	1.4 mm	1.5 mm	1.3 mm
<i># rayons d'arrière-plan / voxel</i>		202	157	72	3
<i>pourcentage de rayons stockés</i>		3%	20%	90%	100%

Cube d'un mètre de largeur					
Profondeur de l'octree		5	6	7	8
<i>centre des arêtes utilisé</i>	$\bar{d}$	26.2 mm	19.3 mm	16.6 mm	15.6 mm
	$\sigma(d)$	8.2 mm	11.8 mm	12.4 mm	12.7 mm
<i>rayons d'arrière-plan utilisés</i>	$\bar{d}$	21.1 mm	19.4 mm	19.8 mm	17.6 mm
	$\sigma(d)$	14.8 mm	14.6 mm	14.6 mm	14.0 mm
<i># rayons d'arrière-plan / voxel</i>		205	146	30	1
<i>pourcentage de rayons stockés</i>		6%	45%	90%	100%

Tableau 2.1 – Erreur sur la surface reconstruite : distance moyenne  $\bar{d}$  et écart type  $\sigma(d)$  entre la surface réelle de l'objet synthétique et le maillage construit par les deux méthodes de reconstruction de maillage. Chaque objet est représenté par 384 images de résolution  $256 \times 256$  (les images sont stockées dans un lumigraphe complet de  $8 \times 8$  vues par slab).

image (parmi les 384) n'a son centre optique positionné sur l'un des plans de la surface à reconstruire. Par conséquent, chaque face du cube est reconstruite avec une forme bombée. Néanmoins, les résultats de la méthode utilisant les rayons montrent de nouveau une certaine invariance selon la profondeur de l'octree : la surface est toujours reconstruite avec une imprécision d'environ 20mm. Cet exemple est le rare cas où la méthode utilisant les rayons d'arrière-plan peut produire des résultats légèrement moins intéressants que la méthode privilégiant le centre des arêtes lorsque le nombre de rayons par voxel est insuffisant. La méthode utilisant le centre des arêtes est avantagée par le fait que le cube soit défini par des faces parallèles aux axes du repère de l'octree.

### Nombre d'images utiles

Les tests de comparaison des différents algorithmes ont été effectués sur des objets représentés par des lumigraphes complets de 384 images. Néanmoins, la surface de l'objet peut être reconstruite avec moins d'images, il suffit que les silhouettes détectées fournissent les renseignements nécessaires à la reconstruction géométrique. Le tableau 2.2 répertorie les résultats de la reconstruction de la surface de la sphère avec seulement 32 vues tout autour de l'objet. Ces résultats révèlent que 32 images sont suffisantes pour reconstruire correctement la surface : les valeurs de distance moyenne et d'écart type sont mêmes légèrement inférieures à celles obtenues précédemment excepté pour une faible profondeur d'octree. Cela nous a conduit à n'utiliser qu'une seule série de 72 photographies (images sur  $360^\circ$ ) pour reconstruire la forme géométrique de chaque objet réel acquis.

Sphère d'un mètre de diamètre					
Profondeur de l'octree		5	6	7	8
rayons d'arrière-plan utilisés	$\bar{d}$	2.5 mm	1.4 mm	1.6 mm	2.4 mm
	$\sigma(d)$	2.2 mm	1.3 mm	1.3 mm	1.4 mm
# rayons d'arrière-plan / voxel		155	59	7	0.4
pourcentage de rayons stockés		40%	95%	100%	100%

Tableau 2.2 – Erreur sur la surface reconstruite, calculée comme dans le tableau 2.1, pour une sphère représentée par une série de seulement 32 images de résolution  $256 \times 256$  (32 vues de la sphère sur  $360^\circ$ ).

### Lissage de la forme reconstruite

Les maillages générés avec la nouvelle méthode sont moins crénelés, le contour des objets reconstruits apparaît plus lisse (voir figure 2.40). Ce lissage de la forme de l'objet apparaît parti-

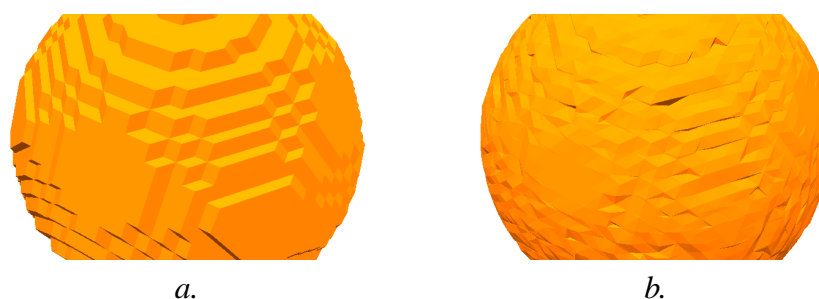


Figure 2.40 – Construction d'un maillage de la sphère à partir d'un octree de profondeur 6 ; a. en utilisant le centre des arêtes ; b. en utilisant les rayons d'arrière-plan, les contours sont plus lisses (384 images utilisées).

culièrement lors de la génération de maillages simples (comportant peu de polygones). Toutefois, la surface produite comporte beaucoup de petites irrégularités locales (rainures creusées par certains rayons d'arrière-plan) qui pourraient être ôtées par un traitement ultérieur, local à un petit voisinage de voxels afin de ne supprimer que les micro-aspérités de la surface. Ce traitement n'a pas été effectué car le contour de la surface est respecté. Les micro-aspérités de la surface vont plutôt gêner les algorithmes de calculs d'illumination car la normale localisée sur chaque triangle ne reflète pas la normale exacte de la surface réelle de l'objet. La section suivante propose une méthode d'estimation de la normale appropriée à notre modèle géométrique, en tout point de la surface.

## C Estimation de la normale

Ne disposant d'aucune information sur les conditions d'éclairage lors de l'acquisition des objets, les informations géométriques reconstruites sont les seules données exploitables pour estimer la normale à la surface des objets.

## C.1 Choix de la représentation

Pour des raisons de coût de stockage, nous choisissons de représenter la normale de la surface à l'intérieur d'un voxel par une seule valeur en ajoutant un nouvel attribut à la structure *FEUILLE* des voxels de la surface. Eventuellement, un traitement peut être envisagé afin de définir, lors d'un calcul d'illumination, une normale en tout point de la surface par interpolation entre les normales des voxels les plus proches du point d'impact. Actuellement, la définition d'une normale par voxel est suffisante pour utiliser les méthodes d'estimation de sources lumineuses (chapitre III).

Dans la littérature, l'estimation de la normale en un point de la surface est obtenue par réécriture d'une équation d'illumination en ce point, en supposant connues les propriétés de réflectance des surfaces et les conditions d'éclairage (méthodes de type *shape from shading*, voir section B.3.3 du chapitre I). Ne disposant pas de telles informations, nous ne pouvons utiliser ce type de techniques et devons nous orienter vers une évaluation à partir de la géométrie reconstruite.

Deux méthodes d'estimation des normales sont proposées dans ce mémoire. La première utilise le maillage reconstruit pour estimer une normale en chaque voxel de la surface. Elle s'appuie sur des méthodes déjà existantes d'estimation de normales à partir du maillage, mais la définition d'une surface à la fois discrète et maillée permet de stocker la normale estimée dans chaque voxel plutôt que sur chaque sommet de triangles (sinon il faut ajouter 12 valeurs par voxel, associées aux 12 pondérations sur les arêtes représentant les sommets des triangles générés par l'algorithme des *marching cubes*). La seconde méthode d'estimation des normales utilise uniquement la surface discrète et permet ainsi d'estimer les normales sans avoir à générer un quelconque maillage polygonal de la surface. Cette méthode est inspirée des travaux de G. Thürmer [TW97] mais les résultats montre qu'elle peut être améliorée par d'autres méthodes discrètes, déjà existantes, plus efficaces.

## C.2 Normales à partir du maillage

### C.2.1 Evaluation locale

Le maillage construit dans chaque voxel de la surface (marqué *ON\_VOXEL*) est représenté par une liste de 1 à 5 triangles et la normale évaluée dans un voxel correspond à la moyenne des normales des triangles pondérées par l'aire de ces triangles (voir figure 2.41.a).

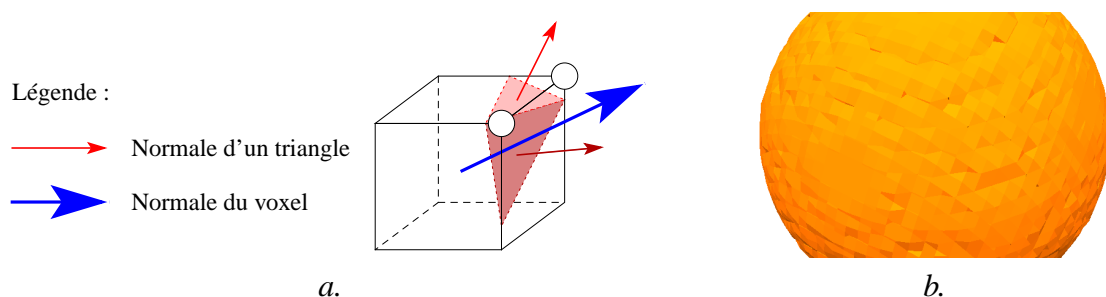


Figure 2.41 – a. Estimation de la normale d'un voxel à partir des normales des triangles composant le maillage ; b. Rendu diffus du maillage en utilisant, pour chaque triangle affiché, la normale du voxel auquel il appartient (le maillage est construit à partir d'un octree de profondeur 6).

La figure 2.41.b montre de fortes variations de la luminance entre les voxels adjacents, illustrant une grande différence des normales estimées entre les voxels voisins. Ces variations proviennent des petites aspérités apparaissant sur la surface maillée. Elles peuvent être atténuées en estimant la normale d'un voxel à partir d'une surface maillée de plus grande dimension.

## C.2.2 Evaluation sur un voisinage

L'évaluation de la normale d'un voxel sur un voisinage de voxels plutôt que localement permet la suppression, lors du rendu, des micro-aspérités géométriques de la surface (correspondant généralement à du bruit introduit lors du processus de reconstruction). Elle permet également d'utiliser avec plus de précision les méthodes de détection de sources lumineuses présentées dans le chapitre III.

La normale d'un voxel  $V$  est estimée comme précédemment par une moyenne des normales de triangles pondérées par l'aire, mais la surface maillée utilisée est étendue aux voxels voisins de  $V$  (voir figure 2.42.a). Seuls les voxels marqués  $ON\_VOXEL$ , 6-connexes au voxel  $V$  et pos-

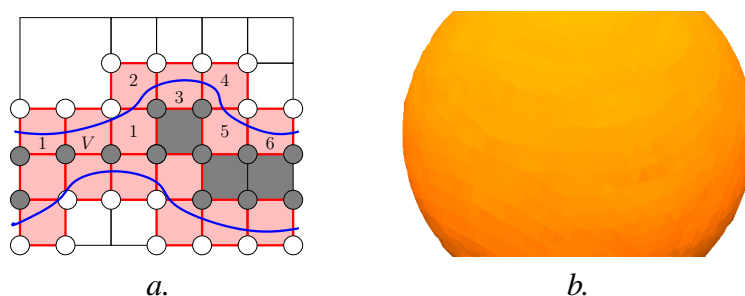


Figure 2.42 – a. Estimation de la normale du voxel  $V$  à partir des normales des triangles des voxels situés à une distance inférieure ou égale à la distance de lissage fixée par l'utilisateur ; b. Rendu diffus du maillage en utilisant, pour chaque triangle affiché, la normale du voxel auquel il appartient (le maillage est construit à partir d'un octree de profondeur 6, la normale est estimée avec une distance de lissage égale à 1).

sedant une face mixte commune au voxel  $V$  sont considérés à une distance de 1 du voxel  $V$  et utilisés pour l'estimation de la normale. Une face est définie mixte lorsqu'elle est composée d'au moins un sommet  $IN\_VERTEX$  et un sommet  $OUT\_VERTEX$ . Cette définition stricte de l'ensemble des voxels à utiliser permet par exemple, sur la figure 2.42.a, de ne pas utiliser le voxel marqué  $ON\_VOXEL$  situé juste au dessous du voxel  $V$  dans l'estimation de la normale de  $V$  et n'appartenant pas à la même portion de surface.

Une distance de lissage est mise en place afin d'utiliser une taille de maillage fixée par l'utilisateur. Un voxel est utilisé dans l'estimation de la normale du voxel  $V$  si le plus court chemin 6-connexe reliant ces deux voxels est inférieur ou égal à la distance de lissage. Ce calcul est légèrement différent de la distance de Manhattan (1-distance) car le plus court chemin doit suivre la surface discrète (voir figure 2.42.a pour l'évaluation de la distance sur quelques voxels).

Cet élargissement de la surface utilisée pour l'estimation des normales permet de lisser la surface reconstruite lors de son affichage par un algorithme de rendu (voir figure 2.42.b). Nous avons à présent une surface d'aspect lisse sur les contours (dû au raffinement du maillage reconstruit) et sur le reste de la surface (dû à l'évaluation de la normale sur un voisinage de voxels).



### C.3 Normales à partir de la surface discrète

De nombreux travaux existent déjà sur l'évaluation de la normale à partir de la surface discrète (voir section B.3.2 du chapitre I). Les plus pertinents utilisent des algorithmes de reconnaissance de droites discrètes pour estimer, en un voxel de la surface, les droites tangentes à une de ses faces et ainsi définir le vecteur normal comme étant orthogonal aux tangentes discrètes [TDR99, LV03].

L'algorithme utilisant le maillage reconstruit, présenté en section C.2, paraît plus intéressant puisqu'il s'appuie sur une représentation précise de la surface. Néanmoins, nous avons également défini une méthode d'évaluation de la normale basée sur la surface discrète afin de pouvoir s'affranchir de la génération du maillage.

La méthode proposée ne s'appuie pas sur l'évaluation d'un plan discret tangent à la surface en chaque voxel bien que cette méthode aurait pu être utilisée (elle demande une extraction de droites discrètes 3D qui n'est pas toujours très stable). Nous avons préféré développer une méthode d'estimation de normales discrètes simple, ne demandant aucune construction particulière : elle utilise les voxels situés à l'extérieur de l'objet pour estimer la normale, comme dans [TW97].

La normale d'un voxel  $V$  est donc estimée par une moyenne des directions des voxels situés à l'extérieur de l'objet et 26-voisins de  $V$  (les directions des voxels sont normées avant d'être moyennés). De plus, les voxels marqués *OUT\_VOXEL* de plus grande taille que le voxel  $V$  sont

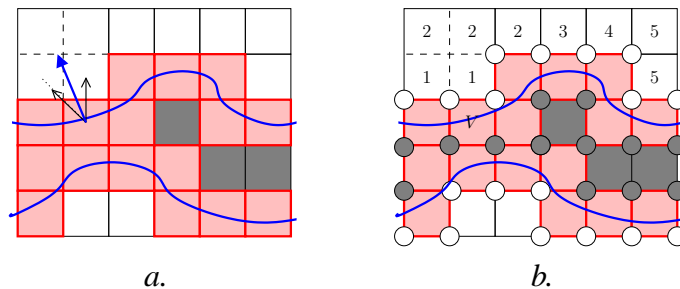


Figure 2.43 – a. Estimation de la normale du voxel  $V$  à partir des directions des voxels *OUT\_VOXEL* 26-adjacents à  $V$  (distance de lissage égale à 1); b. évaluation de la distance séparant chaque voxel marqué *OUT\_VOXEL* du voxel  $V$  : plusieurs valeurs de distance sont calculés pour les voxels de plus grande taille que  $V$ .

représentés comme une subdivision de voxels réguliers de la taille de  $V$  pour pouvoir appliquer l'algorithme d'estimation des normales (voir figure 2.43.a).

Comme dans l'estimation des normales à partir du maillage, une distance de lissage est définie pour utiliser un nombre plus important de voxels marqués *OUT\_VOXELS* dans le calcul de la normale. Un voxel est utilisé dans l'estimation de la normale du voxel  $V$  si le plus court chemin 26-connexe reliant ces deux voxels est inférieur ou égal à la distance de lissage. Ce calcul est légèrement différent de la distance de Tchebychev ( $\infty$ -distance) car le plus court chemin doit être composé de voxels *OUT\_VOXEL* excepté le voxel  $V$ . Ceci évite ainsi d'utiliser les voxels situés du mauvais côté de l'objet (exemple des voxels marqués *OUT\_VOXEL* situés au dessous du voxel  $V$  sur la figure 2.43.b).

Les normales sont ainsi définies à partir d'un nombre fixe de directions. Il est possible de définir une liste finie de valeurs pour toutes les normales estimées afin d'obtenir un stockage plus compact de la normale dans chaque voxel (2 octets représentent environ 32000 combinaisons

différentes alors qu'un stockage de trois flottants nécessitent entre 12 et 24 octets suivant la précision).

## C.4 Analyse des résultats

### C.4.1 Critères d'évaluation

Une erreur est estimée sur les normales produites afin de vérifier l'impact du nouvel algorithme de reconstruction sur l'estimation de la normale. Cette erreur correspond à l'angle moyen  $\bar{\alpha}$  formé sur chaque voxel entre la normale estimée et la normale réelle de la surface (voir figure 2.44). L'écart type  $\sigma(\alpha)$  sur cet angle  $\alpha$  est également évalué pour vérifier la dispersion ou non

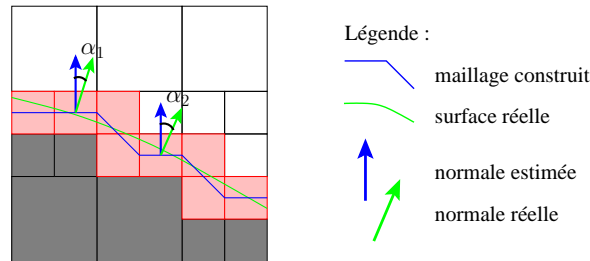


Figure 2.44 – Estimation de l'erreur sur la normale estimée en chaque voxel.

des normales estimées par rapport à la normale réelle (voir équation 2.22).

$$\bar{\alpha} = \frac{1}{\#ON\_VOXEL} \sum_{i=1}^{\#ON\_VOXEL} \alpha_i \quad (2.21)$$

$$\sigma(\alpha) = \sqrt{\frac{1}{\#ON\_VOXEL} \sum_{i=1}^{\#ON\_VOXEL} (\alpha_i - \bar{\alpha})^2} \quad (2.22)$$

### C.4.2 Comparaison des différents algorithmes

Le critère d'évaluation est appliqué sur la sphère utilisée lors de l'évaluation de la surface reconstruite afin de comparer la précision des normales estimées soit à partir de la surface discrète, soit à partir du maillage. Les normales sont estimées sur les deux types de maillage reconstruit (avec et sans l'utilisation des rayons d'arrière-plan) afin de vérifier si le nouvel algorithme de construction de maillage contribue à la précision de l'estimation des normales (voir tableau 2.3 pour les détails).

Sur une faible distance de lissage (distance 1), l'utilisation de la surface discrète ne fournit pas de normales suffisamment précises ( $11^\circ$  d'erreur en moyenne) pour pouvoir être ensuite exploitées alors que l'utilisation du maillage fournit toujours des normales avec une erreur inférieure à  $6^\circ$ . Cependant, il existe une différence notable entre les deux méthodes de construction de maillage. L'utilisation d'un maillage généré à partir du centre des arêtes fournit des normales avec une imprécision de  $5 - 6^\circ$  alors que l'utilisation d'un maillage généré avec les rayons d'arrière-plan fournit de meilleurs résultats ( $2 - 3^\circ$  d'imprécision). Nous avons vu précédemment que le nouvel algorithme de reconstruction de maillage fournit un maillage d'excellente

Sphère d'un mètre de diamètre										
Profondeur de l'octree		5	6		7			8		
Distance de lissage		1	1	2	1	2	3	1	3	5
<i>surface discrète utilisée</i>	$\bar{\alpha}$	10.1°	10.9°	3.6°	11.3°	3.5°	2.0°	12.0°	2.5°	1.8°
	$\sigma(\alpha)$	5.5°	6.7°	2.3°	6.8°	2.1°	1.1°	6.9°	1.4°	0.81°
<i>maillage utilisé (centre des arêtes)</i>	$\bar{\alpha}$	5.2°	5.3°	2.4°	5.4°	2.6°	1.6°	6.4°	2.1°	1.1°
	$\sigma(\alpha)$	3.0°	3.5°	1.6°	3.5°	1.6°	0.96°	3.7°	1.2°	0.69°
<i>maillage utilisé (rayons d'arrière-plan)</i>	$\bar{\alpha}$	2.4°	2.4°	1.3°	4.4°	2.1°	1.3°	7.0°	2.1°	1.1°
	$\sigma(\alpha)$	1.8°	1.4°	0.74°	2.7°	1.2°	0.77°	4.1°	1.2°	0.71°

Tableau 2.3 – Erreur sur la normale estimée : angle moyen  $\bar{\alpha}$  et écart type  $\sigma(\alpha)$  entre la normale réelle de l'objet synthétique et la normale estimée soit à partir de la surface discrète, soit à partir des deux méthodes de reconstruction de maillage (la sphère est représentée par 384 images de résolution  $256 \times 256$ ).

qualité pour une faible profondeur d'octree (5 ou 6 sur l'exemple de la sphère) car la précision de la reconstruction dépend essentiellement du nombre de rayons d'arrière-plan stockés par voxel de surface. L'algorithme d'estimation de la normale à partir du maillage suit le même principe : l'imprécision sur l'orientation de la normale n'est que de  $2.4^\circ$  en moyenne sur un octree de profondeur 5 ou 6, soit deux fois moins qu'avec l'utilisation d'un maillage généré à partir du centre des arêtes.

A partir d'une géométrie basée-voxel plus détaillée, un léger lissage de la normale estimée permet d'obtenir d'aussi bons résultats : les normales sont estimées avec, en moyenne,  $2.1^\circ$  d'imprécision avec une distance de lissage 2 pour un octree de profondeur 7 ou avec une distance 3 pour un octree de profondeur 8. Un lissage plus important permet de réduire l'erreur aux alentours de  $1^\circ$  mais ce résultat est faussé par le fait que la sphère est un cas idéal en terme de lissage de la normale estimée.

### C.4.3 Eventuelles améliorations

Une pondération entre les différents voxels utilisés peut être définie lors du lissage des normales en fonction de la distance de chaque voxel : en pratique, nous n'avons pas constaté de différences significatives après les tests de plusieurs valeurs de pondération permettant d'atténuer la prise en compte des normales des voxels les plus éloignés. Néanmoins, nous montrons dans la section D de ce chapitre que le lissage des normales a des effets néfastes sur les bords des surfaces lors du rééclairage (zones de la surface qui ne sont pas de continuité C1).

Avec l'utilisation de la surface discrète, la distance de lissage doit être au moins égale à 2 pour obtenir un résultat convenable (normale estimée avec une erreur inférieure à  $5^\circ$ ). Il faut utiliser une méthode similaire à [TDR99, LV03], évaluant localement un plan discret tangent à chaque voxel, pour espérer obtenir de meilleurs résultats en discret.

### C.4.4 Normales nulles

La normale de certains voxels peut être estimée comme un vecteur nul. Ce cas de figure survient pour les voxels contenant deux morceaux de surfaces : les configurations 10 (figure 2.32) et 17 (figure 2.31) sont un exemple de voxels contenant deux surfaces distinctes, représentant un amincissement local de l'objet ou une cloison entre deux trous (figure 2.30). Aucune estimation

de normale ne peut être satisfaisante pour ce type de voxels puisqu'il faudrait stocker deux normales distinctes (une par morceau de surface). Une solution est de subdiviser le voxel ambigu en 8 sous-voxels et d'appliquer les algorithmes de sculpture et de raffinement de surface discrète sur ces nouveaux voxels. Comme l'amincissement de la surface discrète interdit d'avoir une partie de l'objet ou un trou composé d'un seul voxel d'épaisseur, ces configurations apparaissent très rarement (figures 2.22, 2.23 et 2.24). Dans ce cas nous avons choisi d'estimer la normale en utilisant un seul des deux morceaux de surface définis dans le voxel et son voisinage associé (parcours des arêtes mixtes des voxels de la surface).

### C.4.5 Choix des paramètres de reconstruction

La distance de lissage et la profondeur de l'octree doivent être définies en fonction de la résolution des images représentant l'objet et de la micro-géométrie de la surface. Pour des images de résolution  $256 \times 256$ , une surface d'aspect lisse est représentée par une faible profondeur d'octree (une profondeur de 6 permet de stocker suffisamment de rayons d'arrière-plan par voxel de surface). Lorsque la surface contient plus de petits détails géométriques, une profondeur d'octree plus importante permet de récupérer les contours accidentés de l'objet. L'idéal est de représenter chaque petit détail géométrique par deux ou trois voxels afin de pouvoir effectuer un léger lissage de la normale lors de son estimation. Ainsi, si ces détails sont représentés par deux ou trois pixels sur des images de résolution  $256 \times 256$ , raffiner l'octree à une profondeur de 8 semble un bon compromis.

## D Résultats

Les résultats sur la reconstruction géométrique, utilisant les pixels des images pour raffiner le maillage, ont été publiés en partie en 2005 dans le journal de WSCG [\[MM05\]](#).

### D.1 Statistiques sur les rayons projetés

Les tableaux 2.4 et 2.5 permettent de vérifier la qualité de la reconstruction géométrique, respectivement d'objets synthétiques et d'objets réels, à partir de statistiques sur les pixels des images projetées sur la géométrie reconstruite.

La précision de la reconstruction de la surface discrète des objets (section B.3) est validée par le pourcentage des rayons d'arrière-plan touchant les voxels situés à l'intérieur de l'objet et des rayons de l'objet n'atteignant jamais la surface discrète. Toutes ces valeurs sont proches de 0% ( $< 0.2\%$ ) excepté pour le clown (5.2%). Cette différence s'explique par un calibrage moins précis des photographies représentant le clown. En effet, la version définitive du système d'acquisition (présenté à la section A.2.2) a été utilisée pour l'acquisition du tournesol mais le clown avait été construit précédemment, pendant la phase de calibrage du système. Les valeurs non parfaitement à 0% dans le cas de la reconstruction des objets synthétiques s'explique par l'étape d'amincissement de la surface supprimant des voxels de surface composés de rayons d'arrière-plan et de rayons de l'objet (section B.4).

Le pourcentage de rayons d'arrière-plan qui traversent le maillage est proche de 0 (0.02% pour les objets synthétiques et 0.03% pour les objets réels). Ceci s'explique par la mise en place

<b>Sphère, objet synthétique</b>					
Pourcentage de rayons qui touchent		OUT	ON	Mesh	IN
<i>rayons d'arrière-plan (pixels noirs)</i>	<i>traversant l'octree</i>	96.8%	3.2%	0.02%	0%
	<i>traversant les VOXEL_ON mais pas les VOXEL_IN</i>	×	99.2%	0.8%	×
<i>rayons de l'objet (pixels de la sphère)</i>	<i>traversant l'octree</i>	0%	0.1%	3.1%	96.8%
	<i>traversant les VOXEL_ON mais pas les VOXEL_IN</i>	×	4.1%	95.9%	×

<b>Vélo, objet synthétique</b>					
Pourcentage de rayons qui touchent		OUT	ON	Mesh	IN
<i>rayons d'arrière-plan (pixels noirs)</i>	<i>traversant l'octree</i>	98.2%	1.8%	0.005%	0%
	<i>traversant les VOXEL_ON mais pas les VOXEL_IN</i>	×	99.7%	0.3%	×
<i>rayons de l'objet (pixels du vélo)</i>	<i>traversant l'octree</i>	0.03%	3.8%	16.6%	79.6%
	<i>traversant les VOXEL_ON mais pas les VOXEL_IN</i>	×	18.8%	81.2%	×

OUT : les voxels externes (VOXEL\_OUT) mais pas la surface discrète (VOXEL\_ON) ;

ON : la surface discrète (VOXEL\_ON) mais pas le maillage ;

Mesh : le maillage mais pas les voxels internes (VOXEL\_IN) ;

IN : les voxels internes (VOXEL\_IN).

Tableau 2.4 – Statistiques sur l'ensemble des rayons projetés sur la structure de données construite. Les deux objets sont des objets synthétiques, représentés par un lumigraphe complet de 384 images de résolution  $256 \times 256$ . Chaque objet est reconstruit avec un octree de profondeur 7 et un maillage généré avec l'utilisation des rayons d'arrière-plan.

d'un algorithme exploitant les rayons d'arrière-plan pour construire et raffiner le maillage produit (section B.6).

En revanche le pourcentage de rayons provenant de l'objet et qui ne traversent pas le maillage est moins négligeable (entre 0.1% et 3.8%), il est la conséquence directe de l'algorithme exploitant les rayons d'arrière-plan (au détriment des rayons de l'objet). Ceci signifie, entre autre, une perte d'information sur les contours de l'objet : ceux-ci sont légèrement érodés, entraînant un petit rétrécissement de l'objet reconstruit. Cette érosion des contours est plus visible sur les objets à géométrie plus complexe (3.8% sur le vélo contre 0.1% sur la sphère).

L'exemple du tournesol (objet réel avec une géométrie de complexité comparable au vélo) montre la fiabilité du système d'acquisition : 2.1% de rayons de l'objet ne touchant pas le maillage contre 3.8% dans le cas du vélo. La similitude se remarque plus facilement sur les rayons de l'objet traversant la surface discrète sans atteindre l'intérieur de l'objet : 21.2% de ces rayons ne touchent pas le maillage du tournesol contre 18.8% dans le cas du vélo.

<b>Clown, objet réel</b>					
Pourcentage de rayons qui touchent		OUT	ON	Mesh	IN
<i>rayons d'arrière-plan (pixels noirs)</i>	<i>traversant l'octree</i>	98.6%	1.4%	0.03%	0.001%
	<i>traversant les VOXEL_ON mais pas les VOXEL_IN</i>	×	98.0%	2.0%	×
<i>rayons de l'objet (pixels du clown)</i>	<i>traversant l'octree</i>	5.2%	3.8%	6.3%	84.7%
	<i>traversant les VOXEL_ON mais pas les VOXEL_IN</i>	×	37.9%	62.1%	×

<b>Tournesol, objet réel</b>					
Pourcentage de rayons qui touchent		OUT	ON	Mesh	IN
<i>rayons d'arrière-plan (pixels noirs)</i>	<i>traversant l'octree</i>	99.1%	0.8%	0.01%	0.001%
	<i>traversant les VOXEL_ON mais pas les VOXEL_IN</i>	×	98.7%	1.3%	×
<i>rayons de l'objet (pixels du tournesol)</i>	<i>traversant l'octree</i>	0.2%	2.1%	7.9%	89.8%
	<i>traversant les VOXEL_ON mais pas les VOXEL_IN</i>	×	21.2%	78.8%	×

Tableau 2.5 – Statistiques identiques à celles présentées dans le tableau 2.4 mais sur des objets réels. Les deux objets sont représentés par une série de 72 vues. Le clown, composé de photographies de résolution  $256 \times 256$ , est reconstruit avec un octree de profondeur 7 (comme les objets synthétiques) alors que le tournesol, composé de photographies de résolution  $512 \times 512$ , est reconstruit avec un octree de profondeur 8.

## D.2 Occupation mémoire

Le tableau 2.6 résume l'occupation mémoire de l'application permettant de reconstruire la géométrie d'un objet. Les coûts mémoire dépendent essentiellement de la place prise par le

Profondeur de l'octree	<b>Velo</b>		<b>Clown</b>	
	7	8	7	8
<i>nombre d'images</i>	384		72	
<i>taille d'une image</i>	256 × 256 192 Ko		256 × 256 192 Ko	
<i>coût du cache d'images</i>	72 Mo		14 Mo	
<i>nombre de VOXEL_ON</i>	9 358	37 706	13 482	54 708
<i>nombre de triangles</i>	18 724	75 440	26 956	109 404
<i>taille de la structure FEUILLE</i>	1.55 Ko à 2.0 Ko			
<i>coût de l'octree</i>	19 Mo	75 Mo	27 Mo	108 Mo
<i>coût total de l'application</i>	92 Mo	148 Mo	42 Mo	123 Mo

Tableau 2.6 – Coût mémoire du processus complet de reconstruction géométrique. Ce coût est linéaire en fonction du nombre d'images et du nombre de voxels de surface générés.

chargement de l'ensemble des images représentant l'objet et par la construction de l'octree. Les



voxels de la surface (marqués *VOXEL\_ON*) occupent la presque totalité de la mémoire allouée à la structure de l'octree (le reste de la structure prend un peu moins de  $1Mo$ ). Chaque voxel *ON\_VOXEL* contient une structure *FEUILLE* permettant de stocker l'ensemble des informations nécessaires à la reconstruction géométrique de l'objet (environ  $1.55Ko$ ). Comme cette structure est également utilisée pour les étapes d'analyse des chapitres suivants, une feuille doit contenir des informations supplémentaires et prend ainsi environ  $2Ko$  en mémoire.

Le vélo, défini par 384 images de résolution  $256 \times 256$ , prend autant de place mémoire pour le stockage des images (environ  $75Mo$ ) que pour le stockage de la structure construite (également  $75Mo$ ), soit un total de  $150Mo$ . L'extrapolation à une précision légèrement plus importante ( $512 \times 512$  pour la résolution des images et 9 pour la profondeur de l'octree) montre que la taille totale de l'application va être multipliée par 4 à chaque niveau de précision supplémentaire (4 fois plus de pixels, 4 fois plus de voxels de surface), soit environ  $600Mo$  de données chargées en mémoire.

Par conséquent, nous avons mis en place un système de cache d'images, permettant de limiter le nombre d'images chargées simultanément. Par exemple, un cache de 100 images permet de réduire par 4 l'occupation mémoire des images. Ce système de cache fonctionne relativement bien car les algorithmes utilisent la plupart du temps les images séquentiellement. Lors des parcours de tous les voxels de l'octree, il peut y avoir des soucis d'accès aléatoire aux images. Néanmoins, dans l'exemple de l'estimation de la normale en chaque voxel avec une distance de lissage, l'utilisation d'un voisinage de voxels engendre l'extraction d'informations supplémentaires comprises dans la même plage d'images que le voxel en cours de traitement et demande donc peu de surcoût pour le chargement d'images (toutes les images utiles sont chargées dans le cache).

### D.3 Complexité des algorithmes

Les complexités des différents algorithmes, répertoriées dans le tableau 2.7, sont toujours évaluées en nombre de voxels parcourus (non forcément traités).

	Algorithme	Complexité
<i>construction de l'octree</i>	2.2, 2.3, 2.4	$O(m \times 2^p)$
<i>construction de la surface discrète 6-connexe</i>	2.4	$O(v)$
<i>amincissement de la surface discrète 6-connexe</i>	2.5	$O(v)$
<i>stockage des rayons d'arrière-plan</i>	2.8	$O(m \times 2^p)$
<i>évaluation des pondérations sur les arêtes</i>	2.9	$O(v)$
<i>génération du maillage</i>	2.7	$O(v)$
<i>estimation des normales à partir du maillage</i>		$O(v)$

$m$  : nombre de rayons traversant la grille régulière (nombre de pixels projetés) ;

$2^p$  : nombre de cellules traversées dans la grille par un rayon ;

$p$  : profondeur de l'octree ;

$v$  : nombre de voxels stockés dans l'octree.

Tableau 2.7 – Complexité des algorithmes de reconstruction géométrique. Ces algorithmes dépendent essentiellement du nombre de pixels, du nombre de voxels et de la profondeur de l'octree.



L'algorithme 2.2 construisant de manière itérative l'octree à partir d'une série d'images a une complexité en  $O(m \times 2^n)$  pour chaque niveau  $n$  de l'octree reconstruit ( $m$  étant le nombre total de pixels de la série d'images). Le  $2^n$  provient de la projection de chaque pixel sur l'octree en lançant un rayon à travers une grille accélératrice 3D. Cette grille contenant  $2^{n^3}$  cellules (chaque cellule est de la taille du plus petit voxel contenu dans l'octree), tout rayon traversant la grille intersecte entre  $2^n$  et  $3 \times 2^n$  cellules 6-connexes selon s'il traverse la grille en suivant un axe ou une diagonale de la grille. L'algorithme construisant l'octree itérativement (la profondeur de l'octree est augmentée de 1 à chaque itération de l'algorithme), sa complexité finale est  $\sum_{n=0}^p m \times 2^n = m \times (2^{p+1} - 1)$  voxels parcourus soit en  $O(m \times 2^p)$  avec  $p$  la profondeur finale de l'octree construit. L'algorithme 2.8 stockant les rayons d'arrière-plan dans les voxels de surface projetée chaque pixel des images sur l'octree en utilisant la grille régulière. Sa complexité est comparable à la dernière itération de l'algorithme de construction d'octree, elle est donc en  $O(m \times 2^p)$ .

L'algorithme 2.4 construisant la surface discrète de l'objet parcourt tous les voxels de l'octree ( $v$  voxels) et recherche les 26 voxels voisins pour traiter chacun des voxels soit un total de  $27v$  voxels parcourus : la complexité de cet algorithme est donc en  $O(v)$  avec  $v$  le nombre de voxels contenus dans l'octree. Tous les autres algorithmes (2.5, 2.9, 2.7 et l'estimation des normales) sont de complexité comparable, en  $O(v)$ , avec un parcours en profondeur de l'octree et quelquefois une recherche de voxels voisins pour chaque voxel à traiter.

## D.4 Temps de calcul

Les temps de calcul présentés dans le tableau 2.8 sont obtenus avec un ordinateur équipé d'un processeur Xéon cadencé à 2.4 GHz, de 1 Go de mémoire centrale (RAM) et d'un disque dur de 40 Go. L'algorithme 2.2 construisant l'octree itérativement à partir d'une série d'images

Profondeur de l'octree	Velo		Clown	
	7	8	7	8
<i>construction de l'octree et de la surface discrète (algorithmes 2.2,2.3,2.4)</i>	<i>43min</i>	<i>1h22</i>	<i>11min</i>	<i>20min</i>
<i>amincissement de la surface discrète 6-connexe (algorithme 2.5)</i>	<i>&lt; 1s</i>	<i>&lt; 1s</i>	<i>&lt; 1s</i>	<i>&lt; 1s</i>
<i>stockage des rayons d'arrière-plan (algorithme 2.8)</i>	<i>15min40</i>	<i>31min00</i>	<i>3min30</i>	<i>6min50</i>
<i>évaluation des pondérations sur les arêtes (algorithme 2.9)</i>	<i>5s</i>	<i>14s</i>	<i>8s</i>	<i>19s</i>
<i>génération du maillage (algorithme 2.7)</i>	<i>&lt; 1s</i>	<i>&lt; 1s</i>	<i>&lt; 1s</i>	<i>&lt; 1s</i>
<i>estimation des normales (avec lissage de distance 3)</i>	<i>&lt; 1s</i>	<i>2s</i>	<i>&lt; 1s</i>	<i>2s</i>
<i>reconstruction géométrique complète</i>	<i>≈ 59min</i>	<i>≈ 1h54</i>	<i>≈ 15min</i>	<i>≈ 28min</i>

Tableau 2.8 – Temps de calcul du processus complet de reconstruction géométrique. Ce temps dépend essentiellement du nombre d'images et de la résolution de la grille accélératrice.

est l'étape la plus longue du processus de reconstruction. La reconstruction du vélo à partir de 384 images requiert un peu plus d'une heure afin d'obtenir un octree de profondeur 8. Les temps

de reconstruction sont doublés à chaque changement de profondeur de l'octree : 43 minutes pour un octree de profondeur 7 contre 1 heure 22 pour un octree de profondeur 8 lors de la reconstruction du vélo. Ces temps vérifient la complexité de l'algorithme ( $O(m \times 2^p)$  avec  $p$  la profondeur de l'octree) grâce à la mise en place de la grille accélératrice. Sans cette optimisation, les temps dépendent du nombre de voxels entre deux niveaux consécutifs de l'octree et sont alors quadruplés (voir tableau 2.6 pour le nombre de voxels).

L'algorithme de stockage des rayons d'arrière-plan ayant la même complexité que l'algorithme de construction de l'octree, son exécution suit les mêmes variations tout en étant trois fois plus rapide. L'utilisation de 5 fois moins d'images dans le processus de reconstruction du clown permet ainsi de réduire par 5 sa reconstruction, comparée à la reconstruction du vélo.

Les autres algorithmes, définis par un parcours des voxels de l'octree (complexité en  $O(v)$ ), ont des temps d'exécution négligeables de l'ordre de quelques secondes. La structure d'octree permet ainsi de générer un maillage de plus de 100000 triangles en moins d'une seconde, de même pour l'algorithme d'estimation d'une normale en chaque voxel de la surface.

La mise en place d'un cache d'images, expliqué à la section D.2, ralentit les accès aux pixels puisque chaque accès à un pixel requiert un parcours du cache d'images : soit pour trouver l'emplacement de l'image préalablement chargée, soit pour découvrir que l'image est pas en mémoire. Néanmoins, l'utilisation d'une variable stockant l'adresse mémoire de la dernière image utilisée permet de considérablement atténuer les baisses de performances dues au cache d'images : en effet, les accès successifs de pixels demandent généralement une information provenant de la même image.

## D.5 Objets synthétiques reconstruits

La figure 2.45 montre quelques exemples d'objets virtuels reconstruits. La reconstruction du vélo avec un octree de profondeur 7 laisse apparaître quelques imperfections, notamment concernant les tubes bleus. En effet, ils sont de trop petite taille pour pouvoir être correctement éclairés à partir de l'estimation d'une normale par voxel. Le lissage conduit à une utilisation de voxels autour du tube pour définir la normale. Le lissage de la normale entraîne également des erreurs d'illumination sur le guidon, provenant de l'interpolation entre le guidon et le siège du vélo : ces erreurs peuvent être éliminées en restreignant le voisinage de voxels utilisés à des voxels provenant d'un même plan. Il faudrait utiliser une estimation de plan tangent à chaque voxel comme dans [LV03]. Ce plan tangent permettrait de définir une première approximation de la normale en chaque voxel et de restreindre le voisinage de voxels à utiliser pour moyenner les normales entre voxels comme dans [Thu01].

Une reconstruction plus précise de la géométrie du vélo, avec un octree de profondeur 8, atténue les imperfections et affiche plus distinctement les textures (par exemple la texture de bois sur le siège). La reconstruction de la carriole au niveau de profondeur 9 de l'octree montre la fiabilité de l'algorithme de reconstruction sur des images bien calibrées (lors de l'utilisation d'images synthétiques, le calibrage de caméra n'intervient pas sur les erreurs de reconstruction géométrique).

## D.6 Objets réels reconstruits

Les figures 2.46 et 2.47 présentent la reconstruction d'objets réels obtenus avec notre système d'acquisition (détails de ce système à la section A.2.2). Une reconstruction du clown avec un

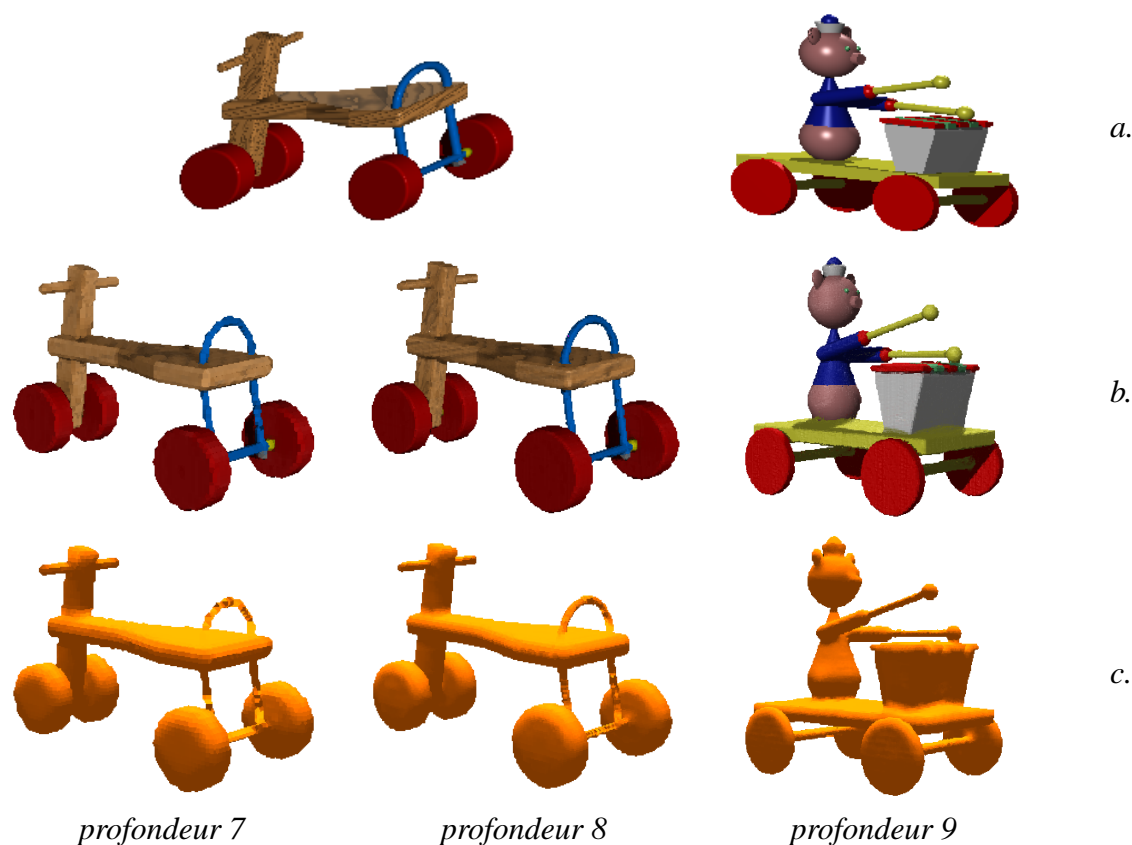


Figure 2.45 – Reconstruction d’objets virtuels à partir d’un lumigraphe complet composé de 384 images. *a.* images utilisées par l’algorithme de reconstruction de résolution  $256 \times 256$  pour le vélo et  $350 \times 350$  pour la carriole ; *b.* rendu du maillage sans rééclairage : la luminance de chaque triangle correspond à la luminance moyenne émise par le voxel auquel il appartient, évaluée à partir de la projection des images du lumigraphe sur l’octree ; *c.* rendu diffus du maillage éclairé par une source lumineuse directionnelle : la normale de chaque triangle est remplacée par la normale du voxel auquel il appartient (l’auto-ombrage n’est pas évalué).

octree de profondeur 8 n’est pas pertinente car la forme de l’objet est suffisamment simple pour être correctement reconstruite dès le niveau 7. Le dessous du noeud papillon est légèrement plus précis au niveau 8 mais de nouveaux artefacts apparaissent entre les cheveux et le visage du clown (mélange des deux textures). Ceci pourrait être corrigé par une technique de photo-cohérence (voir section B.2.2 du chapitre I) en testant la consistance de la teinte de chaque voxel. Si un voxel de la surface n’est pas vu sur toutes les images avec une même teinte, alors ce voxel doit être oté de la surface, la surface discrète doit être étendue vers les voxels de l’intérieur de l’objet et ces nouveaux voxels doivent également être traités. De nouvelles techniques basées sur l’auto-occlusion (*shadow carving*) permettent également de raffiner la géométrie obtenue par l’enveloppe visuelle [SAR<sup>+</sup>06]. Les voxels situés dans des zones d’ombres (après projection sur les images d’origine) sont illuminés par les sources lumineuses (de coordonnées connues) afin de vérifier qu’ils sont réellement non éclairés. Si ce n’est pas le cas, ils sont supprimés de la géométrie de la surface de l’objet. Cette technique est robuste car elle s’appuie sur une détection fiable des régions d’ombre : seules les zones détectées ombrées avec une très forte probabilité (seuil adéquat) sont soumises à cette évaluation. Cette technique nécessite néanmoins



Figure 2.46 – Une série de 72 images de résolution  $256 \times 256$  est utilisée pour la reconstruction de cet objet réel (l'acquisition de ces images a été effectuée avec une méthode d'étalonnage moins précise que celle présentée dans ce manuscrit) ; a. photographies d'un clown en plastique d'aspect spéculaire ; b. rendu du maillage sans rééclairage : l'utilisation de la luminance moyenne émise par chaque voxel réduit les effets spéculaires visibles sur les photographies ; c. rendu diffus du maillage.

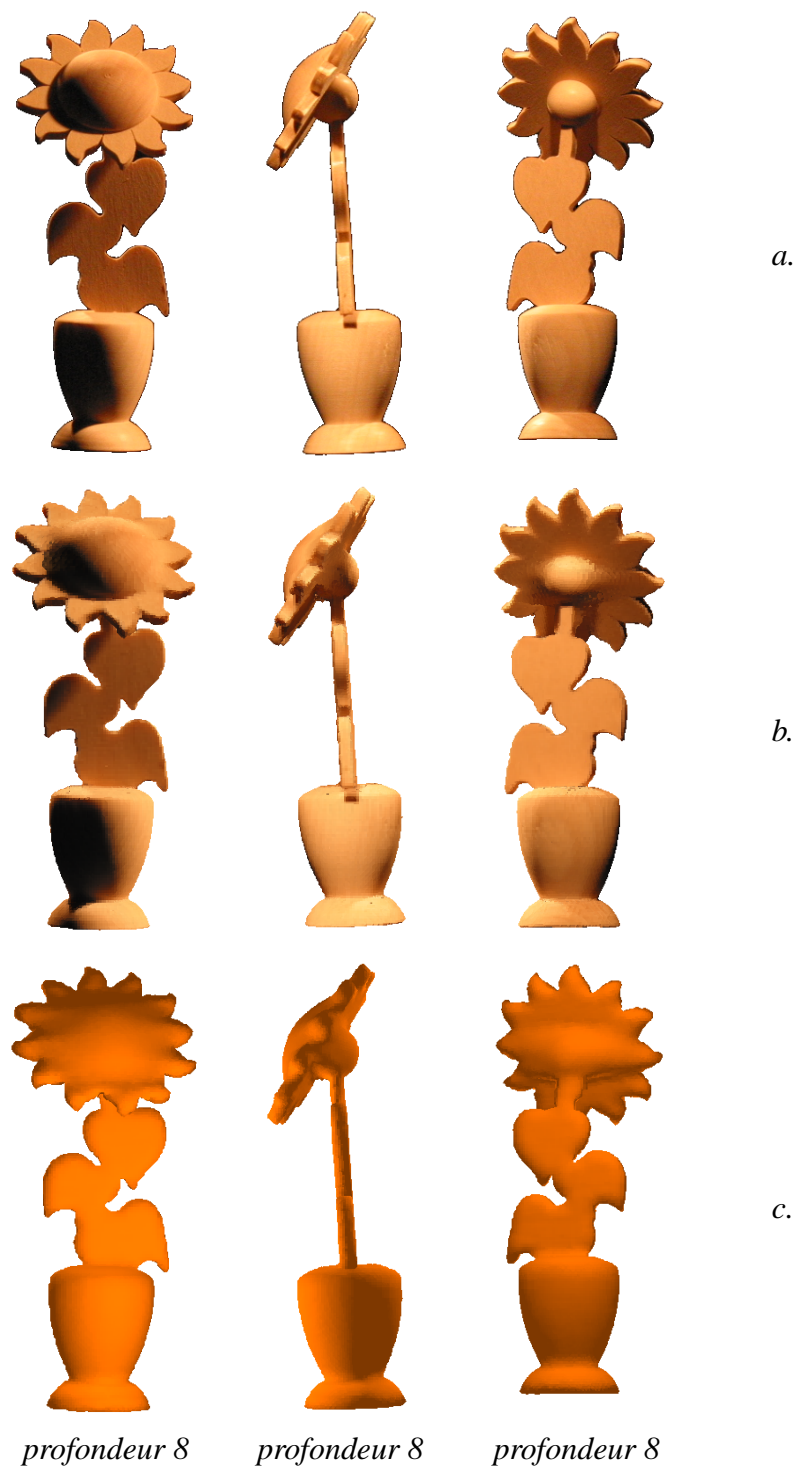


Figure 2.47 – Une série de 72 images de résolution  $512 \times 512$  est utilisée pour la reconstruction de cet objet réel. a. photographies d'un tournesol en bois d'aspect diffus ; b. rendu du maillage sans rééclairage ; c. rendu diffus du maillage.

la connaissance des caractéristiques géométriques des sources lumineuses (non disponibles dans notre approche).

L'imprécision sur le calibrage de caméra est particulièrement visible sur le chapeau du clown et sur le noeud papillon dans la vue de profil. Malgré une segmentation *large* des contours de l'objet sur les photographies (les pixels sombres aux extrémités des contours sont conservés), l'objet reconstruit est érodé par l'algorithme de sculpture utilisé. Après un meilleur calibrage de la caméra, cette érosion disparaît dans la reconstruction du tournesol. Le système d'acquisition final proposé dans ce manuscrit semble suffisant pour obtenir une reconstruction géométrique de qualité.





# DÉTECTION DE SOURCES LUMINEUSES ET ESTIMATION DE BRDFs

---

## Sommaire

---

<b>A</b>	<b>Représentation des données</b> . . . . .	<b>126</b>
A.1	Rayons incidents . . . . .	126
A.2	Modèles utilisés . . . . .	130
<b>B</b>	<b>Principe général</b> . . . . .	<b>132</b>
B.1	Classification des voxels selon la réflectance . . . . .	133
B.2	Classification des voxels selon l'éclairement . . . . .	136
B.3	Evaluation du type de surface . . . . .	137
B.4	Algorithme général . . . . .	138
B.5	Aspects techniques . . . . .	139
<b>C</b>	<b>Détection d'une source ponctuelle</b> . . . . .	<b>143</b>
C.1	Utilisation de la composante diffuse . . . . .	143
C.2	Utilisation de la composante spéculaire . . . . .	149
<b>D</b>	<b>Détection d'une source directionnelle</b> . . . . .	<b>152</b>
D.1	Utilisation de la composante diffuse . . . . .	152
D.2	Utilisation de la composante spéculaire . . . . .	153
<b>E</b>	<b>Estimation conjointe : sources lumineuses / propriétés de réflectance</b> . .	<b>153</b>
E.1	Surface d'aspect diffus . . . . .	154
E.2	Surface d'aspect diffus directionnel . . . . .	154
E.3	Validation de l'estimation conjointe . . . . .	155
E.4	Résultats . . . . .	156
<b>F</b>	<b>Raffinement sur chaque BRDF</b> . . . . .	<b>160</b>
<b>G</b>	<b>Raffinement sur l'espace image</b> . . . . .	<b>161</b>

---

Les objets utilisés sont représentés uniquement par un ensemble d'images calibrées et une reconstruction géométrique de leur surface est obtenue à partir de ces images (chapitre précédent, II). Les informations géométriques reconstruites sont à présent utilisées pour guider notre algorithme de détection de sources lumineuses multiples et d'estimation de propriétés de réflectance.

L'état de l'art sur les méthodes de détection de sources lumineuses montre que toutes les méthodes utilisent des hypothèses soit sur la géométrie de l'objet (connue ou reconstruite manuellement), soit sur la réflectance des matériaux (connue ou lambertienne), soit sur les deux (section C du chapitre I). Les méthodes permettant la détection de sources lumineuses multiples sont peu nombreuses et utilisent généralement une ou plusieurs sphères visibles sur l'ensemble des images afin de guider l'analyse : nous ne disposons pas de telles sphères sur les images acquises et ne voulons pas insérer ce type d'objet intrusif dans notre système d'acquisition. La méthode d'E. Guillou [Gui00] est la plus proche de ce que nous souhaitons obtenir, i.e. de multiples sources ponctuelles ou directionnelles, mais l'utilisation de cette méthode est limitée à des surfaces lambertiennes.

L'état de l'art sur les méthodes d'estimation de BRDFs et sur les méthodes de rééclairage d'objets réels montrent qu'il est préférable d'approcher la BRDF de chaque surface par un modèle paramétrique afin de réduire la quantité d'information stockée tout en préservant un rééclairage réaliste des objets (voir sections D et E, chapitre I). Néanmoins, l'estimation de modèles paramétriques repose sur les connaissances précises de la géométrie de l'objet et des conditions d'illumination.

Ne disposant pas de méthode d'estimation de sources lumineuses adaptée à nos données, nous proposons une nouvelle méthode pour détecter de sources lumineuses multiples à partir d'un ensemble d'images calibrées et d'informations géométriques reconstruites. Nous proposons ensuite d'utiliser les sources lumineuses détectées afin d'estimer conjointement leurs caractéristiques et les propriétés de réflectance des surfaces : cette estimation conjointe valide ainsi chaque source lumineuse et fournit un modèle paramétrique de la réflectance des surfaces.

## A Représentation des données

### A.1 Rayons incidents

#### A.1.1 Fonctionnement général d'une caméra

Dans un appareil photographique, les rayons lumineux parallèles traversant le diaphragme  $dS_D$  sont déviés par l'objectif pour converger en un point situé sur le plan focal (voir figure 3.1). Seuls les rayons lumineux passant par le centre du diaphragme (point  $D$ ) ne sont pas déviés par l'objectif qui est collé au diaphragme et assimilable à une lentille mince. La distance focale  $f$  est déterminée en fonction de l'objectif utilisé. Lorsque la pellicule est positionnée sur le plan focal, la profondeur de champ est infinie.

Le dédoublement du plan image et du plan focal permet de modifier la profondeur de champ. Lors de l'acquisition de notre série d'image, le plan image est placé de manière à entièrement positionner l'objet dans la profondeur de champ et obtenir une image nette de l'objet. Ceci se traduit par le fait que tous les rayons lumineux issus d'un élément de surface  $dS_O$  sur l'objet et traversant le diaphragme sont captés par un même pixel  $dS_P$  sur le plan image (voir figure 3.2). La profondeur de champ peut être étendue en rétrécissant le diamètre du diaphragme : une perte

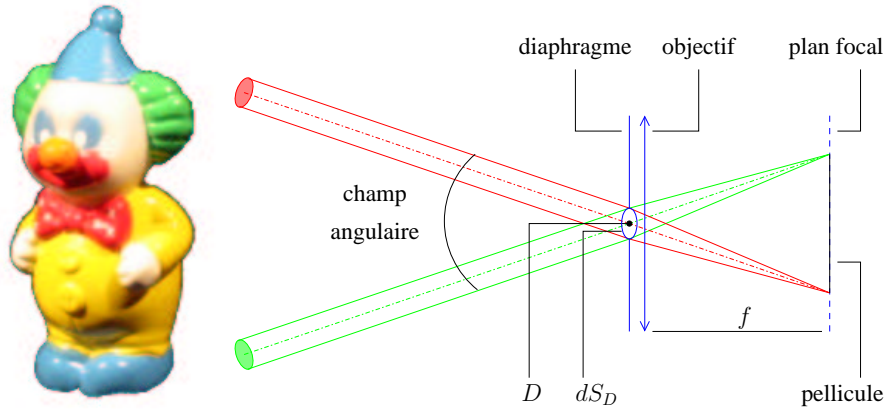


Figure 3.1 – Construction d'une image dans un appareil photographique où la profondeur de champ est réglée à l'infini. Tous les rayons parallèles convergent en un point sur le plan focal, i.e. en un point de la pellicule ou en un pixel d'un capteur photosensible (appareil photo numérique).

d'énergie lumineuse entrant dans la caméra est alors enregistrée, elle est compensée en fixant une vitesse d'obturation plus lente.

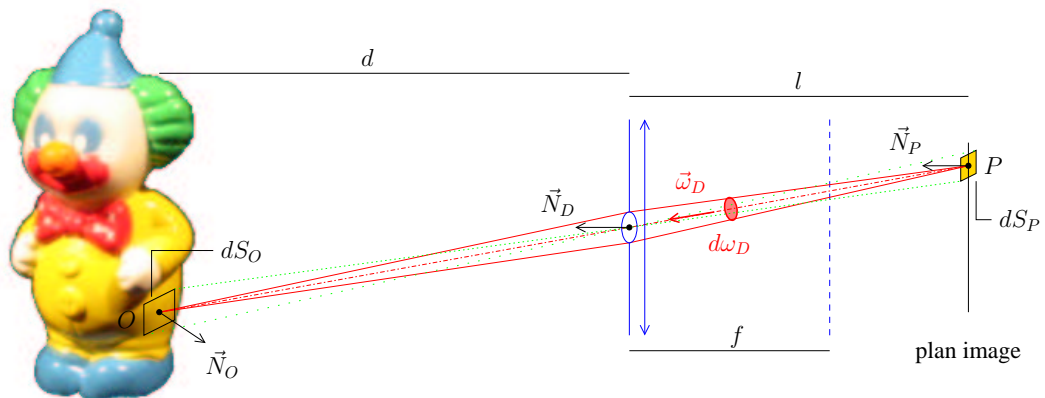


Figure 3.2 – Les rayons rouges montrent que le point  $O$  est dans la profondeur de champ de la caméra ; les rayons en pointillés verts montrent que l'énergie lumineuse émise par l'élément de surface  $dS_O$  en direction du diaphragme est entièrement captée par le pixel de surface  $dS_P$

### A.1.2 Mesure du flux lumineux

La figure 3.2 montre que le flux lumineux reçu par un pixel de surface  $dS_P$  provient de l'angle solide  $d\omega_D$  formé par le diaphragme et correspond exactement à l'ensemble des luminances émises par un élément de surface  $dS_O$  de l'objet en direction du diaphragme. Ce flux lumineux, correspondant à une énergie par unité de temps, peut être mis en équation :

$$\Phi(dS_D \rightarrow dS_P) = \Phi(dS_O \rightarrow dS_D) \quad (3.1)$$

Chaque valeur stockée dans un pixel non noir d'une image correspond à une énergie émise par un élément de surface de l'objet pendant un certain laps de temps, i.e. le produit du flux lumineux

émis par le temps d'exposition (ou vitesse d'obturation). Comme le temps d'exposition est fixe pendant toute la phase d'acquisition, la valeur des pixels correspond à une valeur de flux, à une constante près.

L'équation du flux peut être réécrite en fonction de la luminance reçue au centre du pixel  $P$ , en provenance du centre du diaphragme  $D$  (les équations des grandeurs physiques sont définies dans l'annexe I) :

$$\begin{aligned}\Phi(dS_D \rightarrow dS_P) &= \int_{d\omega_D} \left( \int_{dS_P} L_i(P, \vec{\omega}_D) \cdot \cos \theta \cdot dS \right) \cdot d\omega \\ &= \int_{d\omega_D} \left( \int_{dS_P} L_i(P, \vec{\omega}_D) \cdot (\vec{N}_P \cdot \vec{\omega}_D) \cdot dS \right) \cdot d\omega\end{aligned}\quad (3.2)$$

La surface  $dS_P$  représentant la surface du pixel  $P$  peut être considérée comme une surface élémentaire où la direction du centre du diaphragme  $\vec{\omega}_D$  ainsi que la luminance reçue en provenance du centre du diaphragme restent inchangées en tout point de  $dS_P$ . De plus, l'angle solide  $d\omega_D$  vu depuis le pixel  $P$  peut être considéré comme un angle solide élémentaire où toute direction prise dans cet angle solide est assimilable à la direction  $\vec{\omega}_D$ .

Par conséquent l'équation de flux peut se réécrire :

$$\begin{aligned}\Phi(dS_D \rightarrow dS_P) &= \int_{d\omega_D} L_i(P, \vec{\omega}_D) \cdot (\vec{N}_P \cdot \vec{\omega}_D) \cdot \left( \int_{dS_P} dS \right) \cdot d\omega \\ &= \int_{d\omega_D} L_i(P, \vec{\omega}_D) \cdot (\vec{N}_P \cdot \vec{\omega}_D) \cdot dS_P \cdot d\omega \\ &= L_i(P, \vec{\omega}_D) \cdot (\vec{N}_P \cdot \vec{\omega}_D) \cdot dS_P \cdot \int_{d\omega_D} d\omega \\ &= L_i(P, \vec{\omega}_D) \cdot (\vec{N}_P \cdot \vec{\omega}_D) \cdot dS_P \cdot d\omega_D \\ &= L_i(P, \vec{\omega}_D) \cdot \cos \theta_D \cdot dS_P \cdot d\omega_D\end{aligned}\quad (3.3)$$

avec  $(\vec{N}_P \cdot \vec{\omega}_D) = (\vec{N}_D \cdot \vec{\omega}_D) = \cos \theta_D$ ;  $\theta_D$  étant l'angle d'incidence du flux entrant dans la caméra.

L'angle solide  $d\omega_D$ , dépendant du pixel choisi, peut s'écrire :

$$\begin{aligned}d\omega_D &= \frac{dS_D \cdot \cos \theta_D}{(DP)^2} \\ &= \frac{dS_D \cdot \cos \theta_D}{(l / (\cos \theta_D))^2} \\ &= dS_D \cdot \cos^3 \theta_D \cdot \frac{1}{l^2}\end{aligned}\quad (3.4)$$

avec  $l$  la distance séparant le plan image du diaphragme.

Par conséquent, l'équation du flux peut s'exprimer par :

$$\Phi(dS_D \rightarrow dS_P) = L_i(P, \vec{\omega}_D) \cdot dS_P \cdot dS_D \cdot \frac{1}{l^2} \cdot \cos^4 \theta_D \quad (3.5)$$

avec  $dS_P$ ,  $dS_D$  et  $l$  constants sur toutes les images et pour tout pixel. Seuls  $L_i(P, \vec{\omega}_D)$  et  $\theta_D$  varient en fonction du pixel  $P$  choisi sur l'image.

Les pixels recevant de l'énergie en provenance de l'objet correspondent à une région d'environ  $256 \times 256$  pixels centrée sur des images de résolution  $640 \times 480$  pixels (zone grisée de la figure 3.3). La distance  $l$  étant fixe, l'angle  $\theta_D$  varie en fonction de  $b$  et prend une valeur maximale pour un pixel  $P$  situé dans un coin de la zone grisée. D'après les notations de la figure 3.3, l'angle  $\theta_D$  peut être exprimé par :

$$\begin{aligned}\cos^4 \theta_D &\geq \left( \frac{l}{\sqrt{l^2 + b^2}} \right)^4 \\ &\geq \frac{l^4}{(l^2 + 2a^2)^2}\end{aligned}\quad (3.6)$$

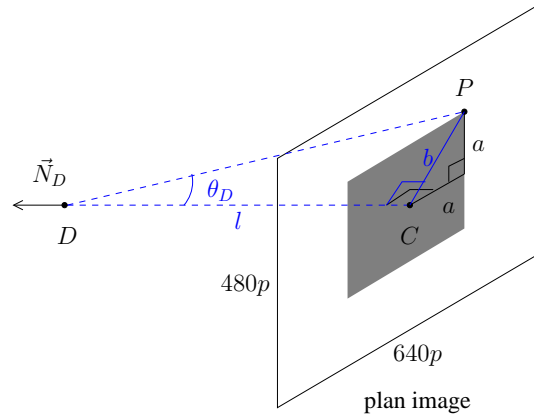


Figure 3.3 – Mesure angulaire de  $\theta_D$ , l'angle d'incidence pour le pixel  $P$

avec  $a$  une distance de 128 pixels. Comme la pellicule mesure  $24 \times 36mm$  pour  $480 \times 640$  pixels,  $a = 7.2mm$ . De plus comme  $\lim_{l \rightarrow \infty} \cos^4 \theta_D = 1$ ,  $l > f$  et  $f = 105mm$ , l'angle  $\theta_D$  peut être approché par :

$$\begin{aligned} \cos^4 \theta_D &> \frac{f^4}{(f^2 + 2a^2)^2} \\ &> \frac{105^4}{(105^2 + 2 \cdot 7.2^2)^2} \\ &> 0.98 \end{aligned} \quad (3.7)$$

Nous remarquons que l'influence de  $\cos^4 \theta_D$  sur le flux est négligeable, sa valeur peut être arrondie à 1. L'expression du flux peut ainsi être réduite à :

$$\begin{aligned} \Phi(dS_D \rightarrow dS_P) &\approx L_i(P, \vec{\omega}_D) \cdot dS_P \cdot dS_D \cdot \frac{1}{r^2} \\ &\approx L_i(P, \vec{\omega}_D) \cdot C \end{aligned} \quad (3.8)$$

avec  $C$  une constante (indépendante de  $P$  et  $\vec{\omega}_D$ ).

Par conséquent, nous assimilons la valeur énergétique stockée par le capteur d'un pixel à une luminance reçue par ce pixel en provenance du centre optique de la caméra et nous travaillons à présent avec des échantillons de luminance correspondant directement aux valeurs des pixels.

### A.1.3 Représentation des luminances

Chaque capteur de la caméra enregistre l'énergie reçue en un pixel sous la forme d'un triplet  $(R, G, B)$  correspondant à des valeurs de composantes rouge ( $R$ ), verte ( $G$ ) et bleue ( $B$ ) bornées entre 0 et 255. Pour obtenir la luminance émise à partir de ce triplet, nous changeons de système de représentation de la couleur afin d'utiliser la composante  $Y$  de l'espace  $XYZ$  comme représentation de la luminance reçue par chaque pixel.

Une seule série d'images (un tour complet de photographies) est utilisée pour reconstruire précisément la surface d'un objet réel (chapitre II). En revanche, l'estimation de l'ensemble des sources lumineuses et des propriétés de réflectance des surfaces nécessite plus de données afin d'obtenir un échantillonnage représentatif de la BRDF des surfaces : par conséquent toutes les images sont utilisées.

Les pixels représentant la luminance émise par la surface de l'objet sont projetés sur l'octree afin d'obtenir, en chaque voxel de la surface, un échantillonnage dense de luminances provenant

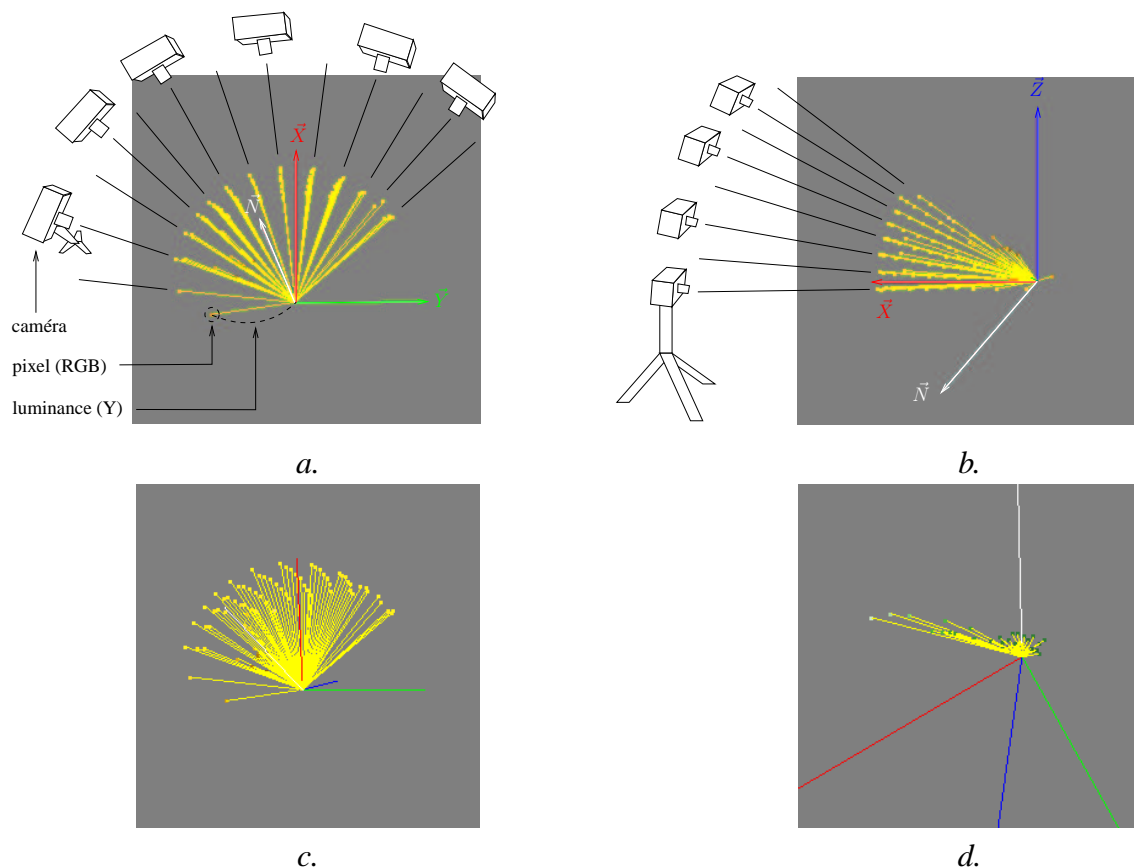


Figure 3.4 – Luminances émises par un voxel de la surface du clown, objet réel ; la longueur des segments jaunes représente l'information de luminance (valeur  $Y$ ) ; la couleur située à l'extrémité des segments jaunes représente le pixel (valeur  $RGB$ ) ; a. luminances émises en direction de plusieurs caméras sur une série d'images acquises autour de l'objet (rotation du plateau tournant) ; b. utilisation des 8 séries d'images acquises en faisant varier la hauteur de la caméra ; c. les échantillons de luminance de ce voxel montrent que la surface jaune est plutôt diffuse (la réflexion spéculaire n'est pas visible) ; d. sur cet autre voxel, les échantillons de luminance montrent que la surface verte est plutôt spéculaire.

de différentes directions de caméra (figures 3.4.a et 3.4.b). Malgré l'utilisation de cet ensemble de données, la spécularité de l'objet n'est pas toujours visible sur la surface reconstruite de l'objet (figures 3.4.c et 3.4.d).

## A.2 Modèles utilisés

### A.2.1 Types de sources détectées

Notre méthode permet d'estimer les paramètres de plusieurs sources lumineuses blanches éclairant un objet. Ces sources lumineuses peuvent être une combinaison de sources ponctuelles et directionnelles. La section C.1 du chapitre I donne une description précise des sources lumineuses utilisées en informatique graphique.

Les sources ponctuelles sont représentées par leurs coordonnées (point 3D) et leur valeur de luminance émise, constante dans toutes les directions d'émission (scalaire). Les sources direc-

tionnelles sont représentées par leur direction d'émission (vecteur 3D) et leur valeur de luminance émise (scalaire).

Les spots ne sont pas détectés car il est difficile d'évaluer le cône d'éclairage du spot sans aucune indication sur les propriétés de réflectance des surfaces. Les sources surfaciques sont également écartées car l'estimation de leur surface n'est pas triviale.

### A.2.2 Types de surfaces analysées

Dans ce travail, nous traitons des surfaces texturées ou non, comportant des propriétés de réflexion diffuse et/ou spéculaire. Nous supposons que les objets peuvent être composés de divers matériaux ayant chacun leurs propres propriétés de réflectance. Chaque matériau peut ainsi être soit parfaitement diffus, soit d'aspect diffus directionnel avec une direction de réflexion privilégiée : la direction de réflexion miroir (voir section [D.1](#) du chapitre I pour plus de détails). Les échantillons de luminance réelle montrés sur les figures [3.4.c](#) et [3.4.d](#) caractérisent ces deux types de surfaces.

Les surfaces anisotropes ne sont pas prises en compte car nous ne disposons pas de suffisamment d'échantillons de luminance par voxel. En effet, il faut un échantillonnage plus large sur la figure [3.4.b](#) (actuellement, l'angle d'élévation de la caméra est de  $45^\circ$  maximum).

### A.2.3 Modèles de BRDF utilisés

Nous choisissons d'approcher les propriétés de réflectance des surfaces par un modèle paramétrique de BRDF. Cette méthode permet de limiter la quantité d'information à stocker et également d'obtenir une valeur de réflectance pour les BRDFs incomplètes (c'est le cas lorsque les objets réels sont acquis sous un nombre de points de vue restreint). L'état de l'art sur l'estimation des propriétés de réflectance montre que cette solution est la plus souvent utilisée et donne des résultats exploitables pour un rééclairage réaliste (section [D](#) du chapitre I).

Nous avons choisi d'utiliser le modèle de Phong modifié [[Lew94](#), [LW94](#)] car c'est un modèle simple permettant d'exprimer des propriétés de réflectance de surfaces à la fois diffuses et spéculaires tout en respectant la loi de conservation de l'énergie et le principe de réciprocité de Helmholtz. Ce modèle paramétrique, exprimé de la façon suivante :

$$f_r(P, \vec{R}, \vec{I}) = K_d \cdot \frac{1}{\pi} + K_s \cdot \frac{n+2}{2\pi} \cdot \cos^n \phi \quad (3.9)$$

permet de représenter les surfaces diffuses par un seul coefficient ( $K_d$ ) et les surfaces à composante spéculaire par deux coefficients supplémentaires ( $K_s$  et  $n$ ). Plus de détails sur ce modèle sont disponibles dans la section [D.2](#) du chapitre I.

### A.2.4 Systèmes de représentation de la couleur utilisés

Nous utilisons plusieurs systèmes de représentation de la couleur, chacun étant utilisé pour obtenir un type d'information bien précis. L'espace *RGB* est le système de représentation de la couleur le plus utilisé car il correspond au format de communication avec le matériel : appareil photographique, carte graphique et écran. Il est donc naturel de travailler et d'exprimer les images finales produites dans cet espace.

Néanmoins, d'autres systèmes de représentation existent, ils sont définis pour des applications spécifiques. Nous avons vu à la section [A.1.3](#) que la composante *Y* de l'espace *CIEXYZ*



est utilisée pour représenter la luminance reçue par un pixel. Cette composante est usuellement considérée comme la luminance du spectre incident car elle représente approximativement la sensibilité de l’œil humain à la luminosité. La valeur  $Y$  est calculée par combinaison linéaire des composantes de l’espace  $RGB$  et bornée sur l’intervalle  $[0, 1]$  :

$$Y = (0.265 * R + 0.717 * G + 0.009 * B) / 252.705 \quad (3.10)$$

Nous avons essayé d’autres systèmes de représentation de la luminance ( $Lab$  et  $Luv$ ) mais après expérimentations, le système  $XYZ$  a été choisi car il donne les meilleurs résultats pour notre méthode de détection de sources lumineuses.

Nous utilisons également l’espace  $HSV$  défini comme un système perceptuel, i.e. un système de représentation de la couleur répondant à la perception humaine. Il est défini par une valeur de teinte ( $H$ , *hue*), une valeur de saturation ( $S$ ) et une valeur de luminosité ( $V$ , *value*). La teinte correspond à une primitive de couleur (rouge, verte, jaune, etc), la saturation à la pureté de la couleur (pastel, délavée, etc) et la luminosité au taux de blanc présent (clair, foncé, etc). Nous utilisons ce système car la teinte d’une surface est insensible à son éclaircissement et la saturation donne une indication sur la fiabilité de la teinte estimée. Voici comment sont calculées les valeurs  $H$  et  $S$  de cet espace (les calculs sont légèrement différents de ceux trouvés dans la littérature afin de borner les valeurs sur l’intervalle  $[0, 1]$ ) :

$$H = \begin{cases} \text{non exprimable} & \text{si } \max(R, G, B) = \min(R, G, B), \\ \left(1 + \frac{G-B}{\max(R,G,B)-\min(R,G,B)}\right) / 6 & \text{si } \max(R, G, B) = R, \\ \left(3 + \frac{B-R}{\max(R,G,B)-\min(R,G,B)}\right) / 6 & \text{si } \max(R, G, B) = G, \\ \left(5 + \frac{R-G}{\max(R,G,B)-\min(R,G,B)}\right) / 6 & \text{si } \max(R, G, B) = B. \end{cases} \quad (3.11)$$

$$S = \begin{cases} 0 & \text{si } \max(R, G, B) = 0, \\ \frac{\max(R,G,B)-\min(R,G,B)}{\max(R,G,B)} & \text{sinon.} \end{cases} \quad (3.12)$$

Nous pouvons remarquer que la teinte n’est pas exploitable lorsque la saturation est nulle : cela correspond aux différents niveaux de gris. Ce système peut être vu comme un cône hexagonal car la variation linéaire de la teinte permet de représenter toutes les couleurs primaires avec une continuité de la teinte entre chacune des trois fonctions de calcul. Ainsi les valeurs de teintes sont cycliques,  $H = 0$  et  $H = 1$  représentent la même primitive magenta (voir figure 3.5).

## B Principe général

Nous utilisons à présent les échantillons de luminance, stockés dans les voxels de la surface discrète, et les informations géométriques reconstruites afin de détecter l’ensemble des sources lumineuses éclairant l’objet et les propriétés de réflectance de ses surfaces. Evidemment, le problème n’est pas solvable sous cette définition : nous regroupons donc les informations en différentes classes afin de restreindre le problème à l’estimation d’une source lumineuse et d’un type de surface. Nous proposons ensuite différentes stratégies de résolution en fonction des types de données à analyser.



Figure 3.5 – Cône hexagonal représentant les valeurs de teinte, saturation et luminosité de l'espace HSV ; l'axe achromatique représente les différents niveaux de gris (valeur de luminosité) avec des valeurs de saturation nulles et des valeurs de teinte non calculables ; l'éloignement de l'axe achromatique représente la valeur de saturation (maximale sur le bord du cône) ; la valeur angulaire autour de l'axe achromatique représente la valeur de teinte (à une constante près).

## B.1 Classification des voxels selon la réflectance

L'analyse des échantillons de luminance émise par chaque voxel permet de déduire des informations sur les propriétés de réflectance de ce voxel et ainsi de regrouper les voxels de même réflectance. Pour cela, nous analysons les échantillons de luminance émise par les voxels en utilisant le système perceptuel HSV : ce système a la particularité d'avoir une composante invariante selon l'éclairage.

### B.1.1 Propriétés du système perceptuel HSV

La figure 3.6 montre une représentation des valeurs de teinte et de saturation sur une image d'un de nos objets synthétisés sous PovRay : l'indépendance de la teinte vis-à-vis de l'éclairage des surfaces est nettement visible sur cette image. Nous montrons cette indépendance en utilisant la définition de la teinte et en l'évaluant à partir de l'équation de luminance d'un pixel.

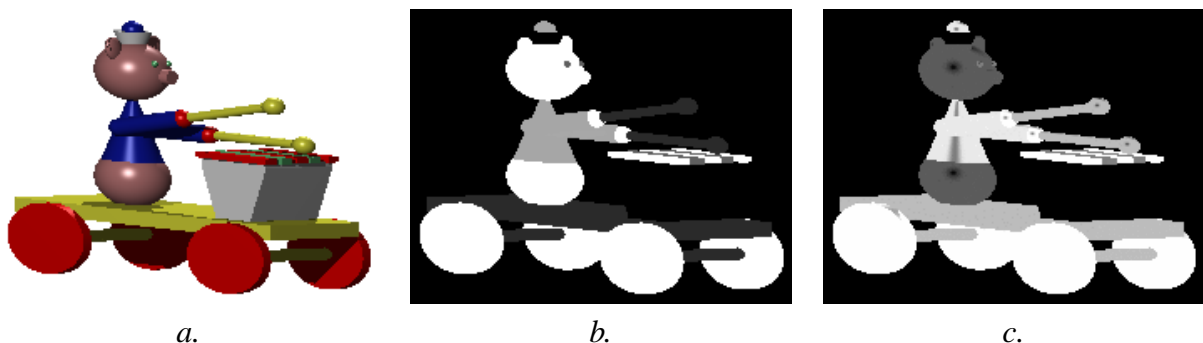


Figure 3.6 – a. Image de synthèse représentée dans l'espace RGB ; b. représentation de la teinte ; c. représentation de la saturation.

L'objet est éclairé par 3 sources lumineuses blanches et est composé de surfaces simulant l'impression de plastique avec des coefficients de réflexion diffuse variant selon la composante chromatique ( $R, G$  ou  $B$ ) et des coefficients de réflexion spéculaire identiques pour les trois composantes  $R, G$  et  $B$ . Ecrivons l'équation de la luminance émise par un point  $O$  en direction d'un pixel  $P$ , avec  $O$  appartenant à une surface de l'objet dont nous connaissons les coefficients de réflexion dans le modèle de Phong modifié ( $K_{d,R}, K_{d,G}, K_{d,B}, K_s$  et  $n$ ) :

$$\begin{aligned}
 C(P) &= \sum_{s=1}^{NbSV} L_i(O, \vec{I}_s) \cdot \left( K_{d,C} \cdot \frac{1}{\pi} + K_s \cdot \frac{n+2}{2\pi} \cdot \cos^n \phi_s \right) \cdot \cos \theta_s \cdot \frac{1}{r_s^2} \\
 &= K_{d,C} \cdot \sum_{s=1}^{NbSV} \left( L_i(O, \vec{I}_s) \cdot \frac{1}{\pi \cdot r_s^2} \cdot \cos \theta_s \right) + K_s \cdot \sum_{s=1}^{NbSV} \left( L_i(O, \vec{I}_s) \cdot \frac{n+2}{2\pi \cdot r_s^2} \cdot \cos^n \phi_s \cdot \cos \theta_s \right) \\
 &= K_{d,C} \cdot D(O, P) + K_s \cdot S(O, P)
 \end{aligned} \tag{3.13}$$

avec  $C(P)$  la composante chromatique  $R, G$  ou  $B$  reçue par le pixel  $P$ ;  $NbSV$  le nombre de sources lumineuses visibles depuis le point  $O$ ;  $K_{d,C} \cdot D(O, P)$  la composante d'illumination diffuse émise par le point  $O$  en direction du pixel  $P$  et  $K_s \cdot S(O, P)$  la composante d'illumination spéculaire.

Exprimons à présent la teinte  $H$  du pixel  $P$  en fonction des notations précédemment définies, en émettant l'hypothèse que  $R > G > B$  :

$$\begin{aligned}
 H &= \left( 1 + \frac{G-B}{R-B} \right) / 6 \\
 &= \left( 1 + \frac{K_{d,G} \cdot D(O, P) + K_s \cdot S(O, P) - K_{d,B} \cdot D(O, P) - K_s \cdot S(O, P)}{K_{d,R} \cdot D(O, P) + K_s \cdot S(O, P) - K_{d,B} \cdot D(O, P) - K_s \cdot S(O, P)} \right) / 6 \\
 &= \left( 1 + \frac{D(O, P) \cdot (K_{d,G} - K_{d,B})}{D(O, P) \cdot (K_{d,R} - K_{d,B})} \right) / 6 \\
 &= \left( 1 + \frac{K_{d,G} - K_{d,B}}{K_{d,R} - K_{d,B}} \right) / 6
 \end{aligned} \tag{3.14}$$

Nous avons montré que la teinte d'un pixel  $P$  dépend uniquement des coefficients de réflexion diffuse de la surface ( $K_{d,R}, K_{d,G}$  et  $K_{d,B}$ ), et par conséquent qu'elle est invariante selon l'éclairage.

Vérifions à présent comment se comporte la saturation  $S$  d'un pixel  $P$  :

$$\begin{aligned}
 S &= \frac{R-B}{R} \\
 &= \frac{K_{d,R} \cdot D(O, P) + K_s \cdot S(O, P) - K_{d,B} \cdot D(O, P) - K_s \cdot S(O, P)}{K_{d,R} \cdot D(O, P) + K_s \cdot S(O, P)} \\
 &= \frac{D(O, P) \cdot (K_{d,R} - K_{d,B})}{D(O, P) \cdot (K_{d,R} + K_s \cdot S(O, P) / D(O, P))} \\
 &= \frac{K_{d,R} - K_{d,B}}{K_{d,R} + K_s \cdot S(O, P) / D(O, P)}
 \end{aligned} \tag{3.15}$$

Si la surface est purement diffuse ( $K_s = 0$ ) ou vue uniquement depuis des directions éloignées de la direction de réflexion miroir ( $S(O, P) = 0$ ), alors la saturation de la surface est invariable selon l'éclairage. En revanche, la saturation d'une surface spéculaire diminue lorsque la direction de vue se rapproche de la direction de réflexion miroir.

## B.1.2 Utilisation de la teinte

La valeur de teinte d'espace  $HSV$  est donc utilisée pour séparer les surfaces en fonction de leurs propriétés de réflectance. Il est plus difficile d'utiliser la saturation comme critère de classification car elle varie en fonction de la composante d'illumination spéculaire. La solution idéale est de n'utiliser que les échantillons de luminance qui ne sont pas dans le cône de direction de réflexion miroir afin d'effectuer une classification des voxels en fonction de la teinte et de la

saturation. Comme la surface peut être éclairée par plusieurs sources lumineuses, plusieurs cônes de direction miroir peuvent exister, restreignant le nombre d'échantillons utilisables par voxel. De plus, même si la saturation d'une surface est invariable pour la composante d'illumination diffuse, l'utilisation d'images réelles complique l'estimation : la BRDF de surfaces réelles n'a pas toujours les propriétés attendues. De plus les composantes  $R$ ,  $G$  et  $B$  sont stockées sur l'intervalle  $[0, 255]$  et lorsqu'il y a des pixels sur-exposés sur une photographie, i.e. lorsque certaines valeurs sont bornées à 255 alors qu'elles devraient être théoriquement supérieures, la valeur de teinte et de saturation de ces pixels est erronée (la saturation est plus forte, la teinte est moins précise et le pixel a un ton plus pastel).

Par conséquent, nous effectuons une classification des voxels uniquement en fonction de la teinte, avec une valeur de saturation supérieure à un seuil afin de valider la pertinence de la valeur de teinte calculée. Lorsque la saturation est faible, la teinte calculée n'est pas fiable car la couleur se rapproche d'un niveau de gris (axe achromatique de la figure 3.5) : ce voxel est considéré comme une surface particulière (voir section B.5 pour les détails). La figure 3.7 montre l'exemple de la classification des surfaces du clown, objet réel, en fonction de la teinte. La figure

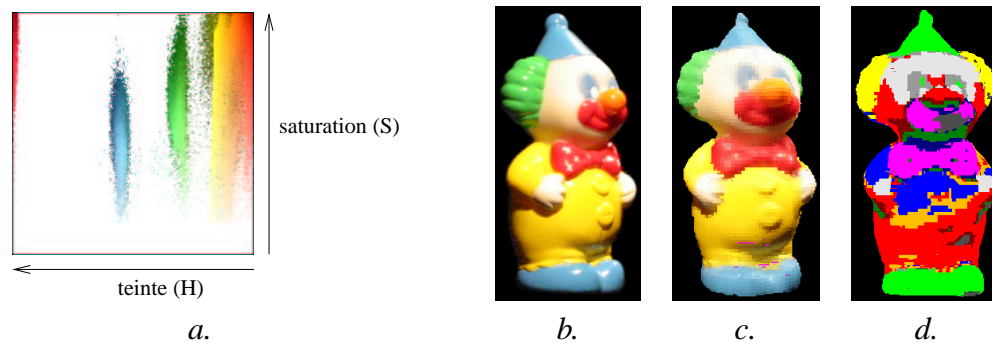


Figure 3.7 – a. Classification des pixels du clown en fonction de leur teinte et de leur saturation (les pixels dont la saturation est trop faible sont déjà otés) ; b. photographie du clown ; c. objet reconstruit au niveau 7 avec une couleur par voxel, correspondant à la moyenne de ses échantillons de luminance ; d. objet séparé en 9 surfaces de réflectance uniforme avec une couleur par classe de voxels (la couleur de chaque classe est choisie aléatoirement et n'est pas représentative de la teinte de la surface).

a montre que la teinte varie peu avec l'éclairage. En revanche, la valeur de saturation varie fortement selon l'illumination et ne peut donc pas être utilisée comme critère de classification. La figure d montre le résultat final de la classification des voxels selon la teinte : les voxels sont correctement groupés en 9 régions, non nécessairement connexes.

Cette classification, uniquement en fonction de la teinte, ne permet donc pas de séparer les surfaces de même teinte avec des valeurs de saturation différentes : par exemple, deux surfaces de teinte bleue, dont l'une représente un ton bleu vif et l'autre un ton bleu délavé (pastel), sont considérées comme une seule surface avec des propriétés de réflectance uniforme. Par conséquent, nous faisons l'hypothèse que les objets utilisés n'ont pas de nuances d'une même couleur avec des tons vifs et des tons pastels : c'est le cas du clown utilisé (ces nuances apparaissent uniquement par les conditions d'illumination). Généralement, les objets ont des nuances de teinte avec des couleurs soit toutes vives, soit toutes délavées afin de préserver une certaine harmonie entre les couleurs. Le problème des tons vifs et délavés employés simultanément pour une même couleur provient souvent des surfaces fortement texturées.

Des détails sur la classification selon la réflectance sont donnés dans les aspects techniques, section B.5. Nous avons développé un algorithme de classification qui détermine automatiquement le nombre de classes à partir de l'analyse d'un histogramme des teintes. Cette classification est plus intéressante que celle proposée par H. Lensch et al. où le nombre de classes de réflectance doit être fourni par l'utilisateur à l'algorithme des k-means [LKG<sup>+</sup>03].

## B.2 Classification des voxels selon l'éclairement

La classification selon la réflectance permet d'utiliser un groupe de voxels disposant des mêmes propriétés de réflectance pour estimer les conditions d'illumination et la BRDF de la surface. Le problème initial est simplifié par l'estimation d'une seule BRDF par groupe de voxels mais la détermination de l'ensemble des conditions d'illumination est encore difficile à réaliser.

### B.2.1 Détection de plusieurs lobes spéculaires

Nous proposons, dans un premier temps, d'écarter les voxels éclairés par plusieurs sources lumineuses afin de n'utiliser, pour chaque région, que des voxels éclairés par une seule source lumineuse : cette sélection est effectuée par une analyse des échantillons de luminance. Lorsque plusieurs lobes spéculaires sont détectés pour un même voxel, alors celui-ci est écarté du processus de détection de sources.

La détection de lobes spéculaires implique nécessairement une composante spéculaire non nulle sur les surfaces de l'objet. C'est généralement le cas avec les surfaces réelles, elles sont rarement parfaitement diffuses. Si malgré cela, des voxels éclairés par plusieurs sources sont gardés, ils perturbent l'algorithme d'estimation de la source lumineuse mais un critère d'erreur est évalué plus tard afin de conserver une cohérence sur l'ensemble des données estimées (voir section E.3).

L'utilisation d'un ensemble restreint de points de vue (variation de  $40^\circ$  de l'angle d'élévation de la caméra) ne nuit pas à la détection des lobes spéculaires puisque la détection se fait par analyse des variations de valeurs de luminances dans toutes les directions : elle ne nécessite pas de disposer de valeurs de luminance dans chaque direction de réflexion miroir mais dans chaque direction de lobe spéculaire.

### B.2.2 Utilisation de la normale

Les informations de géométrie reconstruite sont ensuite utilisées pour effectuer une nouvelle classification sur les voxels restants et obtenir de nouveaux groupes de voxels plus restreints, où tous les voxels d'un même groupe sont supposés avoir la même BRDF et être éclairés par une seule et même source lumineuse. Pour parvenir à ce résultat, les voxels de chaque groupe sont reclassés en fonction de l'orientation de leur surface : la normale estimée en chaque voxel est donc le critère de classification utilisé.

Pour une direction donnée, tous les voxels d'un groupe dont la normale est orientée dans cette direction, avec un angle de déviation possible de  $10^\circ$  maximum, sont de nouveau groupés ensemble. Nous utilisons un algorithme itératif qui, à chaque itération, choisit le voxel non classé de plus forte luminance et construit un nouveau groupe de voxel centré sur celui-ci. Nous choisissons d'utiliser les voxels de forte luminance pour classer les voxels car ce sont des voxels

qui peuvent contenir des informations d'illumination exploitables comme par exemple un lobe spéculaire.

### B.3 Evaluation du type de surface

Nous pouvons à présent analyser chaque groupe de voxels afin d'estimer la source lumineuse éclairant les voxels et les propriétés de réflectance de la surface représentée. Deux algorithmes différents sont développés pour estimer une source lumineuse. Chacun s'appuie sur des caractéristiques différentes de la surface éclairée : l'un utilise la propriété diffuse de la surface, l'autre utilise l'information de spécularité.

Par conséquent, chaque groupe de voxels est de nouveau analysé afin de déterminer les groupes de voxels correspondant à une surface parfaitement lambertienne (pas ou peu de variation de la luminance entre les différents échantillons d'un même voxel) et ceux correspondant à une surface d'aspect diffus directionnel où beaucoup d'échantillons sont exploitables dans la direction du lobe de spécularité. Evidemment, il n'y a pas de limite franche entre ces deux catégories de surfaces : certaines peuvent contenir quelques informations de spécularité et il n'est pas facile, a priori, de savoir si ces informations suffisent à guider convenablement l'algorithme d'estimation de source lumineuse basé sur les informations de spécularité.

Généralement, tous les voxels d'une même classe devraient contenir une proportion similaire d'échantillons de luminance dans la direction de leur lobe spéculaire car ils ont une normale similaire. Néanmoins, même si la classification selon la normale groupe les voxels dont la normale varie de moins de  $10^\circ$  par rapport à une direction donnée, les normales des voxels d'une même classe peuvent avoir jusqu'à  $20^\circ$  de différence et pour une direction d'incidence fixe, leur direction de spécularité peut varier de  $40^\circ$  dans le pire des cas.

Pour atténuer les différences entre les voxels d'une même classe, un critère de dispersion des luminances est évalué sur l'ensemble des voxels de chaque groupe. N'ayant aucune précision sur la puissance des sources lumineuses éclairant l'objet, nous appliquons un écart-type relatif aux luminances d'un voxel  $V$  :

$$\sigma_r(V) = \sqrt{\frac{1}{NbL} \cdot \sum_{i=1}^{NbL} \left( \frac{L_i - L^{moy}}{L^{moy}} \right)^2} \quad (3.16)$$

avec  $NbL$  le nombre d'échantillons de luminance du voxel  $V$  ;  $L_i$  la luminance du  $i^e$  échantillon représentant un pixel (cette luminance est exprimée par la valeur  $Y$  de l'espace couleur  $XYZ$ ) et  $L^{moy}$  la luminance moyenne du voxel  $V$ .

En réalité, cette définition est étendue à l'ensemble des échantillons d'une classe de voxels  $C$  avec une normalisation des écarts de luminance sur chaque voxel :

$$\sigma_r(C) = \sqrt{\frac{1}{\sum_{j=1}^{NbV} NbL_j} \cdot \sum_{j=1}^{NbV} \sum_{i=1}^{NbL_j} \left( \frac{L_{i,j} - L_j^{moy}}{L_j^{moy}} \right)^2} \quad (3.17)$$

avec  $NbV$  le nombre de voxels de la classe  $C$  ;  $NbL_j$  le nombre d'échantillons de luminance du voxel  $V_j$  ;  $L_{i,j}$  la luminance du  $i^e$  échantillon du voxel  $V_j$  et  $L_j^{moy}$  la luminance moyenne du voxel  $V_j$ .

Lorsque la surface est parfaitement lambertienne,  $L_{i,j} - L_j^{moy} = 0$  pour tous les échantillons et l'écart type relatif  $\sigma_r(C)$  correspond à la valeur nulle. Cet écart augmente lorsque la surface

admet une composante de réflexion spéculaire et que les voxels disposent d'échantillons dans la direction de leur lobe spéculaire. La valeur de  $\sigma_r(C)$  donne ainsi une estimation du type de surface représenté par les échantillons. Nous avons défini expérimentalement des seuils pour la valeur de  $\sigma_r(C)$  :

- si  $\sigma_r(C) < 0.15$ , la surface représentée par les voxels de la classe  $C$  possède des propriétés très diffuses : nous utilisons donc un algorithme de détection de source lumineuse basé sur la composante diffuse de la surface et estimons uniquement le coefficient de réflexion diffuse  $K_d$  ;
- si  $\sigma_r(C) > 0.30$ , la surface représentée par les voxels de la classe  $C$  dispose de propriétés spéculaires exploitables : nous utilisons de préférence un algorithme de détection de source lumineuse basé sur la composante de réflexion spéculaire de la surface et estimons tous les coefficients de réflexion du modèle de Phong modifié ( $K_d$ ,  $K_s$  et  $n$ ) ;
- si  $0.15 \leq \sigma_r(C) \leq 0.30$ , la surface, représentée par les voxels de la classe  $C$ , possède des échantillons de luminance dans la direction du lobe spéculaire mais en nombre insuffisant pour être exploitables à coup sûr : les deux algorithmes de détection de source lumineuse sont alors utilisés (ceux présentés ci-dessus), produisant chacun une source lumineuse et une estimation différente des propriétés de réflectance de la surface ; un critère d'erreur est ensuite évalué afin de garder les résultats les plus pertinents (section E.3).

### B.4 Algorithme général

Les deux algorithmes de détection de source, soit celui utilisant les propriétés diffuses de la surface soit celui utilisant les propriétés spéculaires de la surface, disposent de deux méthodes de détection d'une source lumineuse, dépendant du type de source à évaluer : une première estimant les caractéristiques d'une source ponctuelle, une seconde estimant les caractéristiques d'une source directionnelle. Au final, notre méthode générale analysant les échantillons de luminance de chaque groupe de voxels peut utiliser jusqu'à 4 méthodes différentes de détection de source lumineuse, chacune produisant son propre résultat :

- détection d'une source ponctuelle en utilisant la composante diffuse (section C.1) ;
- détection d'une source ponctuelle en utilisant la composante spéculaire (section C.2) ;
- détection d'une source directionnelle en utilisant la composante diffuse (section D.1) ;
- détection d'une source directionnelle en utilisant la composante spéculaire (section D.2).

Les propriétés de réflectance de la surface sont ensuite estimées pour chaque groupe de voxels et pour chaque source lumineuse détectée par une identification des paramètres du modèle de Phong modifié (voir section E). La luminance de chaque échantillon est évaluée à partir d'un algorithme d'illumination locale, puis comparée aux échantillons de luminance réelle afin d'estimer une erreur sur chaque source lumineuse détectée. Une mise en correspondance de toutes les sources lumineuses détectées est effectuée pour garder uniquement les sources lumineuses les plus pertinentes (dont l'erreur estimée est la plus faible) et définir des propriétés de réflectance uniforme sur chaque classe de voxels obtenue par la classification selon la réflectance.

La section F montre comment obtenir de meilleurs résultats sur l'estimation des propriétés de réflectance en réutilisant les informations des sources lumineuses les plus pertinentes.

La section G décrit comment obtenir de meilleurs résultats sur l'estimation des sources lumineuses en détectant les halos spéculaires apparaissant sur les images.



## B.5 Aspects techniques

### B.5.1 Stockage des échantillons de luminance

Chaque échantillon de luminance, défini à la section A.1.3, est obtenu par projection sur l'octree d'un pixel correspondant à l'objet et par stockage de l'information dans le premier voxel traversé : le maillage reconstruit à l'intérieur de chaque voxel est utilisé pour évaluer l'intersection entre un rayon et un voxel. Chaque échantillon est stocké dans le voxel traversé sous la forme d'un numéro d'image et de coordonnées 2D de pixel, exactement comme le stockage des rayons d'arrière-plan présenté au chapitre II (voir section B.6.2). Le stockage sous une forme aussi condensée (6 octets) permet de limiter la taille due aux échantillons tout en préservant au maximum leurs informations : pour chaque échantillon, aussi bien les coordonnées du rayon que les transformations du pixel dans les espaces  $XYZ$  ou  $HSV$  peuvent être évaluées.

Le processus de stockage des échantillons est complété par une analyse de leurs propriétés géométriques et photométriques vérifiant que chaque échantillon stocké est pertinent et ne perturbe pas les algorithmes d'estimation de sources et de BRDFs.

#### Utilisation de la géométrie

Lors de la projection d'un pixel sur l'octree, le point d'intersection est calculé avec le maillage reconstruit dans chaque voxel traversé par le rayon correspondant. Non seulement l'algorithme donne les coordonnées du premier voxel *touché* par le rayon (voxel  $V_0$  de la figure 3.8.a) mais il fournit également les coordonnées du premier voxel *frôlé* par le rayon (voxel  $V_5$  de cette même figure). Un voxel est considéré frôlé par un rayon lorsque la distance séparant le rayon du maillage reconstruit est inférieure à la demi-largeur du voxel. Lors de la projection d'un pixel, si le pre-

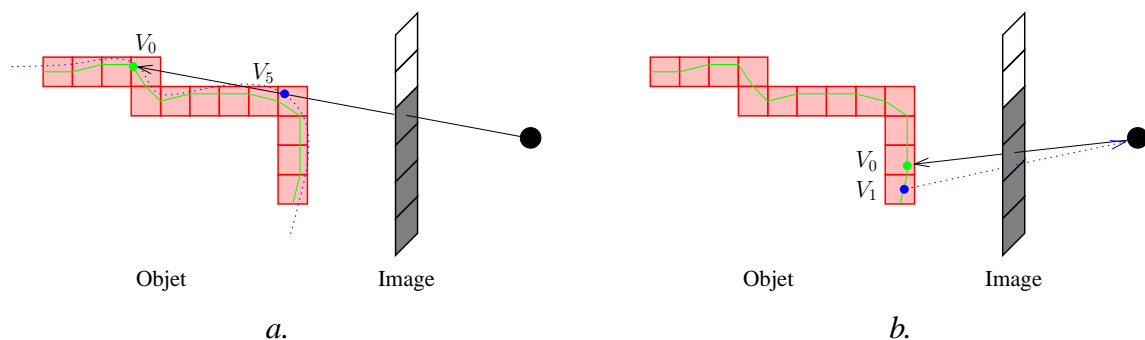


Figure 3.8 – a. La différence entre l'intersection calculée sur la surface reconstruite (point vert) et l'intersection théorique sur la surface réelle (point bleu sur les pointillés) peut être importante pour des rayons rasants ; b. pour lutter contre le sous-échantillonnage de luminance dans les voxels, un même pixel peut être stocké dans plusieurs voxels.

mier voxel frôlé est distant de plus de trois voxels du premier voxel touché alors l'échantillon de luminance correspondant n'est stocké dans aucun voxel : ce rayon rase la surface de l'objet et il y a ambiguïté sur son point d'intersection avec l'objet. En effet, l'objet représenté par le maillage reconstruit peut avoir un volume global plus petit que le volume de l'objet réel (la surface réelle est représentée par des pointillés bleus sur la figure 3.8.a), entraînant un point d'intersection réelle situé dans un autre voxel (point bleu sur la figure). Le rétrécissement du volume de l'objet est essentiellement dû à l'étalonnage de la caméra.

Le nombre d'échantillons stockés par voxel varie en fonction du niveau de reconstruction de l'octree. Lorsque l'octree est très détaillée, certains voxels n'ont aucun échantillon en provenance d'images dans la direction de leur vecteur normal. Pour combler ce manque nous proposons, lors de la projection d'un pixel sur un voxel ( $V_0$  sur la figure 3.8.b), de tracer un rayon en chemin inverse depuis les voxels voisins de  $V_0$  vers le centre de projection de la caméra : si le rayon traverse le même pixel à partir d'un voxel voisin ( $V_1$  sur la même figure) alors l'échantillon correspondant au pixel est également stocké dans ce voxel voisin. Ce processus permet d'augmenter le nombre d'échantillons par voxel mais allonge les temps de calculs sans réellement augmenter la précision de l'estimation des sources et des réflectances : l'utilisation de voxels de trop petite taille conduit généralement à l'estimation de sources lumineuses erronées sur de petits groupes de voxels contenant des échantillons parasites.

Au contraire, lorsque l'octree est peu détaillée, l'échantillonnage par voxel est très dense et au delà de 4 pixels par image projetés sur un même voxel, certains échantillons ne sont pas utilisables (chaque voxel est limité à 256 échantillons de luminance). Pour remédier à ce problème, nous effectuons une insertion triée des échantillons (tri en fonction de la luminance  $Y$ ) et ne gardons que les échantillons de plus forte luminance. Les échantillons de forte luminance sont peu nombreux car ils correspondent à des directions du lobe spéculaire et sont indispensables pour un algorithme d'estimation basé sur la composante spéculaire. Même en gardant un maximum d'échantillons de forte luminance, il reste encore suffisamment d'échantillons dans les directions de réflexion diffuse : une stratégie avec en moyenne 4 pixels d'une même image par voxel est le meilleur compromis (elle donne les meilleurs résultats sur la détection de sources lumineuses).

#### Utilisation de la photo-cohérence

Avant d'effectuer une classification des voxels selon la réflectance, la teinte moyenne de chaque voxel est évaluée et stockée dans la structure *FEUILLE* du voxel. La teinte moyenne d'un voxel est évaluée à partir de la teinte de chaque échantillon en prenant en considération le fait que la valeur de teinte calculée sur un échantillon n'est pas toujours fiable (la valeur de saturation permet de tester cette fiabilité). La teinte n'est pas utilisable lorsque l'échantillon correspond à un niveau de gris. Par conséquent, nous définissons une valeur de *teinte étendue* permettant de représenter trois pseudo-teintes supplémentaires correspondant à la *teinte blanche*, la *teinte grise* et la *teinte noire*.

Après expérimentation sur de multiples images d'objets, voici la définition de la valeur de *teinte étendue* que nous proposons pour chaque échantillon *RGB*, avec l'interprétation qui lui est associée :

- si  $\min(R, G, B) > 224$  alors l'échantillon a une *teinte blanche* : il provient soit des directions de réflexion spéculaire d'une surface de couleur claire, soit d'une surface blanche éclairée ;
- sinon si  $\max(R, G, B) < 64$  alors l'échantillon a une *teinte noire* : il provient soit d'une surface non éclairée, soit d'une surface noire ;
- sinon si  $S < 0.25 - 0.125 \cdot \left( \frac{\max(R, G, B) - 64}{255 - 64} \right)$  alors l'échantillon a une *teinte grise* : il provient soit des directions de réflexion spéculaire d'une surface de couleur sombre, soit d'une surface grise éclairée ;
- sinon l'échantillon a la valeur de *teinte H* obtenue par le système *HSV* : il provient d'une surface éclairée.

Travaillant essentiellement avec des images réelles, les échantillons de luminance stockés dans chaque voxel sont souvent bruités : quelques échantillons *parasites* apparaissent dans la liste

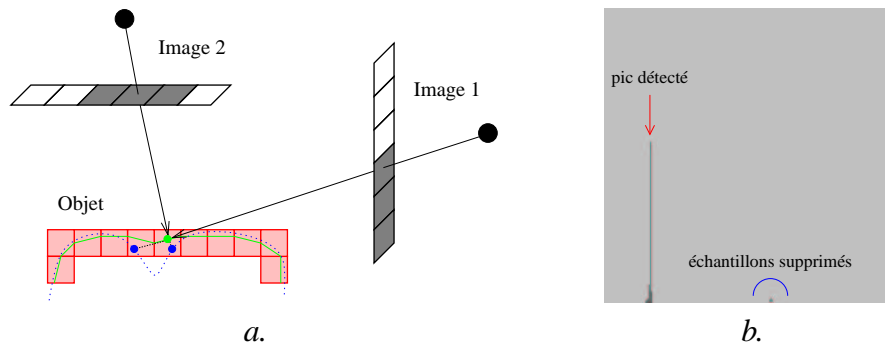


Figure 3.9 – a. Les petites cavités non reconstruites entraînent des imprécisions lors de la projection des pixels : les deux échantillons sont projetés sur le même voxel alors qu’ils appartiennent à deux surfaces orientées différemment ; b. histogramme des teintes d’un voxel, les teintes trop éloignées du pic correspondent à du bruit.

des luminances d’un voxel et, en utilisant la cohérence des valeurs de teinte des échantillons, nous pouvons détecter et supprimer certaines valeurs. La plupart des échantillons parasites proviennent du fait que les petites cavités de l’objet ne peuvent pas être reconstruites par un algorithme utilisant l’enveloppe visuelle. La figure 3.9.a montre un exemple de deux échantillons stockés dans un même voxel alors qu’ils touchent réellement deux surfaces orientées différemment et pouvant avoir des propriétés de réflectance totalement différentes.

La teinte moyenne d’un voxel est donc estimée en prenant en compte la cohérence de la teinte entre les échantillons : un histogramme des teintes est construit à partir des échantillons ayant une valeur de teinte  $H$ , le pic de teinte est détecté dans l’histogramme et les échantillons dont leur teinte n’est pas similaire ( $\pm 3\%$ ) à celle du pic sont détruits (voir figure 3.9.b).

La *teinte étendue* d’un voxel est estimée à partir des échantillons restants :

- si plus d’un tiers des échantillons ont la même teinte étendue (blanche, grise ou noire) alors le voxel a également cette même teinte ;
- sinon le voxel a la teinte obtenue par l’histogramme des teintes (celle du pic).

Une interprétation de la surface est effectuée en fonction de la teinte étendue de chaque voxel :

- une teinte noire représente une surface non éclairée : les échantillons de teinte blanche sont supprimés du voxel ;
- une teinte non noire représente une surface éclairée : les échantillons de teinte noire sont supprimés du voxel ;
- une teinte blanche représente une surface de couleur blanche : il y a trop d’échantillons de teinte blanche pour représenter uniquement la specularité ; les échantillons ayant une teinte  $H$  sont supprimés du voxel ;
- une teinte grise représente une surface de couleur grise : il y a trop d’échantillons de teinte grise pour représenter uniquement la specularité ; les échantillons ayant une teinte  $H$  sont supprimés du voxel.

Toutes ces interprétations sont déduites de l’expérimentation sur des images provenant de divers objets.

## B.5.2 Classification des voxels

Lors de la classification des voxels selon la réflectance (section B.1), ce sont les valeurs de *teinte étendue* qui sont réellement utilisées pour grouper les voxels. Par conséquent, trois classes spéciales sont définies afin de grouper les voxels représentant les surfaces blanches, les surfaces grises et les surfaces non éclairées.

La classification des autres voxels, ayant une valeur de teinte  $H$ , est effectuée par analyse d'un histogramme cyclique de teintes (la teinte  $H = 0$  correspond à la teinte  $H = 1$ ). Chaque pic de teinte détecté sur l'histogramme correspond à une classe de voxels. L'algorithme identifie les pics du plus important au plus faible : un pic est défini par une valeur maximale et une baisse de plus de 25% de chaque côté de la valeur maximale (deux pics ne sont jamais regroupés dans une même classe). Lorsque les valeurs restantes dans l'histogramme deviennent trop faibles, il n'y a plus de pic détecté : les derniers groupes sont construits de manière à minimiser le nombre total de groupes et à ne pas grouper des voxels avec des teintes distantes de plus de 0.05 (avec des valeurs de teinte comprises entre 0 et 1). La figure 3.10 montre le résultat de la classification des voxels du clown, objet réel. La détection du pic entouré évite qu'il soit englobé avec le premier pic détecté. À l'inverse, les teintes dans le demi-cercle ne sont pas suffisamment présentes dans l'histogramme et ne peuvent constituer un pic : elles sont donc groupées avec le pic bleu, distant de moins de 0.05.

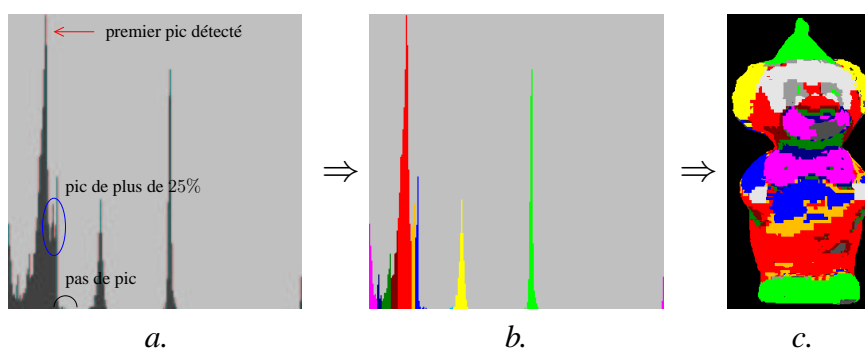


Figure 3.10 – Classification des voxels du clown selon la réflectance ; a. histogramme des teintes de tous les voxels ; b. classification selon la teinte en identifiant les pics dans l'histogramme ; c. visualisation de la classification des voxels sur l'objet reconstruit.

## B.5.3 Détection de sources lumineuses

Lorsqu'un voxel ne dispose pas de suffisamment d'échantillons de luminance, il n'est pas utilisé dans les algorithmes de détection de sources lumineuses. Son faible nombre d'échantillons n'est pas représentatif et son utilisation peut conduire à l'estimation de sources erronées. Le nombre minimal d'échantillons est estimé à 15. Des imprécisions similaires peuvent apparaître lorsque le nombre de voxels dans un groupe est trop faible. Le nombre minimal de voxels dépend du niveau de reconstruction de l'octree, il est situé entre 15 et 60 par groupe de classification selon la normale.

Les trois classes de voxels définies à partir des teintes étendues ne sont pas utilisées pour détecter les sources lumineuses. La classe représentant une surface blanche n'est pas utilisée car

les pixels de sa surface sont souvent sur-exposés sur les photographies : les valeurs de luminance sont donc inutilisables, seules des images à fortes variations de luminance (images HDR [DM97]) permettraient d'acquérir précisément les luminances.

Chacune de nos méthodes de détection d'une source lumineuse travaille sur un groupe restreint de données (données obtenues après classification des voxels selon la normale). Par conséquent une structure de données autre que la représentation compacte dans un octree est utilisée afin d'accélérer les nombreux accès aux valeurs de luminance. Les échantillons du groupe de voxels en cours de traitement sont tous stockés dans un tableau de voxels contenant explicitement la position du voxel, son vecteur normal et des couples de valeurs (luminance  $Y$ , direction d'émission de l'échantillon). Nos méthodes de détection d'une source lumineuse font l'hypothèse que les échantillons de luminance d'un même voxel sont tous émis depuis le centre de ce voxel. Par conséquent, l'angle d'incidence est supposé fixe dans un voxel.

## C Détection d'une source ponctuelle

Pour détecter une source ponctuelle à partir d'un groupe de voxels, la direction de la lumière incidente est estimée en chaque voxel et la position de la source lumineuse est déduite de cet ensemble de directions (figure 3.11). La méthode évalue automatiquement la précision de la source lumineuse détectée sans estimer les propriétés de réflectance de la surface. Lorsque la source lumineuse est détectée loin de la surface de l'objet avec une faible précision, elle est considérée comme une source directionnelle.

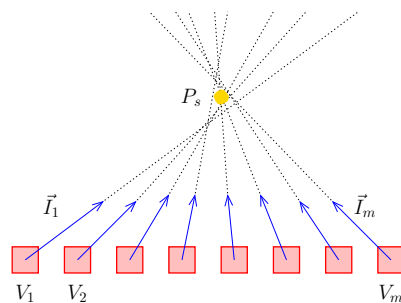


Figure 3.11 – Méthode générale de détection d'une source ponctuelle : estimation de la direction d'incidence pour chaque voxel.

### C.1 Utilisation de la composante diffuse

Cette méthode, utilisant la propriété de réflexion diffuse des surfaces pour détecter une source lumineuse, est appliquée sur un groupe de voxels représentant soit une surface parfaitement diffuse, soit une surface avec réflexion majoritairement diffuse où peu d'échantillons sont disponibles dans la direction de réflexion miroir. Par conséquent la surface représentée par le groupe de voxels est assimilée à une surface lambertienne où la luminance émise par un voxel est constante quelle que soit la direction d'émission (un voxel est considéré comme une surface élémentaire).

La luminance émise par un voxel  $V_j$  est obtenue par une moyenne des échantillons de luminance disponibles pour ce voxel :

$$L(V_j) = L^{moy}(V_j) = \frac{1}{NbL_j} \cdot \sum_{i=1}^{NbL_j} L_{i,j} \quad (3.18)$$

Cette luminance peut être recalculée par une équation d'illumination locale en négligeant les auto-interréflexions lumineuses :

$$\begin{aligned} L(V_j) &= L_r(V_j, \vec{R}) \\ &= f_r(V_j, \vec{R}, \vec{I}_j) \cdot L_i(V_j, \vec{I}_j) \cdot (\vec{N}_j \cdot \vec{I}_j) \cdot \frac{1}{r^2} \\ &= f_r \cdot \frac{L_s}{r^2} \cdot \cos \theta_j \end{aligned} \quad (3.19)$$

avec  $f_r$  la fonction de réflectance constante de la surface lambertienne ;  $L_s$  la luminance constante émise par la source lumineuse ponctuelle ;  $r$  la distance séparant la source du voxel  $V_j$ , supposée constante pour tous les voxels du groupe et  $\theta_j$  l'angle d'incidence du voxel  $V_j$ . Cette équation montre que la luminance émise par un voxel  $V_j$  dépend uniquement de l'orientation de la normale  $\vec{N}_j$  par rapport à la direction d'incidence  $\vec{I}_j$ , et varie linéairement en fonction du cosinus de l'angle d'incidence.

### C.1.1 Méthode principale

#### Direction incidente de référence

L'équation 3.19 admet un maximum pour un angle d'incidence  $\theta = 0$ , correspondant à un voxel dont la normale est parfaitement alignée avec la direction d'incidence. Le voxel du groupe ayant la plus forte valeur de luminance moyenne est recherché et choisi comme voxel de référence  $V_r$  permettant d'initialiser le processus d'estimation de la direction d'incidence sur chaque voxel. La normale de  $V_r$  est supposée être confondue avec la direction d'incidence de ce voxel. Ainsi, une première direction d'incidence est trouvée, notée  $\vec{I}_r$ , et l'équation de luminance en ce voxel peut se réécrire :

$$\begin{aligned} L(V_r) &= f_r \cdot \frac{L_s}{r^2} \cdot \cos \theta_r \\ &= f_r \cdot \frac{L_s}{r^2} \end{aligned} \quad (3.20)$$

#### Direction incidente de chaque voxel

L'estimation du produit de la réflectance de la surface par la luminance de la source, obtenue dans l'équation 3.20, est réutilisée dans l'équation 3.19 afin de déduire une estimation de l'angle d'incidence de tout voxel  $V_j$  du groupe :

$$\theta_j = \arccos \left( \frac{L(V_j)}{L(V_r)} \right) \quad (3.21)$$

L'estimation de l'angle d'incidence  $\theta_j$  ne fournit pas la direction d'incidence  $\vec{I}_j$  mais un cône de directions d'incidence (figure 3.12.a).

La définition d'un plan  $\mathcal{P}$  passant par le centre des voxels  $V_j$ ,  $V_r$  et contenant la direction d'incidence  $\vec{I}_r$  permet de restreindre l'ensemble des directions sur le cône à 2 valeurs de  $\vec{I}_j$

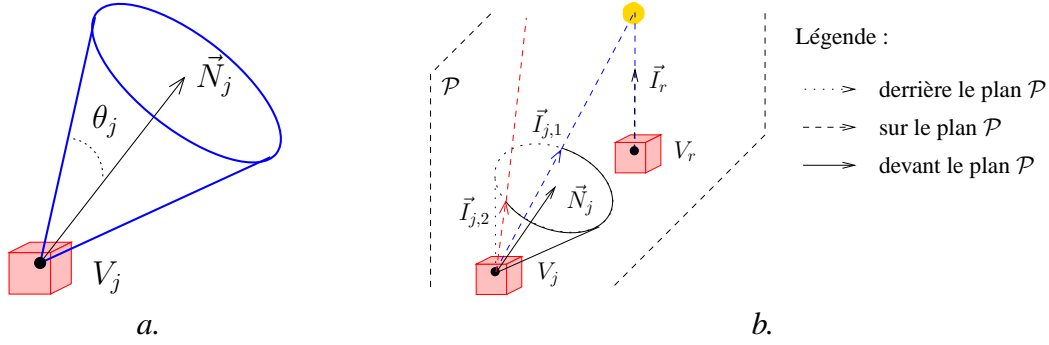


Figure 3.12 – a. La direction d'incidence du voxel  $V_j$  est située sur un cône ; b. l'intersection du cône par un plan  $\mathcal{P}$  passant par les voxels  $V_j$ ,  $V_r$  et contenant  $\vec{I}_r$  permet de réduire l'ensemble des directions d'incidence  $\vec{I}_j$  à deux valeurs.

(figure 3.12.b). L'intersection du cône avec le plan  $\mathcal{P}$  est calculée par résolution d'un système de trois équations à trois inconnues (les coordonnées 3D de  $\vec{I}_j$ ) :

$$\begin{cases} \|\vec{I}_j\| = 1 \\ (\vec{I}_j \cdot \vec{N}_j) = \frac{L(V_j)}{L(V_r)} \\ (\vec{I}_j \cdot \vec{N}_{\mathcal{P}}) = 0 \end{cases} \quad (3.22)$$

En réalité, la résolution de ce système fournit 0, 1 ou 2 solutions correspondant respectivement aux cas où le plan ne coupe pas le cône, le plan est tangent au cône ou le plan coupe franchement le cône comme sur la figure 3.12.b (deux directions d'incidence sont obtenues,  $\vec{I}_{j,1}$  et  $\vec{I}_{j,2}$ ).

Les solutions obtenues ne sont pas toujours valides. Si les directions  $\vec{I}_{j,k}$  et  $\vec{I}_r$  se coupent du côté visible de la surface représentée par les voxels  $V_j$  et  $V_r$ , alors la direction  $\vec{I}_{j,k}$  est valide. Chaque solution peut être testée en résolvant le système :

$$\left\{ P_j + t_{j,k} \cdot \vec{I}_{j,k} = P_r + t_r \cdot \vec{I}_r, \text{ sur les 3 axes du repère.} \right. \quad (3.23)$$

$P_j$  et  $P_r$  sont les coordonnées 3D des voxels  $V_j$  et  $V_r$  ;  $t_{j,k}$  et  $t_r$  sont les scalaires obtenus par la résolution. La direction  $\vec{I}_{j,k}$  est valide uniquement lorsque les deux scalaires sont positifs.

Lorsqu'une seule solution est valide (cas idéal présenté sur la figure 3.13.a), la direction d'incidence du voxel  $V_j$  est correctement définie et l'algorithme passe à l'estimation de la direction d'incidence du voxel suivant. En revanche si aucune solution n'est valide ou si les deux solutions sont valides, la direction d'incidence du voxel  $V_j$  ne peut pas être déterminée. Aucune estimation de la direction d'incidence n'est effectuée pour ce voxel et l'algorithme passe directement au voxel suivant (figures 3.13.b et 3.13.c).

### Résolution matricielle

Lorsque la direction d'incidence  $\vec{I}_j$  du voxel  $V_j$  est correctement définie, la droite d'incidence passant par le centre de  $V_j$  et de vecteur directeur  $\vec{I}_j$  est représentée par l'intersection de deux plans orthogonaux d'équation :

$$\begin{cases} a_1x + b_1y + c_1z + d_1 = 0 \\ a_2x + b_2y + c_2z + d_2 = 0 \end{cases} \quad (3.24)$$



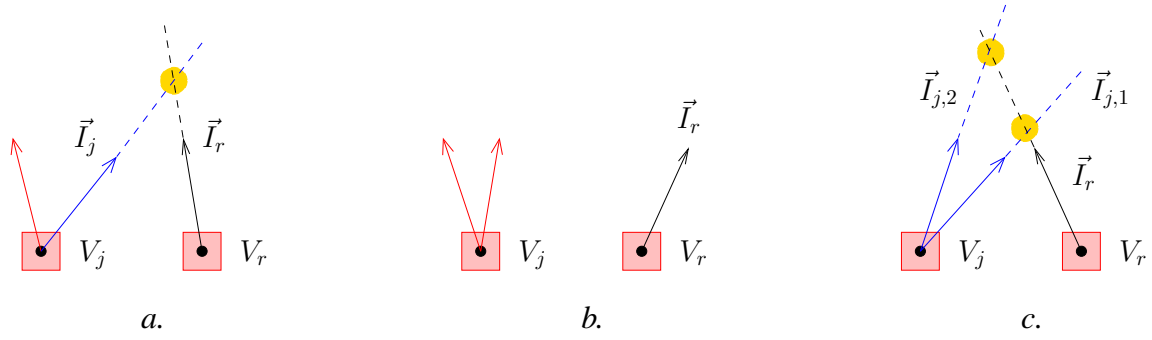


Figure 3.13 – a. Une seule direction d’incidence est valide pour le voxel  $V_j$  : elle représente l’unique solution recherchée ; b. aucune direction d’incidence n’est valide : le voxel  $V_j$  n’est pas utilisé dans la détection de la source lumineuse ; c. les deux directions sont valides : aucun choix ne peut être fait facilement, le voxel  $V_j$  n’est donc pas utilisé.

Les coefficients de ces deux plans sont insérés dans un système matriciel permettant de stocker les directions d’incidence correctement définies sur l’ensemble des voxels d’un même groupe :

$$\begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ \vdots & \vdots & \vdots \\ a_{2m} & b_{2m} & c_{2m} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -d_1 \\ -d_2 \\ \vdots \\ -d_{2m} \end{pmatrix} \quad (3.25)$$

où  $m$  représente le nombre de directions d’incidence stockées. Ce système matriciel, exprimé sous une forme plus compacte  $\mathcal{M}\mathcal{X} = \mathcal{D}$ , permet de définir  $\mathcal{X}$  comme un point appartenant à l’ensemble des directions d’incidence, i.e. comme la position de la source lumineuse recherchée.

Si les  $m$  directions d’incidence se coupent réellement en un point 3D alors la solution est  $\mathcal{X} = \mathcal{M}^{-1}\mathcal{D}$ . La matrice  $\mathcal{M}$  n’est généralement pas inversible (les directions d’incidence ne se coupent pas exactement en un point) mais comme le système matriciel est sur-contraint, il est toujours possible de construire une matrice pseudo-inverse de  $\mathcal{M}$ , appelée  $\mathcal{M}^{pi}$ , telle que  $\mathcal{M}^{pi}\mathcal{M}$  corresponde à la matrice identité [PTVF99]. Par conséquent, la position 3D de la source lumineuse est estimée par l’évaluation :

$$\mathcal{X} = \mathcal{M}^{pi}\mathcal{M}\mathcal{X} = \mathcal{M}^{pi}\mathcal{D} \quad (3.26)$$

Des détails sur la résolution matricielle et le calcul de matrices pseudo-inverses peuvent être obtenus dans le rapport de stage de DEA [Mer02].

### C.1.2 Méthode complémentaire

Nous avons vu précédemment, lors de l’intersection du plan avec le cône, que si les deux directions d’incidence d’un voxel  $V_j$  sont valides (comme sur la figure 3.13.c) alors aucune décision ne peut être prise sur la direction réelle de la source lumineuse. Maintenant qu’une première estimation de la position de la source lumineuse est obtenue, elle est utilisée sur le voxel  $V_j$  afin de déterminer la direction d’incidence correcte parmi les deux possibles.

Ainsi, en traitant tous les voxels dans le même cas de figure que  $V_j$ , nous insérons de nouvelles directions d'incidence dans le système matriciel. Une réévaluation de la matrice  $\mathcal{M}^{pi}$  permet ainsi d'estimer une nouvelle position de source lumineuse.

Cette estimation doit normalement être plus précise que la précédente puisqu'elle utilise un nombre plus conséquent d'informations. Pour vérifier cela, nous évaluons une erreur sur la position de la source estimée :

$$E_d = \frac{1}{m} \cdot \sum_{i=1}^{2m} (\mathcal{M}_i \mathcal{X} - \mathcal{D}_i)^2 \quad (3.27)$$

avec  $m$  le nombre de directions d'incidence insérées dans le système matriciel ;  $\mathcal{M}_i$  la  $i^e$  ligne de la matrice  $\mathcal{M}$  et  $\mathcal{D}_i$  le  $i^e$  scalaire de la matrice  $\mathcal{D}$ .

Cette erreur  $E_d$  correspond à la distance moyenne au carré séparant la source lumineuse de chaque droite d'incidence. L'erreur est évaluée pour chaque système matriciel (avant et après utilisation des directions complémentaires) et nous conservons la source lumineuse estimée avec la plus faible erreur.

L'algorithme 3.1 récapitule la méthode de détection d'une source ponctuelle.

### C.1.3 Méthode itérative

#### Nouvelles directions d'incidence

La qualité de l'estimation de la source lumineuse ponctuelle dépend essentiellement du choix du voxel de référence  $V_r$  puisque tous les calculs de direction d'incidence utilisent sa valeur de luminance  $L(V_r)$  et définissent un plan passant par  $V_r$  et contenant  $\vec{N}_r$ .

L'estimation de la position de la source lumineuse montre que le voxel  $V_r$  n'est pas parfaitement orienté dans la direction d'incidence (généralement  $\vec{N}_r \neq \vec{I}_r$ ). La réévaluation de son angle d'incidence  $\theta_r$  permet de déduire une nouvelle estimation de l'angle d'incidence de tout voxel  $V_j$  (précédemment évaluée par l'équation 3.21) :

$$\theta_j = \arccos \left( \frac{L(V_j)}{L(V_r)} \cdot \cos \theta_r \right) \quad (3.28)$$

La modification de l'équation évaluant  $\theta_j$  entraîne une modification du cône d'incidence de tous les voxels du groupe donc une réévaluation de la direction d'incidence de tous les voxels afin d'obtenir un nouveau système matriciel et d'estimer une nouvelle position de source lumineuse. Nous proposons de vérifier la précision de la nouvelle estimation par une évaluation de l'erreur  $E_d$ . Tant que l'erreur diminue, la nouvelle position est plus précise et peut de nouveau être raffinée en réévaluant la valeur de  $\theta_r$ . Nous avons donc un processus itératif qui converge grâce à la décroissance stricte d'une erreur toujours évaluée positive.

#### Nouvelle direction de référence

Lors des nombreux tests effectués sur des objets virtuels de géométrie connue, nous avons remarqué que la source lumineuse est systématiquement évaluée trop près de l'objet puis est raffinée jusqu'à converger vers une meilleure approximation de sa position réelle. La convergence vers une solution stable peut être lente (plusieurs minutes par groupe de voxels, plusieurs heures pour un objet complet) car la modification de la valeur de  $\theta_r$  est de plus en plus infime.

Pour accélérer les calculs, nous avons retranscrit la courbe de déplacement de la position de source estimée et avons remarqué que cette position varie selon l'axe (barycentre du groupe

```

// Parcours du groupe de voxels
Pour chaque voxel  $V_j$  du groupe faire
    // Moyenne de tous les échantillons de luminance du voxel
    Calculer la luminance  $L(V_j)$  ;
    // Recherche du voxel de référence
    Mettre à jour  $V_r$ , le voxel émettant la plus forte luminance  $L(V_r)$  ;
Fin Pour

// Nouveau parcours du groupe de voxels
Pour chaque voxel  $V_j \neq V_r$  faire
    // Détection des deux directions d'incidence potentielles
    Calculer l'intersection entre le cône de directions et le plan ;
    Vérifier la validité de chaque direction d'incidence obtenue ;
    Si une seule direction est valide alors
        // La direction d'incidence du voxel  $V_j$  est correctement définie
        Insérer la direction dans les matrices  $\mathcal{M}$  et  $\mathcal{D}$  (coefficients de deux plans orthogonaux) ;
    Sinon si deux directions sont valides alors
        // Choix impossible
        Insérer les deux directions dans un tableau temporaire  $T$  ;
    Fin Si
Fin Pour

// Estimation de la position de la source lumineuse ponctuelle
Calculer  $\mathcal{M}^{pi}$  puis  $\mathcal{X}$ , la position de la source lumineuse ;
// Evaluation de l'erreur sur cette estimation de source lumineuse
Calculer l'erreur  $E_d$  sur la source lumineuse détectée ;

// Parcours des voxels  $V_j$  ayant deux directions valides
Pour chaque couple de directions valides extraites de  $T$  faire
    // Evaluation de la direction la plus proche de la source estimée
    Choisir la direction d'incidence en fonction de  $\mathcal{X}$  ;
    // Cette direction correspond à la direction d'incidence du voxel  $V_j$ 
    Insérer cette direction dans les matrices  $\mathcal{M}$  et  $\mathcal{D}$  ;
Fin Pour

// Nouvelle estimation de la position de la source lumineuse ponctuelle
Recalculer  $\mathcal{M}^{pi}$  puis  $\mathcal{X}$ , la nouvelle position de la source lumineuse ;
// Evaluation d'une nouvelle erreur sur cette estimation de source et comparaison avec l'ancienne valeur
Calculer la nouvelle erreur  $E_d$  et garder la position de source lumineuse ayant l'erreur la plus faible ;

```

Algorithme 3.1 – Algorithme d'estimation d'une source ponctuelle appliqué sur un groupe de voxels représentant une surface diffuse

de voxels - position réelle de la source). Par conséquent en choisissant un nouveau voxel de référence  $V_r$  positionné sur l'axe (barycentre du groupe de voxels - position estimée de la source), nous avons une estimation précise de son angle d'incidence  $\theta_r$  et nous sommes assurés que la nouvelle estimation de la position de la source fait peu varier l'angle  $\theta_r$  (voir figure 3.14). Le choix d'un nouveau voxel de référence permet une convergence beaucoup plus rapide de l'algorithme d'estimation de source : 4 ou 5 itérations suffisent à obtenir une estimation stable.

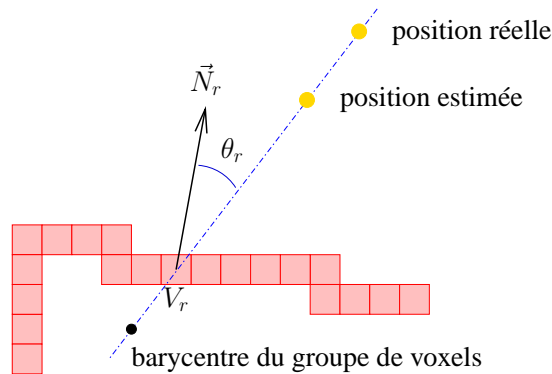


Figure 3.14 – Choix d'un nouveau voxel de référence  $V_r$  à partir du barycentre de voxels et de l'estimation de source ponctuelle ; son angle d'incidence  $\theta_r$  est évalué avec précision.

## C.2 Utilisation de la composante spéculaire

Nous proposons une autre méthode d'estimation d'une source ponctuelle, basée sur l'utilisation de la propriété de réflexion spéculaire des surfaces. Cette méthode est appliquée à un groupe de voxels pour lesquels nous disposons d'échantillons de luminance variable : les échantillons de plus forte luminance sont assimilés à des échantillons orientés dans la direction de réflexion spéculaire. Par conséquent la surface représentée par le groupe de voxels est soit d'aspect diffus directionnel (avec un large lobe spéculaire) soit fortement spéculaire (avec un lobe spéculaire plus étroit). La surface peut également réfléchir de manière diffuse une partie de son énergie reçue (figure 3.15.a).

### C.2.1 Méthode naïve

Pour détecter une source ponctuelle, cette méthode utilise le même principe général que celui décrit dans la section précédente, à savoir : (1) estimation de la direction d'incidence en chaque voxel, (2) représentation de l'ensemble des directions obtenues par un système matriciel puis (3) évaluation d'une matrice pseudo-inverse afin d'estimer la position de la source lumineuse.

Comme une luminance incidente à une surface spéculaire est supposée réfléchiée dans la direction miroir  $\vec{M}$  par rapport à la normale, l'estimation de la direction miroir en chaque voxel permet de déduire la direction d'incidence (figure 3.15.a).

La direction miroir peut être déduite à partir des informations de luminance d'un voxel puisqu'elle correspond à la direction pour laquelle la luminance réfléchiée est maximale. Un parcours de la liste des échantillons permet de récupérer l'échantillon  $\vec{R}_m$  de plus forte luminance, considéré dans la direction de réflexion miroir. L'évaluation de la direction miroir de  $\vec{R}_m$  permet d'obtenir la direction d'incidence  $\vec{I}$  (figure 3.15.b).

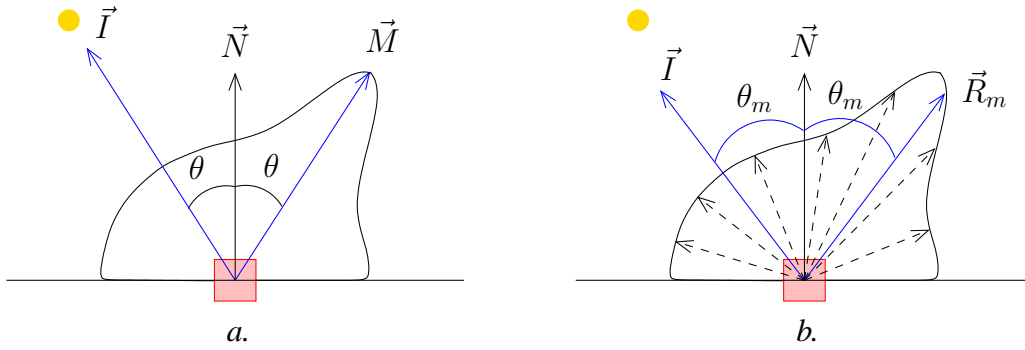


Figure 3.15 – a. Surface ayant des propriétés de réflexion spéculaire : l'essentiel de l'énergie reçue est réfléchié vers la direction de réflexion miroir  $\vec{M}$  ; b. ensemble des échantillons de luminance disponibles pour un voxel : l'échantillon de plus forte luminance  $\vec{R}_m$  est le plus proche de la direction de réflexion miroir.

## C.2.2 Méthode optimale

### Principe

La précédente méthode donne de bons résultats uniquement lorsque l'échantillonnage des luminances est dense, ce qui n'est pas le cas la plupart du temps. Pour définir précisément la direction de réflexion spéculaire, une solution plus efficace consiste à utiliser toutes les informations contenues dans le lobe spéculaire. Pour cela, nous définissons le lobe spéculaire par l'ensemble des échantillons proches de l'échantillon de luminance maximale  $\vec{R}_m$ , avec un angle inférieur à  $20^\circ$  (figure 3.16.a).

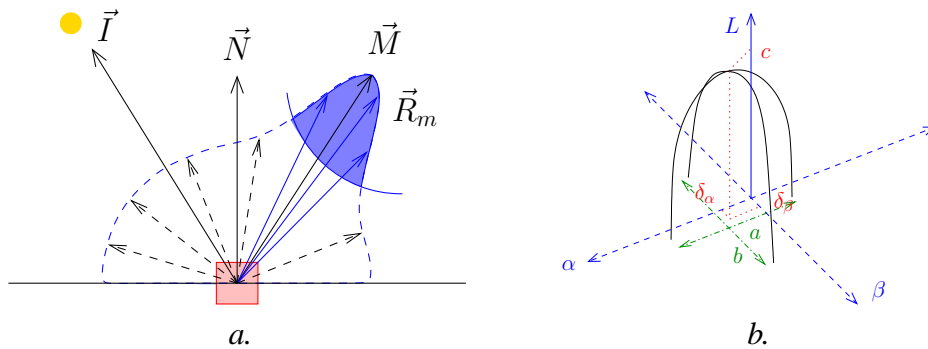


Figure 3.16 – a. Les échantillons autour de  $\vec{R}_m$  sont utilisés pour définir le lobe spéculaire ; b. le lobe spéculaire est estimé par une fonction paramétrique :  $(\delta_\alpha, \delta_\beta)$  est la direction de réflexion miroir estimée et  $c$  sa luminance émise.

La luminance émise dans les directions du lobe spéculaire est ensuite modélisée par une surface parabolique d'équation :

$$L(\alpha, \beta) = a. (\alpha - \delta_\alpha)^2 + b. (\beta - \delta_\beta)^2 + c \quad (3.29)$$

où  $(\alpha, \beta)$  représente une direction de réflexion lumineuse dans le système de coordonnées sphériques.  $a$ ,  $b$ ,  $c$ ,  $\delta_\alpha$  et  $\delta_\beta$  sont des coefficients à déterminer :  $(\delta_\alpha, \delta_\beta)$  représente la direction de réflexion miroir que nous cherchons à estimer ;  $c$  est la luminance émise dans cette direction

miroir ;  $a$  et  $b$  sont les coefficients définissant la courbure de la parabole sur les deux coordonnées sphériques  $\alpha$  et  $\beta$  ; comme la surface est supposée être isotrope, le modèle est simplifié en définissant  $a = b$  (voir figure 3.16.b pour une représentation visuelle des coefficients).

Les coefficients de cette équation sont identifiés en utilisant les échantillons de luminance contenus dans le lobe spéculaire. Une méthode de régression linéaire permet de minimiser une fonction d'erreur correspondant à la différence au carré entre les luminances des échantillons réels et les luminances recalculées par l'équation 3.29 :

$$E_l(V_j) = \sum_{i=1}^{NbL\_Lobe_j} (a. (\alpha_i - \delta_\alpha)^2 + b. (\beta_i - \delta_\beta)^2 + c - L_{i,j})^2 \quad (3.30)$$

avec  $NbL\_Lobe_j$  le nombre d'échantillons de luminance contenus dans le lobe spéculaire du voxel  $V_j$  ;  $(\alpha_i, \beta_i)$  les coordonnées sphériques du  $i^e$  échantillon de luminance utilisé et  $L_{i,j}$  la luminance réelle du  $i^e$  échantillon.

L'échantillonnage des luminances doit être suffisamment dense dans le lobe spéculaire pour permettre une estimation correcte de la direction miroir  $M$ . En deçà de 10 échantillons de luminance, les coefficients de la surface parabolique ne sont pas estimés et le voxel  $V_j$  n'est pas utilisé dans la détection de la source lumineuse. De même lorsque l'échantillon de luminance maximale  $\vec{R}_m$  est orienté à plus de  $70^\circ$  par rapport à la normale, aucune estimation n'est effectuée pour ce voxel : non seulement le lobe spéculaire n'est pas complet donc difficile à modéliser mais il correspond à une direction d'incidence rasante, peu fiable. En effet sur les surfaces réelles, la direction émettant la plus forte luminance n'est pas toujours située dans la direction miroir pour des angles d'incidence rasants : un léger biais est généralement constaté.

### Mise en œuvre

Nous avons choisi de modéliser le lobe spéculaire par une surface parabolique car sa représentation est plus intuitive qu'une fonction en cosinus, bien que celle-ci soit plus proche du modèle de Phong modifié. En fait, il n'y a pas de différence notable entre ces deux types de modèles, comme le montre la figure 3.17.

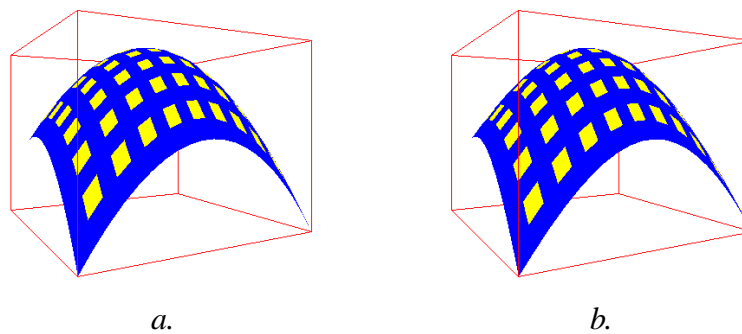


Figure 3.17 – La luminance est calculée en fonction des coordonnées sphériques, comme dans 3.16.b ; a. surface parabolique d'équation  $L(\alpha, \beta) = -\alpha^2 - \beta^2 + 2$  ; b. surface sinusoidale d'équation  $L(\alpha, \beta) = \cos(3\alpha/2) + \cos(3\beta/2)$ .

Les coefficients de la surface parabolique sont identifiés par application de la méthode de Levenberg - Marquardt [PTVF99] : elle minimise la fonction d'erreur  $E_l(a, c, \delta_\alpha, \delta_\beta)$  par une

descente de gradient linéaire. Cette méthode ne peut être appliquée sans une première estimation des 4 coefficients. Nous proposons donc d'utiliser  $\vec{R}_m$  comme première estimation de la direction de réflexion miroir du voxel  $V_j$  :  $(\delta_\alpha, \delta_\beta)$  sont estimés par  $(\alpha_m, \beta_m)$  et  $c$  par  $L_{m,j}$ .  $a$  est ensuite estimé en utilisant plusieurs échantillons de luminance contenus dans le lobe spéculaire.

La méthode utilise l'estimation des 4 coefficients pour calculer la dérivée partielle de  $E_l$  selon chaque coefficient. La valeur absolue de la dérivée partielle selon un coefficient représente la pente de la courbe d'erreur selon ce coefficient et le signe de la dérivée partielle représente le sens de cette pente. Par conséquent, le coefficient dont la valeur absolue de la dérivée partielle est la plus importante est le coefficient modifié en priorité : la modification de ce coefficient (plutôt qu'un autre) assure une plus forte diminution de l'erreur  $E_l$ . Lorsqu'un coefficient est modifié dans le sens contraire de ses précédentes modifications, son pas de déplacement est divisé par deux afin d'assurer la convergence de l'algorithme. Celui-ci est terminé lorsque le pas de déplacement d'un coefficient ou la diminution de l'erreur  $E_l$  après le déplacement d'un coefficient est inférieur à un seuil fixé.

## D Détection d'une source directionnelle

La détection d'une source directionnelle à partir d'un groupe de voxels suppose que la direction de la lumière incidence  $\vec{I}_s$  est identique pour tous les voxels (figure 3.18).

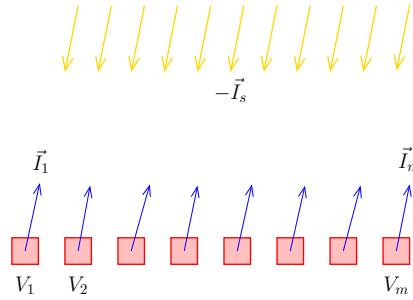


Figure 3.18 – La direction d'incidence est identique pour tous les voxels.

### D.1 Utilisation de la composante diffuse

Comme dans la section C.1, cette méthode est utilisée sur un groupe de voxels représentant une surface lambertienne où la luminance émise par un voxel est considérée constante. Cette luminance est donc obtenue par une moyenne des échantillons de luminance du voxel ( $L(V_j) = L^{moy}(V_j)$ , voir équation 3.18) et peut être réécrite en reprenant les notations de l'équation 3.19 :

$$\begin{aligned}
 L(V_j) &= f_r \cdot L_s \cdot \cos \theta_j \\
 &= f_r \cdot L_s \cdot (\vec{N}_j \cdot \vec{I}_s) \\
 &= f_r \cdot L_s \cdot (N_{j,x} \cdot I_{s,x} + N_{j,y} \cdot I_{s,y} + N_{j,z} \cdot I_{s,z})
 \end{aligned} \tag{3.31}$$

avec  $f_r$  la fonction de réflectance constante de la surface lambertienne ;  $L_s$  la luminance constante émise par la source lumineuse  $s$  et  $\vec{I}_s$  la direction d'incidence constante pour tous les voxels. Cette



équation montre que la luminance émise par un voxel  $V_j$  dépend uniquement de l'orientation de sa normale  $\vec{N}_j$ .

L'équation 3.31 peut être insérée dans un système matriciel pour l'ensemble des  $m$  voxels du groupe :

$$\begin{pmatrix} N_{1,x} & N_{1,y} & N_{1,z} \\ N_{2,x} & N_{2,y} & N_{2,z} \\ \vdots & \vdots & \vdots \\ N_{m,x} & N_{m,y} & N_{m,z} \end{pmatrix} \cdot \begin{pmatrix} f_r \cdot L_s \cdot I_{s,x} \\ f_r \cdot L_s \cdot I_{s,y} \\ f_r \cdot L_s \cdot I_{s,z} \end{pmatrix} = \begin{pmatrix} L(V_1) \\ L(V_2) \\ \vdots \\ L(V_m) \end{pmatrix} \quad (3.32)$$

Ce système est de nouveau de la forme  $\mathcal{M}\mathcal{X} = \mathcal{D}$  avec  $\mathcal{M}$  et  $\mathcal{D}$  des matrices connues.  $\mathcal{X}$  représente  $f_r \cdot L_s \cdot \vec{I}_s$  avec  $\vec{I}_s$ , la direction d'incidence recherchée, pouvant être obtenue en normant  $\mathcal{X}$ .

La matrice  $\mathcal{M}$  n'est généralement pas inversible mais comme le système est sur-contraint, nous pouvons également construire la matrice pseudo-inverse de  $\mathcal{M}$ , notée  $\mathcal{M}^{pi}$ , et obtenir une estimation de la direction d'incidence :

$$\mathcal{X} = \mathcal{M}^{pi} \mathcal{D} \quad (3.33)$$

## D.2 Utilisation de la composante spéculaire

Comme dans la section C.2, cette méthode est utilisée sur un groupe de voxels représentant une surface d'aspect diffus directionnel ou fortement spéculaire, avec une composante de réflexion diffuse non nécessairement nulle.

La méthode utilisée pour détecter la source directionnelle est similaire à celle utilisée pour détecter une source ponctuelle, i.e. la direction de réflexion miroir est détectée sur chaque voxel puis la direction d'incidence est évaluée par symétrie par rapport à la normale. La direction de réflexion miroir peut être obtenue soit à partir de l'échantillon de plus forte luminance (voir section C.2.1 et figure 3.15.b), soit en estimant le lobe spéculaire (voir section C.2.2 et figure 3.16.a).

Lorsqu'une estimation de la direction d'incidence est obtenue pour chaque voxel, la direction d'incidence commune  $\vec{I}_s$  est évaluée par une moyenne sur tous les voxels :

$$\vec{I}_s = \frac{1}{NbV} \cdot \sum_{j=1}^{NbV} (\vec{I}_j) \quad (3.34)$$

avec  $NbV$  le nombre de voxels du groupe et  $\vec{I}_j$  la direction d'incidence estimée pour le voxel  $V_j$ .

## E Estimation conjointe : sources lumineuses / propriétés de réflectance

Chaque méthode de détection de source lumineuse est indépendamment appliquée sur chaque groupe de voxels afin de fournir soit une source lumineuse ponctuelle, soit une source lumineuse directionnelle.

Pour chaque groupe de voxels et chaque source lumineuse détectée, les propriétés de réflectance de la surface sont estimées en utilisant une représentation paramétrique de sa BRDF (modèle de Phong modifié) :

$$f_r(P, \vec{R}, \vec{I}) = K_d \cdot \frac{1}{\pi} + K_s \cdot \frac{n+2}{2\pi} \cdot \cos^n \phi \quad (3.35)$$

## E.1 Surface d'aspect diffus

Lorsque la source est détectée par une méthode utilisant les propriétés de réflexion diffuse de la surface, le modèle de Lambert est utilisé et la luminance recalculée à partir de ce modèle est exprimée par :

$$\begin{aligned} L(V_j) &= f_r \cdot L_s \cdot \cos \theta_j \cdot \frac{1}{r_j^2} \\ &= L_s \cdot K_d \cdot \frac{1}{\pi \cdot r_j^2} \cdot \cos \theta_j \end{aligned} \quad (3.36)$$

avec  $K_d$  le seul coefficient à estimer, constant sur toute la surface ;  $L_s$  la luminance constante émise par la source lumineuse détectée ;  $\theta_j$  l'angle d'incidence du voxel  $V_j$  et  $r_j$  la distance séparant la source du voxel  $V_j$ . Cette distance  $r_j$  est utilisée uniquement dans le cas de la détection d'une source ponctuelle et peut être évaluée pour chaque voxel grâce à l'estimation de la position de la source.

Pour recalculer la luminance à partir de cette équation, nous avons besoin de connaître la valeur des coefficients  $L_s$  et  $K_d$ . Comme ces deux coefficients dépendent linéairement l'un de l'autre, aucune distinction ne peut être faite entre leurs valeurs : nous estimons donc le produit  $L_s \cdot K_d$ . Ce produit est estimé par minimisation d'une fonction d'erreur représentant la différence au carré entre la luminance des échantillons et la luminance recalculée par le modèle :

$$E(L_s \cdot K_d) = \sum_{j=1}^{NbV} \sum_{i=1}^{NbL_j} \left( L_s \cdot K_d \cdot \frac{1}{\pi \cdot r_j^2} \cdot \cos \theta_j - L_{i,j} \right)^2 \quad (3.37)$$

Le minimum de cette fonction est obtenu en suivant le signe de sa dérivée, donnant le sens de la pente de  $E$ . Nous obtenons ainsi la meilleure estimation du produit  $L_s \cdot K_d$  et une valeur d'erreur  $E(L_s \cdot K_d)$  sur cette estimation. La valeur d'erreur obtenue est comparée entre les deux estimations de  $L_s \cdot K_d$  (celle utilisant la source ponctuelle et celle utilisant la source directionnelle) et l'estimation ayant la plus faible erreur est conservée comme seul résultat de cette estimation conjointe :  $L_s \cdot K_d$ ,  $E(L_s \cdot K_d)$  et la source lumineuse correspondante sont conservés.

## E.2 Surface d'aspect diffus directionnel

Lorsque la source est détectée par une méthode utilisant les propriétés de réflexion spéculaire de la surface, le modèle de Phong modifié est utilisé (équation 3.35) et la luminance recalculée à partir de ce modèle est exprimée par :

$$\begin{aligned} L(V_j, \vec{R}) &= f_r(V_j, \vec{R}, \vec{I}_j) \cdot L_s \cdot \cos \theta_j \cdot \frac{1}{r_j^2} \\ &= L_s \cdot \left( K_d \cdot \frac{1}{\pi \cdot r_j^2} + K_s \cdot \frac{n+2}{2\pi \cdot r_j^2} \cdot \cos^n \phi_{R,j} \right) \cdot \cos \theta_j \\ &= L_s \cdot K_d \cdot \frac{1}{\pi \cdot r_j^2} \cdot \cos \theta_j + L_s \cdot K_s \cdot \frac{n+2}{2\pi \cdot r_j^2} \cdot \cos^n \phi_{R,j} \cdot \cos \theta_j \end{aligned} \quad (3.38)$$

avec  $\vec{R}$  la direction d'émission de la luminance et  $\phi_{R,j}$  l'angle entre la direction de réflexion miroir du voxel  $V_j$  et la direction  $\vec{R}$ .

Pour recalculer la luminance à partir de cette équation, nous devons connaître la valeur des coefficients  $L_s$ ,  $K_d$ ,  $K_s$  et  $n$ . Comme précédemment, la luminance de la source lumineuse ne peut être séparée des coefficients  $K_d$  et  $K_s$ . Une estimation des coefficients est obtenue par minimisation d'une fonction d'erreur représentant la différence au carré entre la luminance des échantillons et la luminance recalculée par le modèle de Phong modifié :

$$E(L_s.K_d, L_s.K_s, n) = \sum_{j=1}^{NbV} \sum_{i=1}^{NbL_j} \left( L_s.K_d \cdot \frac{1}{\pi \cdot r_j^2} \cdot \cos \theta_j + L_s.K_s \cdot \frac{n+2}{2\pi \cdot r_j^2} \cdot \cos^n \phi_{i,j} \cdot \cos \theta_j - L_{i,j} \right)^2 \quad (3.39)$$

Le minimum de cette fonction d'erreur est obtenu en appliquant une méthode de régression non-linéaire similaire à la méthode de Levenberg/Marquardt présentée à la section [C.2.2](#). Les dérivées partielles selon chaque coefficient sont utilisées :

$$\begin{cases} E &= \sum_{j=1}^{NbV} \sum_{i=1}^{NbL_j} (y)^2 \\ \frac{\partial E}{\partial L_s.K_d} &= \frac{2}{\pi} \cdot \sum_{j=1}^{NbV} \sum_{i=1}^{NbL_j} \left( y \cdot \frac{\cos \theta_j}{r_j^2} \right) \\ \frac{\partial E}{\partial L_s.K_s} &= \frac{n+2}{\pi} \cdot \sum_{j=1}^{NbV} \sum_{i=1}^{NbL_j} \left( y \cdot \frac{\cos \theta_j}{r_j^2} \cdot \cos^n \phi_{i,j} \right) \\ \frac{\partial E}{\partial n} &= \frac{L_s.K_s}{\pi} \cdot \sum_{j=1}^{NbV} \sum_{i=1}^{NbL_j} \left( y \cdot (1 + (n+2) \cdot \ln(\cos \phi_{i,j})) \cdot \frac{\cos \theta_j}{r_j^2} \cdot \cos^n \phi_{i,j} \right) \end{cases} \quad (3.40)$$

$n$  est un coefficient utilisé dans la partie exponentielle du calcul : il a un rôle différent par rapport aux deux autres coefficients  $L_s.K_d$  et  $L_s.K_s$ . Par conséquent, nous modifions l'algorithme de Levenberg - Marquardt afin de s'assurer que le coefficient  $n$  est correctement mis à jour : comme la valeur absolue de la dérivée partielle selon  $n$  est généralement beaucoup plus faible que les autres dérivées partielles, le coefficient  $n$  est obligatoirement modifié après plusieurs modifications successives des coefficients  $L_s.K_d$  et  $L_s.K_s$  : un raffinement similaire sur le coefficient de rugosité du modèle simplifié de Torrance-Sparrow avait été proposé par K. Hara et al. [\[HNI05\]](#).

Nous obtenons ainsi une estimation correcte des coefficients du modèle de Phong modifié à une constante près, la luminance de la source lumineuse. Comme pour la surface diffuse, la valeur d'erreur est comparée entre les deux estimations (celle utilisant la source ponctuelle et celle utilisant la source directionnelle) et l'estimation ayant la plus faible erreur est conservée comme seul résultat de cette estimation conjointe.

### E.3 Validation de l'estimation conjointe

Lorsqu'un groupe de voxels n'est pas clairement défini comme une surface lambertienne ou comme une surface disposant de suffisamment d'échantillons de luminance dans le lobe spéculaire, les quatre méthodes de détection de source sont utilisées successivement. Par conséquent, un tel groupe de voxels dispose de deux estimations de propriétés de réflectance, l'une représentant une surface diffuse, l'autre représentant une surface d'aspect diffus directionnel avec, pour chaque estimation, une valeur d'erreur et une estimation de source lumineuse. L'estimation ayant la plus faible erreur est de nouveau choisie afin de conserver une seule estimation de paramètres par groupe de voxels.

Par conséquent, chaque groupe de voxels dispose à présent d'une estimation de source lumineuse (ponctuelle ou directionnelle), d'une estimation de BRDF (lambertienne ou diffuse/spéculaire) et d'une évaluation d'erreur entre les échantillons réels et les échantillons recalculés à partir de la BRDF et de la source détectée. L'obligation de détecter une source par groupe de voxels entraîne la détection de beaucoup de sources lumineuses dont certaines sont erronées (par exemple lorsque des groupes de voxels sont réellement éclairés par plusieurs sources lumineuses alors qu'une seule n'est détectée).

L'erreur évaluée entre les luminances des échantillons et les luminances recalculées est utilisée pour retrouver les sources lumineuses mal estimées. Les valeurs d'erreur doivent être normalisées entre les différents groupes de voxels en divisant chaque erreur par le nombre d'échantillons de luminance disponibles dans le groupe de voxels. Ainsi, les valeurs d'erreur de tous les groupes de voxels sont comparées et seules les estimations ayant une faible erreur sont conservées. Les autres estimations, ayant une erreur plus importante, sont supprimées car elles sont considérées comme non pertinentes.

Nous appliquons ensuite une mise en correspondance de l'ensemble des sources lumineuses restantes afin de rassembler les sources identiques : il faut moins de  $10^\circ$  entre deux sources directionnelles ou moins de 20% de la distance source-objet entre deux sources ponctuelles pour qu'elles puissent être rassemblées. La liste de sources lumineuses obtenue est ainsi réduite au minimum et elle correspond généralement au résultat recherché.

Enfin, nous effectuons une dernière mise en correspondance entre les propriétés de réflectance des groupes de voxels appartenant à une même classe de réflectance (voxels de même teinte). Si certains groupes affichent une réflectance diffuse alors que d'autres affichent une réflectance spéculaire, nous estimons une réflectance spéculaire sur l'ensemble de la classe de même teinte afin de ne pas perdre l'information de spécularité sur cette surface. Généralement, une réflectance estimée diffuse correspond à un groupe de voxels pour lesquels aucune image n'est disponible dans la direction de réflexion miroir.

La mise en correspondance entre les voxels de même teinte permet également d'obtenir une estimation de la luminance émise par une source lumineuse par rapport à la luminance émise par une autre source. En effet, lorsque deux groupes de voxels dont l'un est éclairé par une source  $s_1$  et l'autre par une source  $s_2$  appartiennent à une même classe de réflectance, alors le rapport de l'estimation du produit  $L_{s_1} \cdot K_d$  par  $L_{s_2} \cdot K_d$  donne une estimation de  $L_{s_1}/L_{s_2}$ . Nous ne pouvons obtenir plus de précision sur l'estimation de la luminance des sources lumineuses détectées.

## E.4 Résultats

Les résultats de l'estimation conjointe des caractéristiques des sources lumineuses et des propriétés de réflectance des surfaces ont été publiés dans la conférence internationale ICCVG en 2004 [MM04]. Des résultats plus récents sur l'estimation des réflectances d'objets réels et sur un premier système de rééclairage ont été publiés dans la revue internationale IJCV en 2006 [MMF06].

### E.4.1 Objets de géométrie connue

#### Sans reconstruction géométrique

Pour valider notre méthode d'estimation conjointe des sources lumineuses et des propriétés de réflectance, nous avons effectué des premiers tests avec une géométrie connue : ils sont donc indépendants de la reconstruction géométrique de l'objet.

Nous avons généré un ensemble de 256 voxels éclairés par une seule source lumineuse et placés aléatoirement sur un disque d'un mètre de diamètre. Les directions des échantillons de luminance et la normale sont également générés aléatoirement sur chaque voxel, à partir de la connaissance des caractéristiques de la source lumineuse et des propriétés de réflectance de la surface ( $L_s$ ,  $K_d$ ,  $K_s$  et  $n$ ). Les différents algorithmes de détection de source lumineuse sont ensuite appliqués en utilisant uniquement les informations géométriques de la surface et les 60 échantillons de luminance par voxel. Enfin l'estimation conjointe est appliquée afin de retrouver les propriétés de réflectance et vérifier la validité des sources lumineuses détectées.

Le tableau 3.1 récapitule les résultats de l'analyse effectuée. Nous pouvons remarquer que

Surface d'aspect	Distance réelle objet-source	Source estimée	Erreur sur		
			les caractéristiques de la source	$L_s.K_d$ $L_s.K_s$	$n$
<i>diffus</i>	1m	<i>ponctuelle</i>	< 1mm	0.5%	×
	9m		1cm	1%	
	> 9m	<i>directionnelle</i>	< 1°		
<i>diffus directionnel</i>	30cm	<i>ponctuelle</i>	< 1cm	0.5%	3%
	2m		15cm	1%	5%
	6m	<i>aucune</i>	1m	25%	40%
	> 6m	<i>directionnelle</i>	< 1°	2%	8%

Tableau 3.1 – Détection d'une source lumineuse ponctuelle pour un objet d'un mètre de diamètre.

l'algorithme de détection de source ponctuelle est plus précis pour des surfaces diffuses : l'algorithme itératif permet de raffiner la position de la source détectée. Les propriétés de réflectance sont toujours estimées avec précision et l'algorithme détecte automatiquement une source directionnelle lorsque la source ponctuelle est trop loin de l'objet (distance supérieure à 9 mètres dans le cas d'une surface lambertienne).

De plus lors de la détection d'une source ponctuelle, l'estimation de l'erreur  $E_d$  (voir équation 3.27) permet à l'algorithme de rejeter les sources estimées trop imprécisément : c'est le cas de la source placée à 6 mètres d'une surface spéculaire.

#### Avec reconstruction géométrique

Nous utilisons à présent un lumigraphie complet représentant une sphère dont les images de résolution  $256 \times 256$  sont générées par lancer de rayon avec le modèle de Phong modifié : ainsi les propriétés de réflectance estimées sont facilement comparables aux propriétés de la surface originale. La sphère, d'un mètre de diamètre, est éclairée par trois sources lumineuses : une source ponctuelle située à 35 centimètres de l'objet et deux sources directionnelles.

Pour appliquer les algorithmes de détection de source lumineuse, nous avons reconstruit la géométrie avec un octree de profondeur 7 et des normales lissées sur une distance 3. Le tableau

3.2 montre le résultat de l'estimation. Malgré la sensibilité de nos algorithmes de détection à la

Surface	Distance réelle objet-source	Erreur sur			
		la position de la source	la direction des 2 sources	$L_s.K_d$ $L_s.K_s$	$n$
<i>diffuse et/ou spéculaire</i>	<i>35cm</i>	<i>&lt; 4mm</i>	<i>&lt; 1°</i>	<i>&lt; 10%</i>	<i>&lt; 20%</i>

Tableau 3.2 – Détection de plusieurs sources lumineuses à partir d'une sphère d'un mètre de diamètre ; l'objet est reconstruit avec un octree de profondeur 7.

précision du modèle géométrique, notre méthode montre qu'elle est robuste : plusieurs sources lumineuses sont retrouvées avec une imprécision inférieure au degré alors que la normale estimée sur le modèle géométrique a une imprécision moyenne de  $1.3^\circ$  (voir tableau 2.3 du chapitre II). Les trois sources lumineuses sont retrouvées sans exception avec diverses valeurs de réflectance sur la surface. En revanche, les imperfections du modèle géométrique nuisent plus à l'estimation de la BRDF de la surface, avec une imprécision jusqu'à 20% sur  $n$ , l'exposant de réflexion spéculaire. Néanmoins, cette variation n'est pas si importante car elle est visible uniquement sur la taille du halo spéculaire.

### E.4.2 Objets synthétiques de réflectance plus complexe

De multiples lumigraphes synthétiques sont utilisés pour tester nos algorithmes de détection de sources lumineuses. Ce sont des objets de géométrie plus complexe qu'auparavant, composés de plusieurs types de surface. Les images sont toutes de résolution  $256 \times 256$  et l'octree est reconstruit à la profondeur 7. Les images sont générées avec le logiciel PovRay afin de définir des propriétés de réflectance variées et différentes du modèle de Phong modifié. Sur certains objets, des textures ou des cartes de normales (bump mapping) sont définies. La figure 3.19

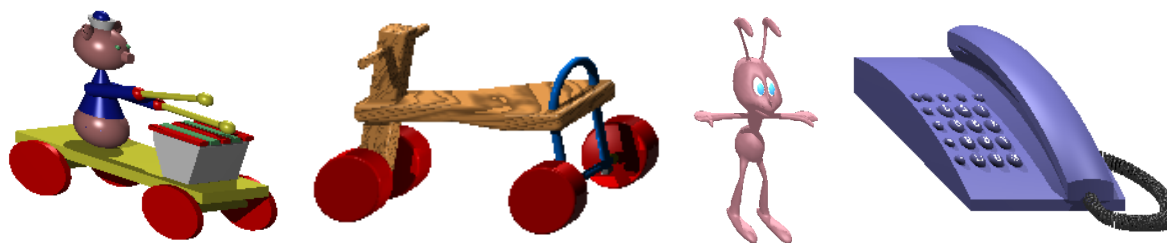


Figure 3.19 – Lumigraphes synthétiques générés avec PovRay, éclairés par deux ou trois sources lumineuses.

montre un échantillon des lumigraphes utilisés. Ils sont tous éclairés par deux ou trois sources lumineuses, dont une ponctuelle placée à 35cm de la surface la plus proche.

Le tableau 3.3 montre le résultat de la détection de sources lumineuses pour des objets éclairés par trois sources comme dans le tableau 3.2. Les résultats sont acceptables bien que certaines surfaces soient texturées ou certaines arêtes chanfreinées. Les sources directionnelles sont détectées avec une imprécision de  $5^\circ$  mais il arrive quelquefois qu'une source directionnelle supplémentaire soit détectée alors qu'elle ne correspond à aucune source lumineuse réelle. Cette détection incohérente provient généralement d'un petit groupe de voxels contenant de fortes



Surface	Distance réelle objet-source	Erreur sur		Nombre de sources suppl. incohérentes
		la position de la source	la direction des 2 sources	
<i>diffuse et/ou spéculaire</i>	<i>35cm</i>	<i>&lt; 5cm</i>	<i>&lt; 5°</i>	<i>1 source directionnelle</i>

Tableau 3.3 – Détection de plusieurs sources lumineuses à partir d’objets virtuels d’un mètre de diamètre ; l’objet est reconstruit avec un octree de profondeur 7.

imprécisions sur leur ensemble d’échantillons de luminance. En revanche, les propriétés de réflectance ne sont pas toujours estimées correctement : les valeurs des coefficients peuvent être jusqu’à deux fois trop élevés ou deux fois trop faibles par rapport à leur valeur originale.

### E.4.3 Objets réels avec petites cavités

Les images calibrées des objets réels proviennent du système d’acquisition décrit à la section [A.2.2](#) du chapitre II. Nous avons utilisé deux objets réels de réflectance différente. Le clown est un objet en plastique, composé de plusieurs surfaces de réflectance uniforme, avec des propriétés de réflexion spéculaire tandis que le tournesol est un objet en bois composé de deux textures de réflectance plus diffuse (voir figure [3.20](#)).

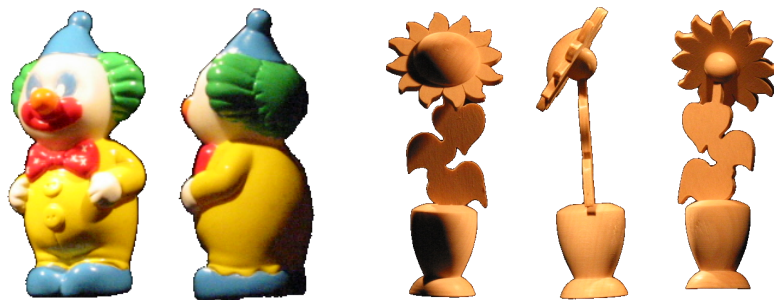


Figure 3.20 – Objets réels acquis dans notre laboratoire, éclairés par deux ampoules lumineuses ; le clown mesure 8cm et le tournesol 12cm.

Les images du clown sont de résolution  $256 \times 256$  et l’objet est reconstruit dans un octree de profondeur 7 avec un lissage des normales de distance 3. Les images du tournesol ont une résolution de  $512 \times 512$  et l’objet est reconstruit dans un octree de profondeur 8 avec un lissage des normales de distance 5. Une seule série de 72 images est utilisée pour reconstruire la géométrie des deux objets mais toutes les images sont utilisées pour détecter les sources lumineuses et estimer les propriétés de réflectance des surfaces (8 séries couvrant un angle d’élévation de la caméra entre  $40^\circ$  et  $50^\circ$ ).

L’estimation des sources et des BRDFs demande environ 30 minutes de calculs pour le clown et un peu plus de deux heures pour le tournesol avec un processeur Xéon cadencé à 2.4 GHz. L’estimation est plus lente sur le tournesol car le nombre de voxels par groupe est plus important (le nombre de classes de réflectance est plus faible). Le tableau [3.4](#) montre le résultat de cette estimation pour les deux objets. Vu la petite taille des objets, les sources lumineuses sont trop loins pour être détectées comme des sources ponctuelles. En effet, chaque source éclaire l’objet



Objet	Distance réelle objet-sources	Erreur sur		Nombre de sources supplémentaires incohérentes
		la direction de la 1 <sup>re</sup> source	la direction de la 2 <sup>e</sup> source	
<i>clown</i>	1m	12°	16°	2 sources directionnelles
<i>tournesol</i>	1m	6°	11°	1 source directionnelle

Tableau 3.4 – Détection de deux sources lumineuses à partir d’objets réels ; le clown, mesurant 8cm, est reconstruit avec un octree de profondeur 7 et le tournesol, 12cm, avec un octree de profondeur 8.

sur un cône de 7°, ce qui signifie que la direction d’incidence d’un point de la surface à un autre varie au maximum de 7°.

Les directions des sources lumineuses sont obtenues avec une imprécision inférieure à 16° sur les deux objets. Cette imprécision est plus importante que lors de l’utilisation de lumigraphes synthétiques mais les informations utiles à une détection précise sont moins conséquentes : un nombre de points de vue plus faible est utilisé et certaines cavités des objets ne sont pas reconstruites (voir section D.6 du chapitre II). De plus des auto-interréflexions apparaissent sur les objets, conduisant à la détection de quelques sources lumineuses supplémentaires en contradiction avec les sources lumineuses réelles.

Les résultats sont de meilleure qualité sur le tournesol : ceci s’explique par une résolution plus importante des images originales, un meilleur étalonnage de la caméra que pour le clown et une réflectance plus diffuse de la surface. En effet, notre méthode d’estimation de la réflectance est sensiblement plus précise pour les surfaces diffuses.

## F Raffinement sur chaque BRDF

Dans la section E, les propriétés de réflectance sont estimées sur chaque groupe de voxels avec une évaluation de l’erreur entre les luminances estimées par le modèle paramétrique et les échantillons de luminance réelle. Ensuite, seuls les groupes de plus faible erreur sont utilisés pour représenter la BRDF de leur classe de réflectance associée.

Ceci entraîne généralement une estimation de la BRDF d’une surface uniforme par un nombre très restreint de voxels alors que nous disposons de bien plus d’échantillons de luminances. Nous proposons donc de repartir des classes de réflectance (groupes de voxels de même teinte) et de grouper les voxels en fonction des sources lumineuses visibles depuis ces voxels.

Ainsi, pour chaque classe de voxels de même réflectance et éclairés par une seule et même source lumineuse, nous appliquons l’estimation des propriétés de réflectance de la surface en identifiant les paramètres du modèle de Phong modifié. Lorsque les propriétés de réflectance d’une surface sont estimés pour chaque source lumineuse visible, le rapport de luminance entre les différentes sources lumineuses peut être établi.

La figure 3.21.a montre une estimation du modèle de Phong modifié à partir de l’ensemble des échantillons de réflectance de la surface bleue du clown, éclairée par la première source lumineuse détectée. Même si l’algorithme de descente de gradient permet d’obtenir les meilleures valeurs des coefficients du modèle de BRDF, le résultat n’est pas toujours celui espéré. La plupart des échantillons des voxels éclairés par une lumière rasante sont plus lumineux que prévus, entraînant une forte élévation des produits  $L_s.K_d$  et  $L_s.K_s$  estimés. Pour résoudre ce problème, nous proposons de restreindre l’ensemble des échantillons utilisés par l’estimation à des valeurs

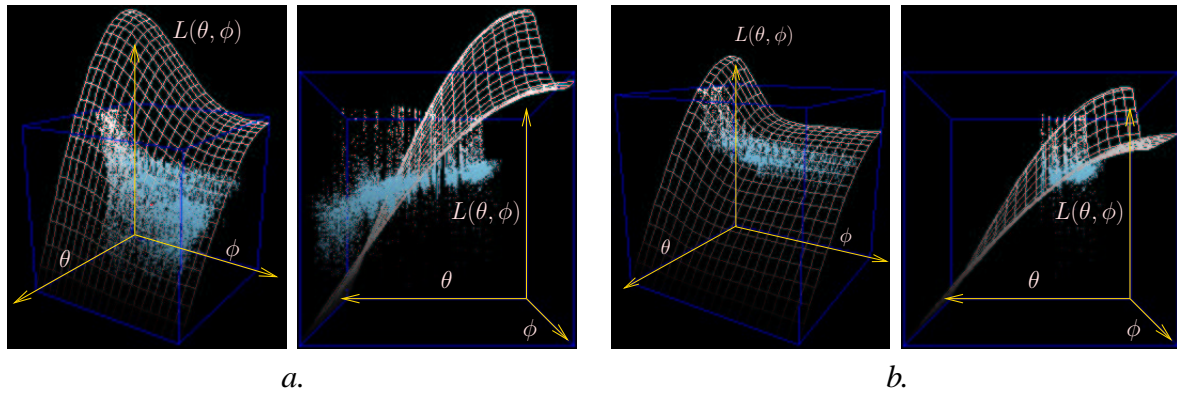


Figure 3.21 – a. Estimation de la BRDF de la surface bleue du clown éclairée par la première source lumineuse détectée ; b. Même estimation en restreignant les échantillons utilisés à des angles d'incidence  $\theta < 45^\circ$ .

d'angle incident  $\theta < 45^\circ$ . La figure 3.21.b montre le résultat obtenu après cette nouvelle définition.

A partir de l'ensemble des sources lumineuses détectées, cela permet d'estimer les propriétés de réflectance de toutes les surfaces éclairées, y compris les surfaces blanches ou grises : les classes regroupant les voxels de teinte blanche ou les voxels de teinte grise ne sont pas utilisées pour la détection des sources lumineuses et n'ont donc pas d'estimation de leurs propriétés de réflectance lors de l'estimation conjointe de la section E.

## G Raffinement sur l'espace image

Dans la section E, les sources lumineuses sont estimées à partir d'un ensemble de voxels très restreint afin d'utiliser des échantillons de luminance provenant d'une surface de réflectance uniforme et éclairée par une seule source lumineuse.

Les voxels peuvent être classés différemment afin d'augmenter le nombre de voxels par classe. Par exemple, les voxels peuvent être classés uniquement en fonction de leur éclairage (selon leur normale). Ceci permet toujours la détection d'une source lumineuse par classe de voxels mais les luminances d'une même classe ne peuvent plus être comparées (les voxels n'appartiennent plus à une surface uniforme). Par conséquent, la détection de la direction de réflexion miroir sur chaque voxel est la seule méthode d'estimation de source pouvant être appliquée.

Au lieu de travailler dans l'espace objet en classant les voxels selon leur normale puis en estimant la direction de réflexion miroir à partir de quelques échantillons de luminance, nous proposons de travailler dans l'espace image afin de détecter les sources lumineuses avec une plus grande précision. En effet sur chaque image d'un objet, des tâches blanches correspondant à des halos spéculaires peuvent être détectées et chacun de leurs centres correspond à une direction de réflexion miroir. Cette technique permet d'estimer précisément des directions de réflexion miroir donc des directions d'incidence pour les voxels correspondant au centre des halos spéculaires. Elle permet également d'utiliser les voxels éclairés par plusieurs sources lumineuses puisqu'elle est basée sur la luminance émise par un large morceau de surface dans une direction particulière plutôt que sur la luminance émise par un élément de surface (le voxel) dans toutes les directions.

Nous utilisons ces informations pour détecter un ensemble de sources lumineuses directionnelles. Les halos spéculaires, correspondant à des pixels de très forte valeur de luminance et/ou de très forte valeur de saturation, sont détectés sur les images et les 4 – 5 pixels correspondant à leur centre sont projetés sur le modèle géométrique de l’objet afin de déterminer les directions d’incidence associées. Toutes ces directions d’incidence sont représentées par des coordonnées sphériques dans un repère global (figure 3.22.a). L’ensemble des directions d’incidence obtenu à

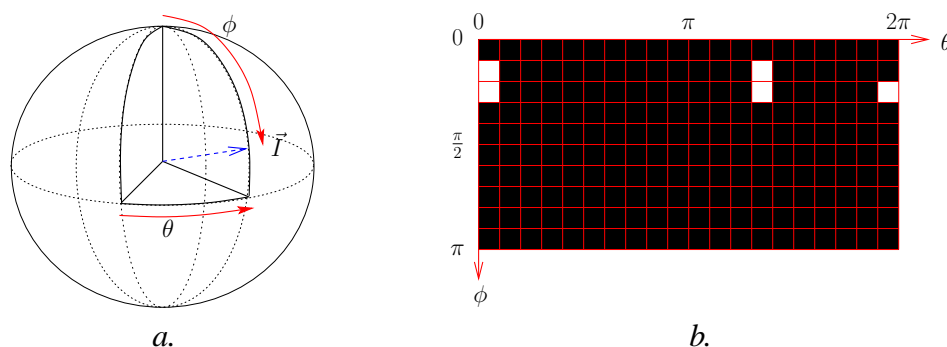


Figure 3.22 – a. Représentation d’une direction d’incidence par les angles  $\theta$  et  $\phi$  du système de coordonnées sphériques ; b. enregistrement des directions d’incidence dans une image puis seuillage pour faire apparaître le nombre de sources lumineuses.

partir de toutes les images représente une ou plusieurs directions de sources lumineuses directionnelles. Une classification de ces directions est obtenue en appliquant l’algorithme des K-means avec une différence d’angle comme critère de distance mais une estimation du nombre de classes (nombre de sources lumineuses) est requise pour pouvoir l’appliquer. Le nombre de classes est détecté automatiquement en enregistrant chaque direction d’incidence obtenue dans une image paramétrée par les angles  $\theta$  et  $\phi$  du système de coordonnées sphériques : chaque pixel contient le nombre de directions d’incidence qui lui sont destinées (figure 3.22.b). Comme les pixels de l’image n’ont pas tous la même valeur d’angle solide, le nombre de directions d’incidence stocké dans chaque pixel est divisé par son angle solide. Ensuite un seuillage de l’image permet d’obtenir un ensemble de régions connexes de pixels représentant les différentes directions de sources lumineuses (nous utilisons la demi-valeur du pixel le plus important comme valeur de seuil).

Objet	Distance réelle objet-sources	Erreur sur		Nombre de sources supplémentaires incohérentes
		la direction de la 1 <sup>re</sup> source	la direction de la 2 <sup>e</sup> source	
<i>clown</i>	1m	7°	11°	0

Tableau 3.5 – Détection de deux sources lumineuses à partir des halos spéculaires visibles sur les images du clown.

Nous avons testé cette nouvelle méthode d’estimation de sources lumineuses sur les images du clown. La figure 3.22.b représente l’image seuillée des directions d’incidence obtenues. Nous pouvons remarquer que seulement deux régions connexes apparaissent sur cette image et l’application de l’algorithme des K-means permet d’obtenir les deux sources directionnelles avec une imprécision de 7° sur l’estimation de la première source et 11° sur la seconde, soit une amélioration de 5° par rapport à la précédente estimation (voir tableaux 3.4 et 3.5). Cette méthode est

également intéressante car elle est plus rapide (les sources sont détectées en 2 – 3 minutes) et ne détecte pas de sources lumineuses incohérentes.



# RÉÉCLAIRAGE D'OBJETS RÉELS

---

## Sommaire

---

<b>A</b>	<b>Intégration d'objets réels dans des environnements virtuels . . . . .</b>	<b>166</b>
A.1	Champs de lumière précalculés . . . . .	166
A.2	Nouvelle structure de représentation . . . . .	167
<b>B</b>	<b>Compression de champs de lumière . . . . .</b>	<b>168</b>
B.1	Quantification vectorielle . . . . .	168
B.2	Résultats . . . . .	172
<b>C</b>	<b>Rééclairage par lancer de rayons . . . . .</b>	<b>172</b>
C.1	Equation d'illumination locale . . . . .	173
C.2	BRDFs uniformes . . . . .	173
C.3	BRDFs variées . . . . .	176
C.4	Temps de calcul global . . . . .	179
C.5	Objet synthétique rééclairé . . . . .	180
C.6	Objet réel rééclairé . . . . .	180

---

## A Intégration d'objets réels dans des environnements virtuels

L'intégration réaliste d'un objet basé-image dans un environnement virtuel nécessite un ré-éclairage de l'objet avec les conditions d'illumination de l'environnement, généralement différentes des conditions d'illumination utilisées lors de l'acquisition du champ de lumière. L'état de l'art sur le rééclairage d'objets réels montre qu'un objet représenté uniquement par un champ de lumière (4 dimensions) ne peut être rééclairé. En effet, le rééclairage d'un objet basé-image nécessite un champ de réflectance complet (8 dimensions) afin de pouvoir utiliser de nouvelles conditions d'illumination : cette forme de représentation requiert un stockage d'informations gigantesque, impraticable dans le cadre de la manipulation de plusieurs objets.

En revanche, la connaissance de la géométrie et de la BRDF de chaque objet permet d'utiliser tout algorithme d'illumination locale ou globale afin de synthétiser de nouvelles vues réalistes de la scène : cet ensemble d'informations, plus compact que les champs de réflectance, permet d'évaluer les ombres projetées ainsi que les interrélaxions entre la scène et les différents objets (section E du chapitre I). Nous avons donc choisi d'utiliser le modèle géométrique reconstruit et les propriétés de réflectance estimées en chaque voxel afin de rééclairer les objets réels avec de nouvelles conditions d'illumination (voir section C pour plus de détails).

La mise en œuvre réalisée pour le rééclairage d'objets prend en compte toutes les informations estimées dans ce travail, i.e. le maillage de triangles, la géométrie basée-voxel, la normale et la BRDF paramétrique de chaque voxel afin de synthétiser de nouvelles vues réalistes par un algorithme de lancer de rayons. A ce jour, nous n'avons mis en place aucune optimisation pour réduire la taille des données utilisées ou accélérer les temps de synthèse d'une image. Nous prévoyons d'approfondir cette partie pour permettre une intégration interactive d'objets réels dans des environnements virtuels.

### A.1 Champs de lumière précalculés

Une forme d'intégration réaliste d'objets réels dans un environnement virtuel peut être l'utilisation de champs de lumière précalculés représentant chaque objet rééclairé avec les conditions d'illumination de l'environnement. Chaque image du champ de lumière représentant un objet est synthétisée par un algorithme d'illumination locale ou globale. L'utilisation d'informations géométriques et de réflectance de l'environnement et de l'ensemble des objets permet de calculer les ombres projetées et les interrélaxions sur chaque image du champ de lumière (auto-ombrage, ombres projetées sur l'objet à rééclairer, auto-interrélaxions, interrélaxions avec les autres objets et la scène). Une exploitation similaire est effectuée par W. Matusik et al. en 2002 [MPN<sup>+</sup>02] où un champ de lumière d'un objet est généré à partir d'un champ de réflectance et des conditions d'illumination de l'environnement d'insertion.

Avec des champs de lumière précalculés comme unique représentation des objets, n'importe quel point de vue de la scène peut être généré en utilisant ses informations géométrique et photométrique. Chaque nouvelle vue peut être synthétisée par un algorithme de lancer de rayons :

- soit le pixel de l'image produite correspond à une information provenant d'un objet, auquel cas il suffit de lancer le rayon correspondant à travers le champ de lumière de l'objet pour récupérer l'échantillon *wst* le plus proche : aucun calcul d'illumination n'est nécessaire pour évaluer ce pixel ;



- soit le pixel correspond à une information provenant de la scène, auquel cas la luminance émise par la scène doit être évaluée au point d'intersection par un algorithme d'illumination.

Les ombres projetées et les interrélaxions peuvent être évaluées sur les surfaces de la scène par un algorithme de lancer de rayons. Le point de la scène correspondant au pixel à afficher peut être ombré en traçant un rayon d'ombre vers chaque source lumineuse et en vérifiant si ce rayon traverse la silhouette des objets pour chaque champ de lumière. Les interrélaxions avec les objets peuvent également être évaluées en traçant un rayon dans la direction miroir de la caméra et en vérifiant si les silhouettes des objets contenues dans les champs de lumière sont traversées par ce rayon.

Cette méthode permet ainsi d'insérer des objets réels dans la scène sans augmenter les temps de rendus puisque la géométrie complexe des objets insérés n'est pas utilisée lors de la synthèse des images finales. Le précalcul peut également être utilisé pour générer des cartes de photons sur les polygones de la scène ou pour effectuer une simulation d'éclairage par radiativité (la géométrie des objets est alors utilisée pendant cette phase).

Néanmoins, cette méthode de rééclairage ne permet pas de modifier dynamiquement les conditions d'illumination ou la position des objets insérés dans la scène. Ce type de modification entraîne systématiquement un recalcul de l'ensemble des champs de lumière afin d'intégrer les perturbations d'illumination.

Nous utilisons généralement des champs de lumière restreints en nombre de vues ( $8 \times 8$  vues par slab) mais un lumigraphe complet (384 images de résolution  $256 \times 256$ ) représente tout de même  $72Mo$  d'espace mémoire. Par conséquent, l'insertion de plusieurs champs de lumière dans un environnement pose des problèmes de coût en mémoire. Pour réduire de manière efficace l'espace mémoire, nous avons développé une méthode de compression adaptée. Une quantification vectorielle est utilisée pour réduire la taille des données par un facteur 15 sans dégradation visible et pour avoir une décompression instantanée sur tout échantillon *uvst* du champ de lumière (voir section B pour plus de détails).

Le faible nombre de vues par champ de lumière n'autorise pas l'interpolation des échantillons *uvst* provenant de plusieurs vues sans utiliser une information de profondeur sur ces champs. Nous utilisons actuellement un canal supplémentaire sur chaque échantillon afin de conserver cette information (33% d'espace de stockage supplémentaire mais la profondeur peut également être compressée par quantification vectorielle). Une autre solution est d'utiliser un maillage polygonal représentant la géométrie approximative des objets insérés, comme dans les lumigraphes de S. Gortler et al. [GGSC96]. Dans ce cas, nous pouvons par exemple utiliser l'octree reconstruit en stockant une information binaire par voxel (définit / ne définit pas l'objet).

## A.2 Nouvelle structure de représentation

Puisque l'utilisation d'un faible nombre de points de vue par champ de lumière nécessite l'utilisation d'une représentation géométrique de chaque objet lors de la synthèse d'une image de l'environnement, alors la luminance émise par chaque objet peut être évaluée dynamiquement en utilisant notre représentation d'objet par octree. En effet, un octree de profondeur 7 occupe actuellement  $20Mo$  d'espace mémoire mais il permet de stocker dans chaque voxel de la surface : une normale, une liste de triangles, une BRDF paramétrique et une liste de 255 échantillons de luminance. Les échantillons de luminance n'étant pas utiles pour un rééclairage utilisant la BRDF paramétrique, la taille de l'octree peut être réduite à une centaine d'octets par voxel, soit moins

de  $2Mo$  par objet (moins de 20000 voxels par objet reconstruit, voir tableau 2.6 du chapitre II). Un algorithme de rendu peut ensuite être utilisé sur ce modèle compact de représentation.

Beaucoup de travaux utilisent déjà des objets représentés par un modèle polygonal (liste de triangles) et des textures sur chaque triangle afin de stocker les informations de normale, de BRDF paramétrique, de visibilité, etc [SKS02, MLH02]. Le maillage de triangles est effectivement intéressant car il permet à la carte graphique d'effectuer des traitements efficaces sur chaque polygone du modèle (produit de textures, placage de textures) afin d'obtenir un rendu interactif (voir section E du chapitre I).

## B Compression de champs de lumière

Un champ de lumière, tel que nous l'utilisons, est un ensemble structuré d'images représentant un objet par divers points de vue. Des méthodes de compression d'images peuvent donc être utilisées. Il existe principalement trois grandes familles de méthodes de compression d'images permettant de réduire de façon significative la taille d'une image sans dégradation visuelle importante :

- la transformée en cosinus discrète (DCT), utilisée par le format JPEG ;
- la transformée en ondelettes, utilisée par le format JPEG 2000 ;
- et la quantification vectorielle.

Les transformées en cosinus discrète et en ondelettes sont des méthodes atteignant des taux de compression très élevés mais ne peuvent être retenues pour une application de visualisation de champs de lumière en temps interactif : elles ne permettent pas d'accéder directement à la valeur d'un pixel, toute une partie de l'image devant être décompressée pour obtenir la valeur d'un pixel.

Par conséquent notre choix s'est naturellement porté sur la quantification vectorielle [GG92], méthode très efficace car elle permet d'accéder à la valeur de n'importe quel pixel en un temps constant : ce temps d'accès est important pour notre outil de visualisation de champs de lumière.

### B.1 Quantification vectorielle

La quantification vectorielle est une méthode réduisant le nombre de couleurs dans une image en choisissant les plus représentatives et en les indexant dans un dictionnaire. Chaque pixel de l'image est codé par une référence à la couleur la plus proche stockée dans le dictionnaire plutôt que par ses trois composantes chromatiques *RGB*. Pour un dictionnaire contenant 256 couleurs soit  $2^8$  valeurs, une référence à une couleur du dictionnaire est codée sur 8 bits au lieu de 24 (8 bits par composante chromatique), soit un facteur de compression de 3 excepté le fait que le dictionnaire doit également être stocké (format GIF).

Pour augmenter le taux de compression les valeurs stockées dans le dictionnaire ne sont pas toujours des pixels, elles peuvent être des blocs de pixels appelés *vecteurs 2D*. Dans le cas des champs de lumière, non seulement les blocs de pixels peuvent apparaître plusieurs fois dans une même image mais ils peuvent également apparaître dans d'autres points de vue puisque les images représentent un seul et même objet. Par conséquent un dictionnaire peut être utilisé pour encoder plusieurs images : cela limite la taille prise par les dictionnaires dans le champ de lumière compressé.

Une autre optimisation est la définition de *vecteurs 4D* représentant des blocs de pixels sur les quatre dimensions du champ de lumière (*uvst*). Les vecteurs 4D représentent des blocs de pixels définis sur plusieurs points de vue simultanément et ont été utilisés par M. Levoy et P. Hanrahan [LH96].

### B.1.1 Choix de la représentation des données

Dans le cadre du stage de master recherche de Raphaël Lerbour, nous avons développé une méthode de quantification vectorielle permettant de paramétrer :

- la dimension des vecteurs (2D ou 4D),
- la taille des vecteurs (par exemple  $2 \times 2$  pixels pour un vecteur 2D),
- le nombre de dictionnaires (1 par image, 1 pour 4 images, ..., 1 par champ de lumière),
- et la taille des dictionnaires (par exemple  $2^8$  vecteurs)

afin d'effectuer une série de tests sur nos différents champs de lumière.

Ceci nous a permis de montrer que l'utilisation de vecteurs 4D est particulièrement bien adaptée à la compression de champs de lumière représentés par  $32 \times 32$  vues sur chaque slab mais n'est pas efficace pour des champs de lumière représentés uniquement par  $8 \times 8$  vues sur chaque slab. En effet, pour compresser les champs de lumière dont nous disposons, l'utilisation de vecteurs 2D est plus appropriée (tableau 4.1). La qualité des images compressées est évaluée

Champ de lumière	Tournesol (4 slabs de $8 \times 8$ vues)		Bouddha (1 slab de $32 \times 32$ vues)	
	4D	2D	4D	2D
<i>Dimension des vecteurs</i>	4D	2D	4D	2D
<i>Taille des dictionnaires</i>	16 384 vecteurs	256 vecteurs	16 384 vecteurs	256 vecteurs
<i>Place mémoire</i>	1 173 Ko	916 Ko	2598 Ko	4 104 Ko
<i>PSNR</i>	30.6 dB	30.7 dB	33.4 dB	32.7 dB

Tableau 4.1 – Comparatif entre l'utilisation des vecteurs 2D et 4D pour des champs de lumière différant sur le nombre de points de vue. Le tournesol est un de nos objets réels et le bouddha est un objet utilisé dans [LH96].

par une estimation du PSNR (*Peak Signal to Noise Ratio*).

Nous avons également testé la taille des vecteurs et le nombre de dictionnaires par slab afin de définir expérimentalement les meilleures valeurs correspondant à nos champs de lumière. Avec 1 dictionnaire par image, le meilleur taux de compression est obtenu pour des vecteurs 2D de taille  $2 \times 2$  pixels (tableau 4.2).

Taille des vecteurs 2D	$1 \times 1$ pixel	$2 \times 2$ pixels	$2 \times 2$ pixels	$4 \times 4$ pixels
<i>Taille des dictionnaires</i>	64 vecteurs	1 024 vecteurs	512 vecteurs	2 048 vecteurs
<i>Taux de compression</i>	4 :1	8.5 :1	10 :1	6.5 :1
<i>PSNR</i>	34.2 dB	34.4 dB	33.4 dB	33.0 dB

Tableau 4.2 – Comparatif selon la taille des vecteurs 2D. Le meilleur taux de compression est obtenu pour des vecteurs de taille  $2 \times 2$  pixels (à qualité d'images équivalente).

Les vecteurs 2D de  $2 \times 2$  pixels sont ensuite utilisés pour évaluer le nombre de dictionnaires par slab à utiliser. Expérimentalement, nous estimons que les meilleurs résultats sont obtenus avec 4 dictionnaires par slab de  $8 \times 8$  vues (tableau 4.3).

Nombre de dictionnaires par slab	64	16	4	1
Taille des dictionnaires	128 vecteurs	256 vecteurs	512 vecteurs	1 024 vecteurs
Place mémoire par slab	1 144 Ko	1 060 Ko	1 073 Ko	1 133 Ko
PSNR	30.8 dB	31.8 dB	32.6 dB	33.3 dB

Tableau 4.3 – Comparatif selon le nombre de dictionnaires par slab avec des vecteurs 2D de taille  $2 \times 2$  pixels. Les images du tournesol sont utilisées pour afficher les résultats de ce tableau. Le meilleur rapport PSNR / Place mémoire est obtenu pour l'utilisation de 4 dictionnaires par slab de  $8 \times 8$  vues.

Un paramètre n'est pas encore fixé, la taille des dictionnaires qui doit varier en fonction de la qualité des images compressées que nous souhaitons obtenir. La section suivante [B.1.2](#) montre qu'elle peut être définie de manière automatique afin d'assurer une qualité visuelle identique pour tous les champs de lumière compressés.

### B.1.2 Algorithmes utilisés

La quantification est le processus de compression des images : elle consiste à construire un dictionnaire à partir d'un ensemble d'apprentissage puis à encoder chaque donnée du champ de lumière par l'index du vecteur le plus représentatif dans le dictionnaire.

Nous utilisons un algorithme de quantification basé sur l'ensemble d'apprentissage complet, i.e. toutes les images du champ de lumière sont utilisées pour construire un dictionnaire le plus représentatif possible. Dans [\[LH96\]](#), l'ensemble d'apprentissage est restreint à quelques vues pour accélérer les temps de construction du dictionnaire. Nous n'appliquons pas cette restriction car l'utilisation de vecteurs 2D accélère les traitements et l'algorithme de quantification utilisé est très rapide, permettant de compresser n'importe lequel de nos champs de lumière en quelques minutes seulement.

Le coeur de notre algorithme est une application successive de *l'itération de Lloyd* sur un dictionnaire de taille fixe afin de construire un dictionnaire de plus en plus représentatif par partitionnement successif de l'ensemble d'apprentissage : une évaluation du PSNR est exécutée à chaque itération pour juger la qualité du dictionnaire nouvellement construit et définir un critère d'arrêt [\[Llo82\]](#).

Les itérations de Lloyd permettent de raffiner un dictionnaire déjà existant pour produire un dictionnaire plus représentatif de l'ensemble d'apprentissage mais ne permettent pas la construction du dictionnaire initial. Pour construire le dictionnaire initial, nous utilisons l'algorithme *LBG* consistant à créer un dictionnaire avec un unique vecteur puis à doubler successivement la taille du dictionnaire par dédoublement de chaque vecteur construit [\[LBG80\]](#). Nous appliquons les itérations de Lloyd à chaque dédoublement du dictionnaire afin de produire le dictionnaire le plus représentatif pour chaque taille.

Plutôt que de fixer arbitrairement la taille maximale du dictionnaire (nombre de vecteurs), nous utilisons l'évaluation du PSNR obtenue par les itérations de Lloyd afin de définir une taille de dictionnaire qui satisfasse un critère de qualité (PSNR supérieur à un seuil fixé). Par conséquent les dictionnaires utilisés pour compresser un champ de lumière complet n'ont pas toujours la même taille : les groupes d'images représentant des surfaces plus uniformes ont besoin de moins de vecteurs représentatifs et peuvent être compressés avec des dictionnaires de plus petite taille (4 dictionnaires sont définis par slab soit 24 au total). La section [B.2](#) montre que la valeur du PSNR peut être fixée après l'analyse des résultats de tests psychovisuels.

### B.1.3 Optimisations

Afin d'augmenter le taux de compression des champs de lumière, nous stockons un champ de lumière compressé sous la forme d'un seul fichier contenant :

- un entête de taille fixe décrivant les paramètres de compression du champ de lumière complet ;
- les dictionnaires stockant les vecteurs représentatifs des images ;
- les images compressées, représentées par des références aux vecteurs.

Nous définissons également une boîte englobante par image, afin de limiter la région des pixels à encoder à la zone représentative de l'objet : un cadre noir apparaît autour de l'objet sur chaque image et ne peut généralement pas être évité lors de la construction d'un champ de lumière complet. Les tests effectués montrent un gain de 50% à 75% suivant les champs de lumière compressés (tableau 4.4).

Nous conservons également l'information de contour de l'objet, présente dans les images originales, en ajoutant un masque binaire par boîte englobante dans la structure compressée : il sert à optimiser l'algorithme de construction des dictionnaires en insérant uniquement les pixels représentant l'objet dans l'ensemble d'apprentissage ; il permet également une visualisation plus réaliste des champs de lumière compressés puisque les informations précises des contours de l'objet sont conservées. Ce n'est pas le cas dans [LH96] où les contours de l'objet sont compressés par les vecteurs 4D, blocs de 16 pixels, entraînant des artefacts visuels lors de la visualisation. Le masque binaire a un coût non négligeable dans la structure compressée : dans le pire des cas, i.e. lorsque les dictionnaires contiennent seulement 256 vecteurs, le masque correspond à 50% de la taille des images compressées soit 1 bit par pixel de la boîte englobante. Toutefois ce coût est réduit lorsque la taille des dictionnaires augmente, sachant qu'un dictionnaire doit comporter plus de 256 vecteurs pour obtenir des images compressées sans dégradation visuelle importante. Les tests montrent que le gain obtenu par l'utilisation des boîtes englobantes est toujours supérieur à la perte due au stockage des masques binaires, même pour des dictionnaires de 256 vecteurs (tableau 4.4). De plus, le masque binaire peut être compressé sur le disque sous forme

Champ de lumière	Tournesol (4 slabs)	Clown (5 slabs)	Vélo (6 slabs)
<i>compressé sans BE ni M</i>	4 096 Ko	5 120 Ko	6 144 Ko
<i>compressé avec BE sans M</i>	916 Ko	1 479 Ko	3 110
<i>compressé avec BE et M</i>	1 346 Ko	2 183 Ko	4 620 Ko

Tableau 4.4 – Comparatif de champs de lumière compressés avec et sans utilisation des boîtes englobantes (BE) et des masques (M). Ces résultats sont obtenus avec des dictionnaires définis sur 256 vecteurs, cas où le stockage des masques prend une grande importance comparé au stockage des images fortement compressées.

d'un codage RLE (*Run Length Encoding*) ou un autre type de compression. Actuellement, nous stockons le masque sans compression afin d'y accéder de manière instantanée.

Le développement de la compression des champs de lumière a été réalisé en collaboration avec Raphaël Lerbour, étudiant de master recherche 2 de l'université de Poitiers, et va prochainement être soumis en vue d'une communication dans une conférence internationale. Des détails complémentaires sont disponibles dans le mémoire de stage [Ler06].

## B.2 Résultats

Nous avons mené une campagne de tests psycho-visuels sur la série de champs de lumière à compresser : une évaluation subjective de la qualité des images compressées est effectuée sur une population constituée d'une vingtaine de personnes extérieures à nos travaux. L'étude a permis de montrer que le PSNR est une estimation fiable de la qualité des images compressées :

- le coefficient de corrélation de Pearson, calculé entre le résultat de l'évaluation (Score d'Opinion Moyen, MOS) et le PSNR, montre une corrélation forte entre la mesure subjective et la mesure objective avec plus de 83% ;
- les images compressées avec un PSNR de 32.5dB ne peuvent être distinguées des images originales pour plus de 70% des observateurs.

Nous avons donc choisi d'utiliser le seuil de 32.5 dB pour la valeur de PSNR utilisée dans l'algorithme LBG afin d'ajuster la taille de chaque dictionnaire en conséquence (voir section B.1.2). Le dictionnaire étant construit récursivement par dédoublement des vecteurs, la valeur de PSNR calculée varie par paliers et elle est généralement retenue aux alentours de 33dB. Le tableau 4.5 montre les résultats de notre méthode de compression automatique sur quelques champs de lumière.

Champ de lumière	Tournesol (4 slabs)	Clown (5 slabs)	Vélo (6 slabs)
<i>PSNR</i>	33.0 dB	33.2 dB	33.4 dB
<i>Taux de compression</i>	32 :1	21 :1	16 :1
<i>Place mémoire</i>	1.52 Mo	2.90 Mo	4.47 Mo
<i>Temps de compression</i>	5 min	26 min	6 min

Tableau 4.5 – Résultats de la compression automatique de champs de lumière. Ces résultats sont obtenus en fixant le PSNR à un seuil de 32.5dB (mesure qualitative) contrairement aux méthodes fixant généralement le taux de compression (mesure quantitative).

Sur l'ensemble des champs de lumière testés, nous obtenons un facteur de compression supérieur à 15 avec un enregistrement non compressé du masque binaire et une qualité visuelle constante pour tous les objets : la visualisation des champs de lumière compressés est par conséquent possible et visuellement indiscernable de la visualisation des champs de lumière non compressés. En effet la valeur *RGB* d'un échantillon *wvst* est obtenue par un accès direct, après un nombre de tests ne pouvant excéder 5 :

- test avec la boîte englobante (4 tests maximum) ;
- test avec le masque (1 test) ;
- récupération de l'indice du vecteur (accès direct) ;
- récupération de la valeur RGB dans le dictionnaire correspondant (accès direct).

## C Rééclairage par lancer de rayons

L'état de l'art sur le rééclairage d'objets réels montre que la prise en compte de nouvelles conditions d'éclairage est complexe et nécessite un recalcul de la luminance émise par les objets en direction de chaque nouveau point de vue synthétisé (section E du chapitre I). Un modèle géométrique (liste de triangles) est généralement utilisé pour évaluer la luminance à partir d'une



BRDF paramétrique et pour appliquer des effets d’auto-ombrage sur l’objet. Certaines optimisations permettent d’accélérer les traitements en stockant de multiples cartes de coefficients sur chaque triangle du modèle géométrique (les coefficients de BRDFs paramétriques, de normales, de transparence, d’illumination, de visibilité, etc) et en effectuant un placage successif de ces textures lors du rendu [SKS02, MLH02]. Des opérations de produits de textures sont effectuées sur les cartes graphiques afin de ne projeter que la texture finale sur chaque triangle, mais les calculs de réflexion spéculaire sont encore peu précis avec des textures : ces calculs relèvent plus de l’optimisation dans le cadre du rendu temps-réel, qui n’est pas abordé dans notre travail.

## C.1 Equation d’illumination locale

Nous avons choisi de rééclairer les objets par une méthode de lancer de rayons : c’est un modèle d’illumination locale pour lequel les optimisations par placage de textures sont facilement applicables [SKS02, MLH02]. De plus, l’utilisation d’un modèle d’illumination locale permet d’accélérer les calculs d’illumination en négligeant les auto-interréflexions. Nous verrons dans les résultats que ce modèle a ses limites lors du rééclairage réaliste des objets réels.

Un algorithme de lancer de rayons est utilisé pour évaluer la luminance traversant chaque pixel de l’image synthétisée. Une luminance émise par un point  $x$  de la surface de l’objet est évaluée par discrétisation de l’équation de réflectance sur les nouvelles conditions d’illumination :

$$L_{r,c}(x, \vec{R}) = L_{a,c} \cdot K_{a,c} + \sum_{s=1}^{NbSV} f_{r,c}(x, \vec{R}, \vec{I}_s) \cdot L_{i,c}(x, \vec{I}_s) \cdot \cos \theta_s \cdot \frac{1}{r_s^2} \quad (4.1)$$

avec  $\vec{I}_s$  la direction et  $r_s$  la distance de la source  $s$  au point  $x$  ;  $\theta_s$  l’angle d’incidence formé avec la source  $s$  au point  $x$  ;  $L_{i,c}(x, \vec{I}_s)$  la luminance reçue en  $x$  provenant de la source  $s$  ;  $\vec{R}$  la direction du pixel depuis  $x$  et  $f_r$  la BRDF paramétrique de la surface. Un terme d’illumination ambiante  $L_a \cdot K_a$  est ajouté pour compenser les auto-interréflexions diffuses non calculées et ainsi éviter que les surfaces non éclairées apparaissent complètement noires.  $NbSV$  représente le nombre de sources visibles depuis le point  $x$ . Cette équation de luminance est évaluée pour les 3 composantes chromatiques  $c$  de chaque pixel de l’image finale.

## C.2 BRDFs uniformes

### C.2.1 Surfaces non éclairées

La section F du chapitre III montre qu’une BRDF est estimée pour chaque groupe de voxels représentant une surface de réflectance uniforme. Un modèle de BRDF paramétrique est donc défini sur l’ensemble des voxels éclairés de la surface ; seule la classe regroupant les voxels sans éclairage direct n’a pas d’estimation de BRDF car nous ne disposons pas d’informations de réflectance.

Il est difficile d’appliquer l’équation d’illumination en tout point de la surface sans la connaissance de la BRDF  $f_r$  sur l’intégralité de la surface. Les voxels n’ayant aucune estimation de  $f_r$  représentent soit une partie de surface sans éclairage direct (à l’ombre de toutes les sources lumineuses), soit un infime élément de surface sur lequel aucun pixel des images originales n’est projeté. Nous proposons donc une stratégie en plusieurs passes, permettant de définir une BRDF sur chaque voxel, avec 3 niveaux de priorité différents :



- premièrement, tous les voxels sans BRDF, entourés de voxels voisins ayant une même BRDF uniforme  $f_r$ , ont finalement la même BRDF que leurs voisins (ceci permet, par exemple, de combler en priorité les voxels représentant un infime élément de surface sans échantillon de luminance) ;
- deuxièmement, tous les voxels sans BRDF, ne disposant d'aucun échantillon de luminance et entourés de voxels voisins ayant chacun leur BRDF, ont finalement la même BRDF que la BRDF commune au plus grand nombre de voxels voisins ;
- troisièmement, tous les voxels sans BRDF, disposant d'échantillons de luminance pour calculer leur luminance moyenne  $L^{moy}$  et entourés de voxels voisins ayant chacun leur BRDF, ont la même BRDF que le voxel voisin ayant la luminance moyenne la plus proche de  $L^{moy}$ .

Afin de supprimer tout éventuel bruit introduit lors de la précédente classification selon la réflectance (section B.1 du chapitre III), les BRDFs uniformes représentées par des surfaces discrètes de trop petite taille, estimées à 3 voxels ou moins après expérimentations, sont supprimées et les voxels correspondant sont reclassés par un processus identique à celui appliqué sur les voxels non éclairés, présenté ci-dessus.

La figure 4.1 montre le résultat de la définition d'une BRDF sur l'intégralité des voxels de la surface discrète du clown.



Figure 4.1 – De gauche à droite : visualisation de la luminance moyenne de chaque voxel, de la première classification selon la réflectance (section B.1 du chapitre III) puis de la nouvelle classification selon la BRDF sur deux vues différentes du clown. Les surfaces non éclairées ont également une BRDF.

### C.2.2 Géométrie utilisée

Généralement, une grille régulière 3D est utilisée pour accélérer les calculs d'intersection entre les rayons et le maillage de triangles : cette grille est coûteuse en mémoire (chaque cellule de la grille peut potentiellement contenir une liste de triangles) mais le gain de temps obtenu est conséquent (parcours incrémental de la grille). Dans notre cas, une structure d'octree est utilisée, moins coûteuse en mémoire et presque aussi efficace que la grille régulière pour un premier test d'intersection. En effet, un parcours incrémental de l'octree est effectué en utilisant une grille virtuelle 3D dont les cellules sont de la taille des plus petits voxels présents dans l'octree (le coût de stockage de la grille est nul). Enfin, nous raffinons les tests d'intersection en utilisant les triangles stockés dans chaque voxel.

L'octree permet de détecter rapidement les rayons n'atteignant pas l'objet. Nous pouvons également utiliser un champ de lumière binaire où chaque pixel est représenté par un bit de visibilité :  $3Mo$  pour un champ de lumière complet de 384 vues de résolution  $256 \times 256$ . La

réutilisation des triangles construits au chapitre II permet d'affiner la précision du test d'intersection et ainsi d'obtenir des contours lisses de l'objet sur les nouvelles images produites.

Nous n'utilisons pas la normale du triangle atteint mais plutôt celle du voxel associé afin d'utiliser le lissage obtenu lors du processus de reconstruction géométrique. La normale permet de réduire les discontinuités de luminance entre les mailles lors de l'évaluation de  $\cos \theta_s = (\vec{N}(x) \cdot \vec{I}_s(x))$  (équation 4.1). La normale peut être interpolée entre plusieurs voxels suivant le point d'intersection obtenu (comme l'interpolation de normales de Phong, utilisant une définition de normales uniquement sur les sommets de triangles) mais avec un octree de profondeur 7, en moyenne seulement 4 pixels d'une image de résolution  $256 \times 256$  utilisent la normale d'un même voxel. L'interpolation est donc intéressante pour produire des images de plus forte résolution.

### C.2.3 BRDF estimée

L'estimation des propriétés de réflectance selon le modèle de Phong modifié a permis d'obtenir une valeur des coefficients  $K_d$ ,  $K_s$  et  $n$  pour chaque surface représentant une BRDF uniforme (voir chapitre III). Les valeurs sont obtenues par une mise en correspondance entre les sources détectées et les échantillons de luminance mais ne nous permettent pas de recalculer directement la luminance de la surface pour chaque composante chromatique : en effet, nous n'avons pas une estimation de  $f_{r,c}$  mais seulement de  $f_r$  permettant de recalculer une image en niveaux de gris, représentant la luminance.

Nous voulons à présent estimer l'albédo de chaque surface afin de représenter une BRDF par les coefficients  $K_{d,R}$ ,  $K_{d,G}$ ,  $K_{d,B}$ ,  $K_s$  et  $n$ . Les coefficients de spécularité ne sont pas représentés pour chaque composante chromatique car nous avons fait l'hypothèse que la spécularité réfléchissait la teinte de la source indépendamment de la teinte de la surface (cas des matériaux plastiques, voir section D.2 du chapitre I). Pour obtenir les valeurs de  $K_{d,R}$ ,  $K_{d,G}$  et  $K_{d,B}$  à partir de celle de  $K_d$ , nous pourrions utiliser les valeurs de teinte et de saturation stockée dans chaque BRDF uniforme estimée. Comme il n'est pas évident de passer du système de représentation  $HSV$  au système  $RGB$ , nous travaillons plutôt avec les systèmes  $RGB$  et  $XYZ$  (plus simples à manipuler). Le raisonnement est d'abord appliqué sur un échantillon orienté à plus de  $90^\circ$  de la direction de réflexion spéculaire, appartenant à un voxel éclairé par une seule source lumineuse. Les composantes de cet échantillon peuvent être exprimées par :

$$\begin{cases} Y &= K_d \cdot L_s \cdot \frac{1}{\pi \cdot r_s^2} \cdot \cos \theta_s = D \cdot K_d \\ R &= D \cdot K_{d,R} \\ G &= D \cdot K_{d,G} \\ B &= D \cdot K_{d,B} \end{cases} \quad (4.2)$$

en supposant la contribution des interrélaxions négligeable. Ces équations peuvent être réécrites :

$$\begin{cases} D &= \frac{Y}{K_d} \\ K_{d,R} &= \frac{R}{D} = \frac{R \cdot K_d}{Y} \\ K_{d,G} &= \frac{G \cdot K_d}{Y} \\ K_{d,B} &= \frac{B \cdot K_d}{Y} \end{cases} \quad (4.3)$$

Un terme ambiant est inséré dans l'équation 4.1 pour compenser les auto-interrélaxions non prises en compte. La plupart du temps, ces auto-interrélaxions sont négligeables mais elles peuvent apparaître dans l'échantillon choisi pour calculer la valeur des coefficients  $K_{d,R}$ ,  $K_{d,G}$  et  $K_{d,B}$ . Pour supprimer cet éventuel problème, nous n'utilisons pas les valeurs  $R$ ,  $G$ ,  $B$  et  $Y$  d'un

seul échantillon mais nous calculons une moyenne sur tous les échantillons de tous les voxels de la BRDF en cours d'estimation. Uniquement les voxels éclairés par au moins une source lumineuse détectée sont pris en compte et les échantillons utilisés doivent être éloignés de la direction de réflexion miroir afin de réduire la participation de la composante de spécularité.

### C.2.4 Résultats



Figure 4.2 – Rééclairage du clown représenté par 12 BRDFs uniformes. Le résultat n'est pas convaincant : les détails géométriques sont perdus et la surface apparaît trop lisse, trop uniforme.

Le rééclairage d'un objet réel défini par un ensemble de surfaces aux propriétés de réflectance uniformes ne permet pas d'obtenir un résultat réaliste (figure 4.2). H. Lensch et al. ont fait le même constat sur des objets réels rééclairés avec une approximation de la BRDF par le modèle de Lafortune. Ils ont également défini une BRDF uniforme par région mais le rééclairage donne un aspect trop lisse à la surface malgré l'utilisation d'un scanner à lumière structurée pour reconstruire la géométrie de leurs objets [LKG<sup>+</sup>03].

### C.3 BRDFs variées

Généralement, lorsqu'un modèle de BRDF paramétrique est estimé sur la surface d'un objet, celui-ci est représenté par un maillage de triangles obtenu par un scanner et une carte d'albédo est estimée sur chaque triangle [SWI97, NZI01, YDMH99]. Le nombre de triangles définissant leurs modèles est moins important que celui de nos objets reconstruits (il correspond à une reconstruction de l'octree à une profondeur 5) mais une texture est utilisée pour stocker des coefficients de réflexion diffuse différents en chaque texel : ceci permet de représenter des surfaces fortement texturées et de garder un maximum de détails provenant des images originales. En revanche, les coefficients de réflexion spéculaire sont définis moins précisément, généralement une seule valeur par triangle : la spécularité d'un matériau varie peu le long de sa surface, une interpolation de Phong est utilisée sur les normales des triangles et sur les coefficients de spécularité (voir section D.4.2 du chapitre I).

Par conséquent, nous définissons également des propriétés de réflectance variées le long des surfaces de l'objet. Nous utilisons de nouveau le modèle géométrique reconstruit par un octree de profondeur 7, nous stockons une valeur des coefficients de réflexion diffuse par voxel et conservons l'estimation des coefficients de réflexion spéculaire par groupe de voxels. Avec une telle répartition, nous avons une définition des valeurs  $K_{d,R}$ ,  $K_{d,G}$  et  $K_{d,B}$  correspondant à 4 pixels sur chaque image originale : cette répartition des coefficients est suffisante car les surfaces

utilisées ne sont jamais texturées au pixel près afin que les classes de voxels utilisées dans les méthodes de détection de sources lumineuses ne soient pas composées de voxels complètement disparates.

### C.3.1 BRDF estimée

Les coefficients  $K_{d,R}$ ,  $K_{d,G}$  et  $K_{d,B}$  sont estimés sur chaque voxel pour lesquels des échantillons de luminance existent. Une petite contribution de lumière ambiante est utilisée en fixant  $K_{a,c} = K_{d,c}/100$  sur chaque composante chromatique  $c$ . Ceci évite d'avoir une surface complètement noire lorsqu'elle n'est plus rééclairée. L'utilisation d'un échantillon  $(R, G, B)$  et des coefficients  $K_s$  et  $n$  de la surface permet d'écrire :

$$\left\{ \begin{array}{l} R = L_a \cdot K_{a,R} + \sum_{s=1}^{NbSV} \left( \left( K_{d,R} \cdot \frac{1}{\pi} + K_s \cdot \frac{n+2}{2\pi} \cdot \cos^n \phi_s \right) \cdot L_s \cdot \cos \theta_s \cdot \frac{1}{r_s^2} \right) \\ = K_{d,R} \cdot \frac{L_a}{100} + K_{d,R} \cdot \sum_{s=1}^{NbSV} \left( L_s \cdot \frac{1}{\pi \cdot r_s^2} \cdot \cos \theta_s \right) + K_s \cdot \sum_{s=1}^{NbSV} \left( L_s \cdot \frac{n+2}{2\pi \cdot r_s^2} \cdot \cos^n \phi_s \cdot \cos \theta_s \right) \\ = K_{d,R} \cdot D + K_s \cdot S \\ K_{d,R} = \max\left(\frac{R - K_s \cdot S}{D}, 0\right) \\ K_{d,G} = \max\left(\frac{G - K_s \cdot S}{D}, 0\right) \\ K_{d,B} = \max\left(\frac{B - K_s \cdot S}{D}, 0\right) \end{array} \right. \quad (4.4)$$

Pour réduire l'imprécision, les coefficients d'un voxel sont évalués par une moyenne des coefficients estimés sur tous ses échantillons.

Pour les voxels sans aucun échantillon, les coefficients  $K_{d,R}$ ,  $K_{d,G}$  et  $K_{d,B}$  sont estimés par moyenne des coefficients des voxels voisins 6-connexes qui appartiennent à la même classe de réflectance. L'algorithme effectue plusieurs itérations jusqu'à ne plus modifier aucun voxel. Une moyenne des coefficients  $K_{d,R}$ ,  $K_{d,G}$  et  $K_{d,B}$  est effectuée sur chaque classe de réflectance (moyenne sur les voxels de la classe) afin de déterminer les coefficients de certains voxels qui n'ont pas encore d'estimation de leur BRDF.

### C.3.2 Interpolation utilisée

Afin de réduire les artefacts visibles entre les voxels, nous proposons une interpolation basée sur l'utilisation des voxels voisins 26-connexes qui appartiennent à la même classe de réflectance. Une pondération est donnée sur chaque voxel utilisé. Cette pondération dépend de l'aire de la surface présente dans le voxel et de la distance séparant le barycentre de la surface et le point d'intersection (algorithme 4.1).

### C.3.3 Résultats

La figure 4.3 montre diverses situations pouvant se produire lors du rééclairage d'un objet réel. Les images  $b$ ,  $c$  et  $d$  représentent l'objet rééclairé avec les sources lumineuses précédemment détectées. L'interpolation entre les voxels améliore nettement la qualité du rendu et l'utilisation d'une BRDF par voxel permet d'obtenir des résultats plus réalistes que ceux de la figure 4.2 : l'image  $c$  est proche de la photographie  $a$ . L'utilisation d'une géométrie reconstruite au niveau 8 permet d'obtenir des halos spéculaires plus réalistes mais des défauts apparaissent sur l'estimation des BRDFs : ceci est dû au manque d'échantillons par voxel et aux quelques imprécisions de la reconstruction (cavités non reconstruites). L'image  $f$  montre que le rééclairage en présence

```

// Données déjà définies
R ; // Un rayon traversant un pixel P de l'image synthétisée
V ; // Le voxel intersecté par le rayon R
x ; // Le point d'intersection entre R et un triangle de V
TV ; // Un tableau pour stocker un 26-voisinage de voxels
// Mise à jour pour pouvoir utiliser V
Stocker V dans le tableau TV ;
Calculer d la distance entre le barycentre des triangles de V et x ;
Si la distance d est nulle alors
    // V est le seul voxel utilisé dans le calcul du pixel P
    Utiliser l'équation 4.1 pour évaluer le pixel P ;
Sinon
    Stocker d multiplié par l'aire représentée par les triangles de V dans un tableau TP et dans un variable
    pt, représentant la pondération totale des voxels à utiliser ;
    // Parcours des voxels 26-connexes
    Pour chaque voxel Vj voisin de V faire
        Si Vj et V sont de la même classe de réflectance alors
            // Vj est utilisé dans le calcul du pixel P
            Stocker Vj dans le tableau des voxels à utiliser TV ;
            Stocker d divisé par la distance entre le barycentre des triangles de Vj et x, et multiplié par
            l'aire représentée par les triangles de Vj dans le tableau TP ; et ajouter cette valeur dans la
            variable pt ;
        Fin Si
    Fin Pour
    // Parcours des voxels à utiliser
    Pour chaque voxel Vj du tableau TV faire
        // La division par pt permet d'obtenir une pondération totale de 1
        Diviser la valeur de pondération correspondant au voxel Vj par pt ;
        // Calcul de la contribution du voxel Vj
        Evaluer l'équation 4.1, multiplier par la valeur de pondération du voxel et ajouter le résultat à la
        valeur du pixel ;
    Fin Pour
Fin Si

```

Algorithme 4.1 – Evaluation d'une pondération sur les voxels utilisés pour calculer la luminance en un point  $x$

d'une nouvelle source lumineuse produit de nouvelles images réalistes et l'image  $h$  montre que nous pouvons générer de nouvelles vues de l'objet, autres que les photographies originales.

L'image  $g$ , quand à elle, souligne une incohérence d'éclairément due à des valeurs de coefficients ambiants trop élevés sur les régions initialement non éclairés (zones sombres de l'image  $e$ ). En effet la dépendance entre les coefficients  $K_a$  et  $K_d$ , exprimée dans l'équation 4.4, implique une estimation de coefficients  $K_{d,c}$  très élevés lorsque la surface n'est pas éclairée par les sources lumineuses détectées : ainsi une zone voisine qui n'est subitement plus éclairée paraît trop sombre et un éclairage de la zone initialement non éclairée fait apparaître une zone surexposée à la lumière alors que celle-ci n'est pas définie avec une forte luminance (image  $g$ ). Même en

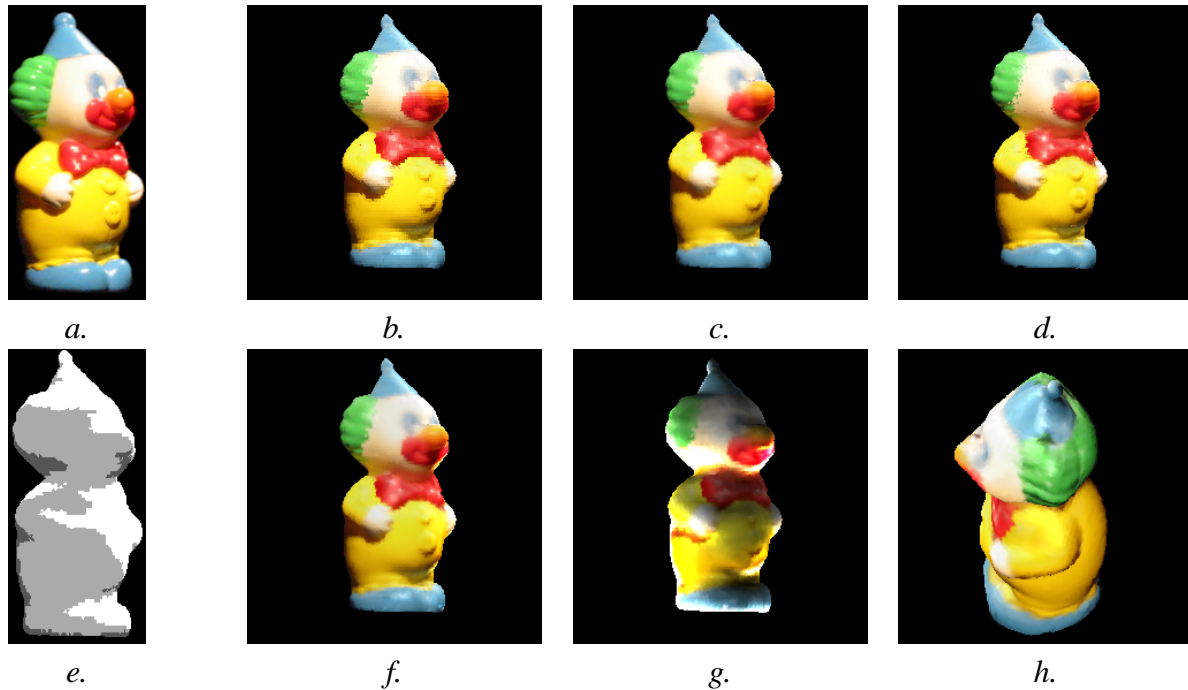


Figure 4.3 – *a. Photo du clown ; b. rééclairage sans interpolation entre les voxels ; c. rééclairage avec interpolation ; d. rééclairage avec un octree de profondeur 8 : les brdfs estimées sont bruitées le long des cheveux et du noeud papillon ; e. visibilité des sources lumineuses détectées : zéro, une ou deux sources visibles ; f. ajout d'une source lumineuse éclairant la joue gauche du clown : le rééclairage semble réaliste ; g. éclairage différent des sources détectées : disparition des halos spéculaires sur le chapeau et les pieds ; h. changement de point de vue pour afficher une vue non disponible sur les photographies originales.*

découplant les valeurs  $K_a$  et  $K_d$  dans l'équation 4.4 et en estimant séparément les voxels éclairés des voxels non éclairés, nous n'obtenons pas un résultat satisfaisant. La zone non éclairée située dans le bas du cou apparaît en fait fortement éclairée (images *a* et *e*) et le seul moyen d'obtenir un éclairage plus réaliste est de prendre en compte les auto-interréflexions. Le bas du cou reflète l'épaule jaune et le noeud papillon rouge : un lancer de rayons récursif ou une simulation d'éclairage permettrait de calculer ces échanges lumineux complexes et sûrement de supprimer ces artefacts.

## C.4 Temps de calcul global

Le tableau 4.6 montre que le temps de précalcul pour reconstruire et analyser les images est inférieur à une heure avec un processeur Xéon cadencé à 2.4 GHz. Le temps de rééclairage, quand à lui, est de trois secondes par image de résolution  $256 \times 256$  : il est linéaire par rapport à la résolution et au nombre moyen de voxels traversés par un rayon (ce nombre double lorsque la profondeur de l'octree augmente de 1). Le coût de l'adjonction de sources lumineuses supplémentaires est également linéaire en fonction du nombre de sources lumineuses puisque nous utilisons un algorithme de lancer de rayons. L'interpolation selon les voxels voisins peut être optimisée en stockant le barycentre et l'aire des triangles dans chaque voxel de la structure.



Clown, objet réel			
Profondeur de l'octree		7	8
<i>Temps d'acquisition</i>	<i>de 8 séries de 72 images</i>	<i>30min / série</i>	
<i>Temps de précalcul</i>	<i>pour la reconstruction géométrique complète</i>	<i>14min30</i>	<i>27min</i>
	<i>pour l'estimation des sources lumineuses et des BRDFs</i>	<i>32min30</i>	<i>6min30</i>
<i>Temps de rééclairage (avec interpolation)</i>	<i>avec une source lumineuse (résolution de l'image : 256 × 256)</i>	<i>3s</i>	<i>6s</i>
	<i>pour chaque source lumineuse additionnelle (résolution de l'image : 256 × 256)</i>	<i>+1.5s</i>	<i>+3s</i>
<i>Nombre de voxels</i>		<i>13 500</i>	<i>54 900</i>
<i>Nombre de triangles</i>		<i>27 000</i>	<i>110 000</i>

Tableau 4.6 – Temps de calcul et taille du maillage pour le processus complet : reconstruction, analyse et rééclairage avec un processeur Xéon cadencé à 2.4 GHz.

Actuellement, nous souhaitons optimiser les calculs pour espérer générer des images en temps interactif. Les récentes avancées du matériel graphique permettent aujourd'hui d'optimiser les accès au tampon de profondeur et de distribuer les calculs sur le processeur de la carte graphique (GPU).

## C.5 Objet synthétique rééclairé

La figure 4.4 montre quelques images du vélo (synthétique) comportant une texture 3D de bois, avec un rééclairage après déplacement d'une des trois sources lumineuses détectées. Sur l'image *c*, une ombre non voulue apparaît sur le guidon : elle provient d'une légère erreur dans la détection de la source lumineuse à l'origine de cette ombre. Les images *d* à *i* montrent différentes positions de la source lumineuse : le rééclairage est cohérent, il y a la présence d'auto-ombrage et de réflexion spéculaire. Néanmoins, le rééclairage semble plus réaliste sur les surfaces texturées que sur les surfaces uniformes. En effet, les défauts de reconstruction des surfaces planes des roues engendrent un affichage de multiples facettes lorsque la caméra est située dans la direction de réflexion spéculaire. En revanche, les bords chanfreinés du vélo sont correctement reconstruits et permettent une nette spécularité sur les arêtes.

## C.6 Objet réel rééclairé

La figure 4.5 montre quelques images du clown (un objet réel en plastique) acquis lors de ces travaux, avec un rééclairage par ajout d'une source lumineuse aux caractéristiques variées. L'image *b* montre que le rééclairage est assez réaliste comparé à la photographie reprojétée sur l'image *a* : néanmoins les halos spéculaires sont plus étalés dans l'image rééclairée. Cela provient sûrement d'une mauvaise estimation de la valeur *n* et surtout d'une reconstruction géométrique imprécise. C'est le cas pour le noeud papillon et pour l'entre-deux pieds. L'image *c* est obtenue en changeant le champ de vue de l'image synthétisée ( focale en position grand angle). Nous



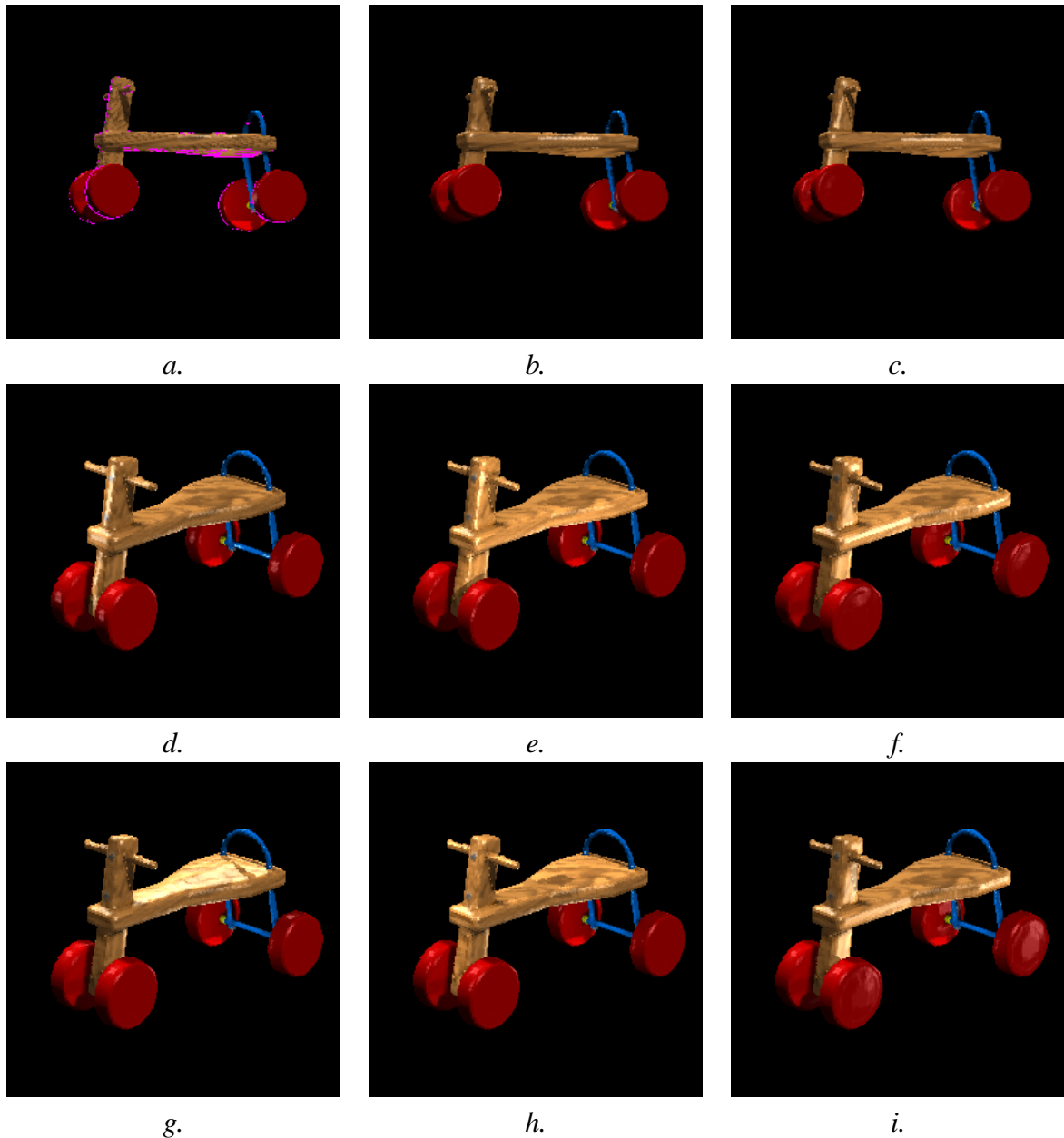


Figure 4.4 – *a. Projection d’une image originale sur le modèle géométrique reconstruit ; b. à i. rééclairage avec déplacement d’une des trois sources lumineuses détectées.*

pouvons remarquer sur les images *d* à *g* que la nouvelle source lumineuse est correctement intégrée et semble éclairer l’objet avec les nuances attendues. Les ombres projetées sont correctes mais nous pouvons de nouveau noter une trop forte illumination pour les surfaces initialement à l’ombre et qui se retrouvent subitement éclairées : c’est un problème difficile à résoudre, un algorithme d’illumination globale permettrait de prendre en compte les auto-interréflexions lors du rééclairage de la surface et lors de l’estimation des coefficients de la BRDF paramétrique.

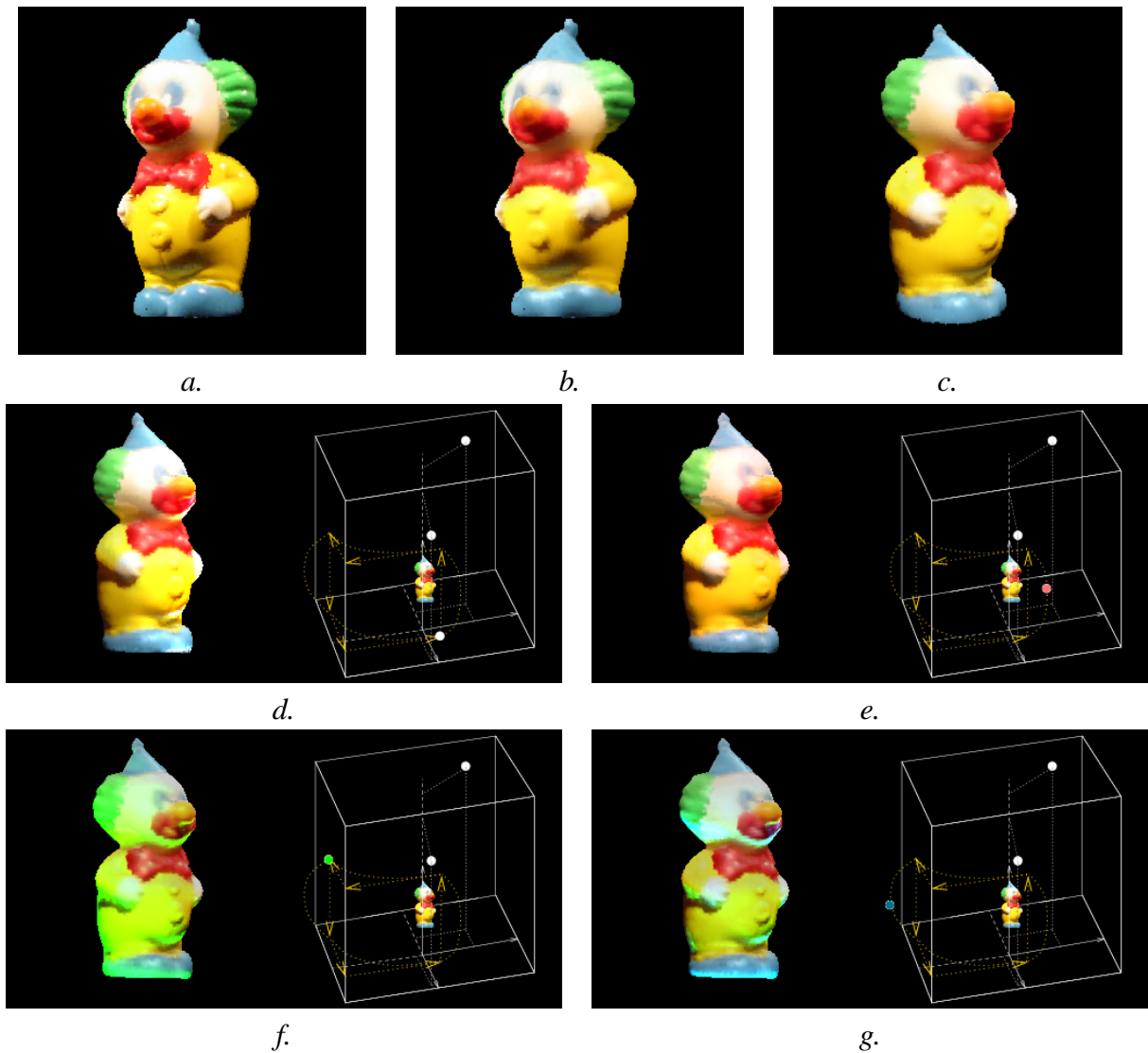


Figure 4.5 – a. Projection d'une photographie sur le modèle géométrique reconstruit ; b. et c. rééclairage avec les sources lumineuses détectées ; d. à g. rééclairage avec atténuation des deux sources lumineuses détectées et ajout d'une troisième source avec variations spatiales et colorimétriques.

---

# CONCLUSION

---

Le rééclairage d'objets réels à partir d'images est un problème difficile à traiter car il implique d'estimer un nombre important de paramètres. Dans la littérature, chaque paramètre est traité de manière indépendante et peu de travaux proposent une méthode complète pour retrouver à la fois la forme de l'objet, la normale de sa surface, la réflectance et les sources lumineuses qui l'éclairent.

Le travail proposé dans ce mémoire décrit une méthodologie de reconstruction automatique de toute la chaîne d'analyse et couvre tous les aspects allant de l'analyse géométrique à l'estimation de sources lumineuses en passant par la réflectance de la surface. Les seules données nécessaires à l'application de cette méthode sont de multiples points de vue de l'objet acquis sous des conditions d'illumination fixes et les paramètres de la caméra. Aucun objet additionnel n'est présent sur les images et la surface de l'objet basé-image peut être diffuse, spéculaire, texturée, etc.

Pour ce travail de thèse, nous avons mis en place un système d'acquisition robuste constitué d'un appareil photographique grand marché et d'un plateau tournant : aucun matériel motorisé n'est utilisé. Cela nous a permis de générer un nombre de vues important d'objets réels afin de construire des lumigraphes. Pour réduire la taille des données sur le disque et en mémoire, nous proposons également un système de compression par quantification validé par une campagne de tests psycho-visuels. La méthode de compression développée a pour objectif de préserver la qualité visuelle des images compressées en adaptant de manière automatique les taux de compression variables au sein de chaque lumigraphe.

La première étape de notre chaîne de traitements consiste à estimer la géométrie de l'objet, c'est-à-dire sa forme et la normale de sa surface. Pour cela, nous proposons de combiner un algorithme de sculpture avec la méthode des *marching cubes* en ajoutant les traitements nécessaires à la construction d'une surface de Jordan. La surface discrète produite est affinée tout en conservant ses propriétés de 6-connexité et de fermeture. L'algorithme des *marching cubes* est adapté en utilisant les pixels des images afin de prendre en compte précisément la forme de l'objet même pour des reconstructions basées-voxel de faible résolution. Le modèle géométrique résultant a la propriété d'être symétrique : il est possible de passer indifféremment du modèle discret au modèle triangulé et inversement.

La seconde partie du traitement permet d'estimer conjointement le type, la position / direction des sources lumineuses ainsi que les propriétés de réflectance de la surface de l'objet. Dans une première phase, le type de surface est analysé par régions de réflectance uniforme. Cela permet d'estimer les sources lumineuses et la réflectance de la surface par région. La précision et la robustesse des traitements reposent sur un raffinement itératif des paramètres par régression non

linéaire. Le résultat de cette étape est une structure de données optimale permettant de représenter toutes les caractéristiques géométriques et photométriques de l'objet (spécularité, textures, etc).

Nous utilisons les données reconstruites pour visualiser les objets avec de nouvelles conditions d'illumination. La version actuelle permet de générer de nouvelles images en quelques secondes, en utilisant l'ensemble des données reconstruites et une méthode de lancer de rayons.

Malgré les résultats encourageants produits par notre méthode, des artefacts sont parfois visibles sur les nouvelles images, provenant de diverses raisons. Pour des objets réels, l'étalonnage des caméras doit être précis au pixel près, c'est-à-dire de l'ordre d'un centième de degré. Même une faible imprécision sur la reconstruction géométrique conduit à des erreurs lors de l'estimation de la réflectance et des conditions d'illumination. Les petites cavités non reconstruites et les auto-interréflexions lumineuses négligées lors de l'estimation de la réflectance sont les sources d'erreur les plus importantes.

Le premier travail à réaliser suite à cette thèse concerne le mécanisme de visualisation et de rééclairage. Nous souhaitons combiner de manière efficace les images des champs de lumière, la géométrie et les informations de réflectance afin d'intégrer des objets basés-image dans un environnement virtuel avec réalisme et de pouvoir visualiser cet environnement de manière réaliste. Pour espérer obtenir un système interactif, une utilisation efficace des processeurs graphiques programmables est prévue.

Nous pensons également pouvoir affiner l'ensemble des paramètres du modèle à partir des données actuellement reconstruites. En effet, les sources lumineuses détectées et les paramètres de réflectance estimés sur toute la surface de l'objet peuvent être utilisés pour guider les méthodes de sculpture par photo-cohérence [KS00]. Ainsi certaines cavités de l'objet peuvent être reconstruites malgré l'apparence de taches spéculaires sur les images. Ces informations peuvent également être utilisées pour raffiner l'estimation de la normale en chaque voxel.

Un calcul d'illumination globale doit pouvoir être envisagé pour obtenir une meilleure estimation des paramètres de BRDFs en prenant en compte les auto-interréflexions lumineuses. Ceci permettrait d'améliorer le rééclairage des objets en supprimant les artefacts visuels sur les surfaces initialement sans éclairage.

Enfin, nous pourrions utiliser un nombre plus varié de points de vue (augmentation de l'angle d'élévation de la caméra) et des images à fortes variations de luminance afin d'estimer des surfaces plus complexes en utilisant un autre modèle de BRDF paramétrique : par exemple, l'estimation de surfaces anisotropiques avec le modèle de Ward comme dans [YDMH99].

# **ANNEXES**



# DÉFINITIONS ET GRANDEURS UTILISÉES

---

## Sommaire

---

A	Les angles solides . . . . .	188
B	Les grandeurs physiques . . . . .	190
C	La fonction de distribution de réflectance bidirectionnelle (BRDF) . . . . .	195
D	L'équation de luminance . . . . .	197

---



Le premier annexe rappelle les notations sur les grandeurs physiques utilisées tout au long de ce mémoire.

## A Les angles solides

L'angle solide est une extension du concept de l'angle à la troisième dimension. En 2D, un angle correspond à la projection de l'arc sur le cercle unité (i.e. longueur de l'arc projeté) et est mesuré en radian (*rad*). Un angle couvrant tout le cercle vaut  $2\pi$  *rad*.

En 3D, l'angle solide  $\omega$  correspond à la projection de l'objet  $S$  sur la sphère unité centrée en  $O$  (i.e. surface de l'objet projeté) et se mesure en stéradian (*sr*) : voir figure 1.1.

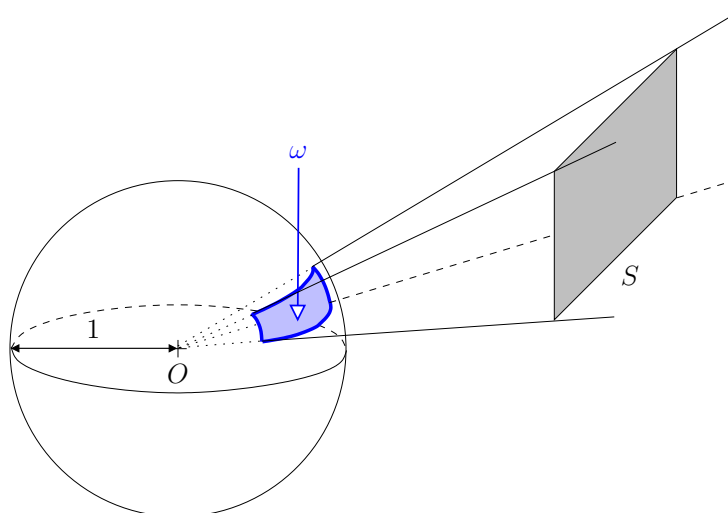


Figure 1.1 – L'évaluation de l'angle solide  $\omega$  formé par un objet  $S$  observé depuis un point  $O$  est effectuée par projection centrale de cet objet sur la sphère unité centrée en  $O$ .

Un angle solide couvrant toute la sphère vaut  $4\pi$  *sr* (aire de la sphère unité).

### L'angle solide élémentaire

Un petit élément d'angle solide, appelé angle solide élémentaire  $d\omega$ , s'exprime sur la sphère unité par des coordonnées sphériques ( $\theta$ ,  $\phi$ ,  $d\theta$  et  $d\phi$ ).

L'expression de l'angle solide élémentaire se déduit donc de la figure 1.2 :

$$d\omega = \sin \theta . d\theta . d\phi \quad (1.1)$$

### Le cône de révolution

L'angle solide  $\omega$  associé au cône de révolution d'angle  $\theta$  (voir figure 1.3) est évalué par l'intégration selon  $d\theta$  et  $d\phi$  de l'expression de l'angle solide élémentaire :

$$\omega = \int_0^{2\pi} \int_0^\theta \sin \theta . d\theta . d\phi = 2\pi(1 - \cos \theta) \quad (1.2)$$

Pour une valeur  $\theta = \pi/2$ , l'angle solide vaut  $2\pi$  *sr* : il représente l'aire de l'hémisphère unité.

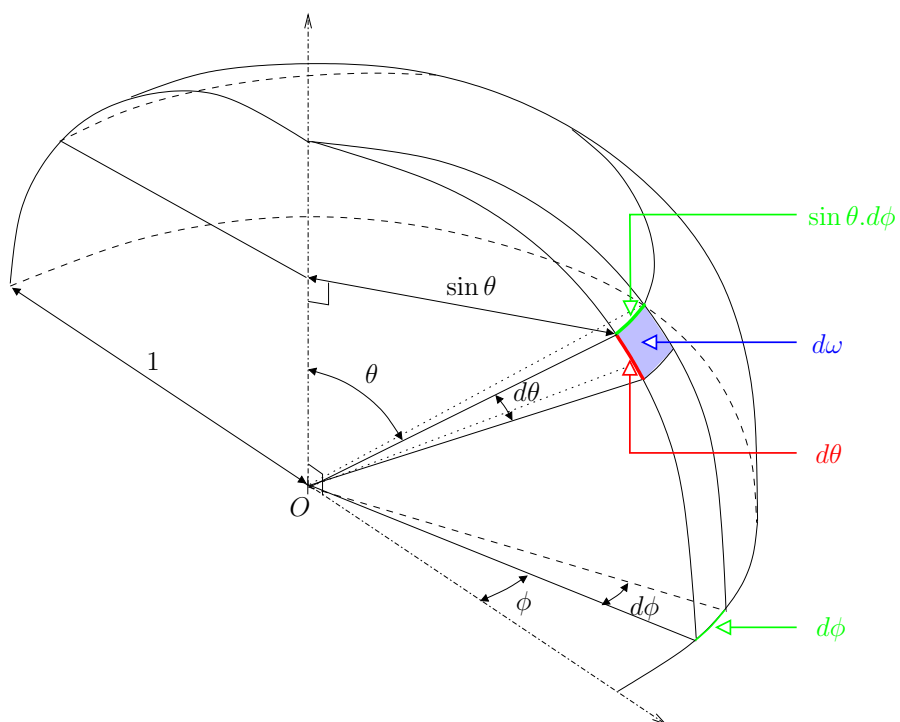


Figure 1.2 – L'angle solide élémentaire  $d\omega$  s'exprime en coordonnées sphériques par  $\sin \theta \cdot d\phi \cdot d\theta$

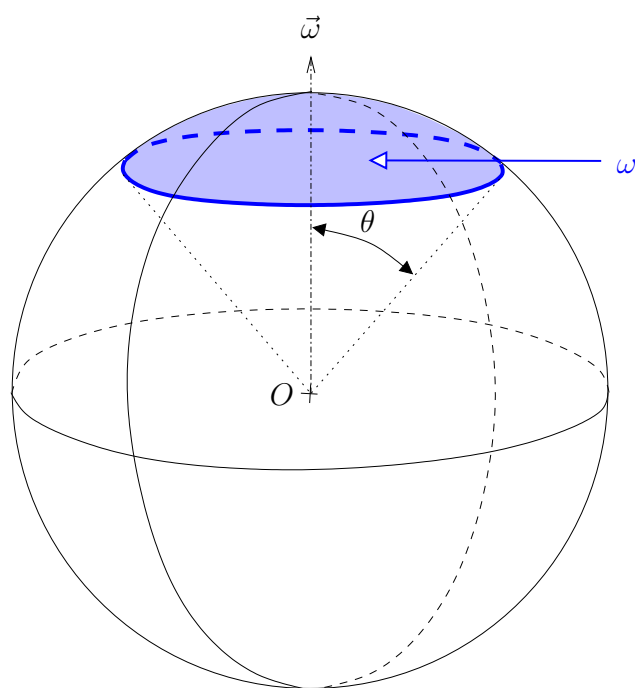


Figure 1.3 – L'angle solide  $\omega$  associé au cône de révolution d'angle  $\theta$  vaut  $2\pi(1 - \cos \theta)$ .

### L'angle solide d'une surface élémentaire

Une surface élémentaire se caractérise par un petit élément de surface  $dS$  centré en un point  $P$ .

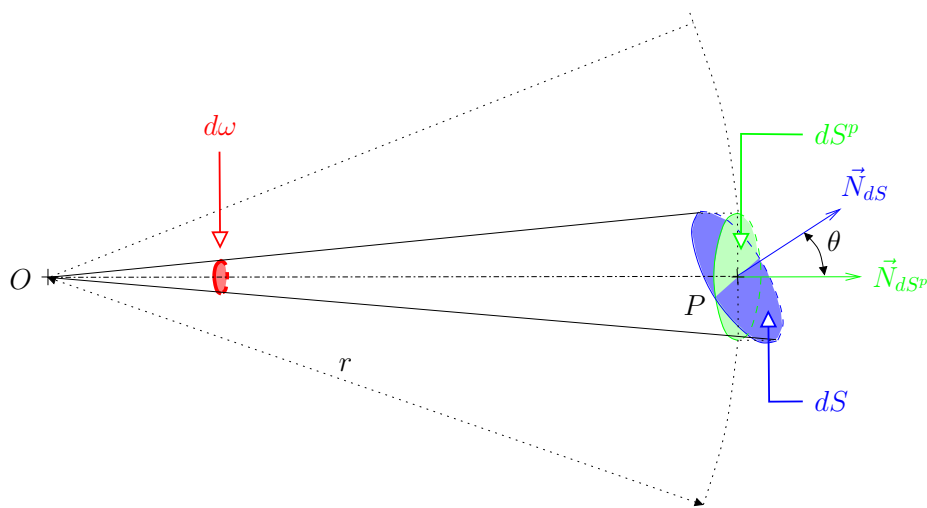


Figure 1.4 – L'angle solide élémentaire  $d\omega$  associé à la surface élémentaire  $dS$  vaut  $dS \cdot \cos \theta / r^2$ .

Pour calculer l'angle solide élémentaire  $d\omega$  associé à la surface élémentaire  $dS$  à partir du point  $O$  situé à une distance  $r$  de  $P$ , la sphère de centre  $O$  et de rayon  $r$  est approchée localement en  $P$  par son plan tangent : voir figure 1.4. Cette hypothèse nécessite que la surface élémentaire soit très petite devant la distance  $r$ .

La projection orthogonale  $dS^p$  de la surface élémentaire  $dS$  sur le plan tangent vaut  $dS \cdot \cos \theta$ . Par conséquent, l'angle solide  $d\omega$  peut s'exprimer par :

$$d\omega = \frac{dS^p}{r^2} = \frac{dS \cdot \cos \theta}{r^2} \quad (1.3)$$

## B Les grandeurs physiques

La radiométrie consiste à étudier les flux d'énergie associés aux rayonnements électromagnétiques. Les flux concernent tout le spectre électromagnétique, allant des rayons gamma (longueurs d'ondes :  $10^{-4}nm$ ) jusqu'aux ondes radio (longueurs d'ondes :  $10^3m$ ).

La photométrie tient compte de la perception visuelle des ondes électromagnétiques : elle est définie uniquement pour la lumière visible, i.e. pour des longueurs d'ondes allant de 380 à 780nm environ.

Les grandeurs photométriques se déduisent des grandeurs radiométriques en pondérant la distribution spectrale du rayonnement par l'efficacité lumineuse relative spectrale de l'oeil humain : voir figure 1.5.

Cette efficacité lumineuse correspond à la réponse moyenne de l'oeil humain, mesurée dans des conditions normales d'éclairage en intérieur.

Les différentes méthodes de rendu s'appuient sur les termes définis en radiométrie pour simuler les échanges d'énergie lumineuse à l'intérieur d'un environnement.

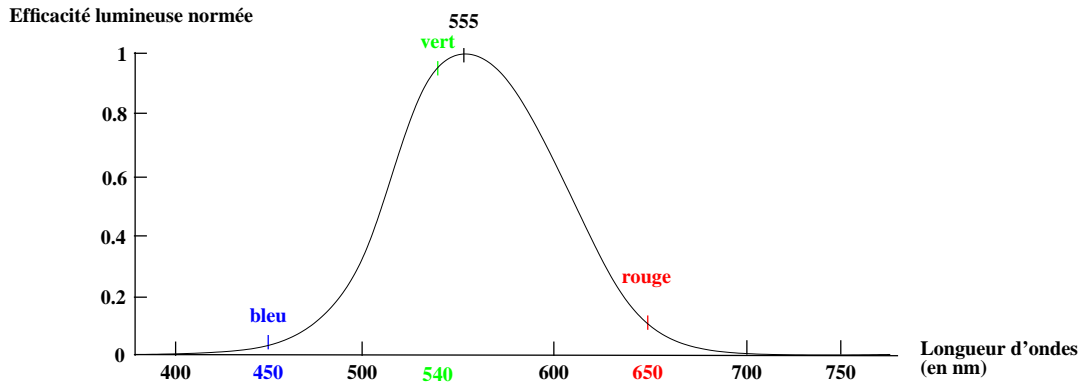


Figure 1.5 – Efficacité lumineuse relative spectrale normée de l’œil humain ; l’efficacité lumineuse est maximale pour une longueur d’onde de 555nm environ ; on peut remarquer que la luminance perçue par l’œil est plus importante pour des longueurs d’ondes proches du vert (540nm) que des longueurs d’ondes proches du rouge (650nm) ou du bleu (450nm).

### L’énergie

L’énergie est l’unité de base utilisée en radiométrie et se mesure toujours en joules ( $J$ ).

#### L’énergie d’un photon

La plus petite particule lumineuse, pouvant être associée à un photon, émet une quantité d’énergie  $e_\lambda$  dépendant de sa longueur d’onde  $\lambda$  :

$$e_\lambda = \frac{h \cdot c}{\lambda} \quad (1.4)$$

où  $h$  est la constante de Planck et  $c$  la vitesse de la lumière.

#### L’énergie spectrale

L’énergie spectrale  $Q_\lambda$  dépend de la densité  $n_\lambda$  de photons de longueur d’ondes  $\lambda$  :

$$Q_\lambda = n_\lambda \cdot e_\lambda \quad (1.5)$$

#### L’énergie lumineuse

L’énergie lumineuse  $Q$  correspond à la somme des énergies de tous les photons à toutes les longueurs d’onde. Dans la littérature, le terme anglais correspondant est *radiant energy*.

$$Q = \int_{380nm}^{780nm} Q_\lambda \cdot d\lambda \quad (1.6)$$

### Le flux

Le flux  $\Phi$ , également appelé la puissance  $P$ , correspond à la quantité d’énergie émise ou reçue par unité de temps, et est mesuré en joules par seconde ( $J \cdot s^{-1}$ ) : unité communément appelée le watt ( $W$ ). En anglais : *radiant flux* ou *radiant power*.

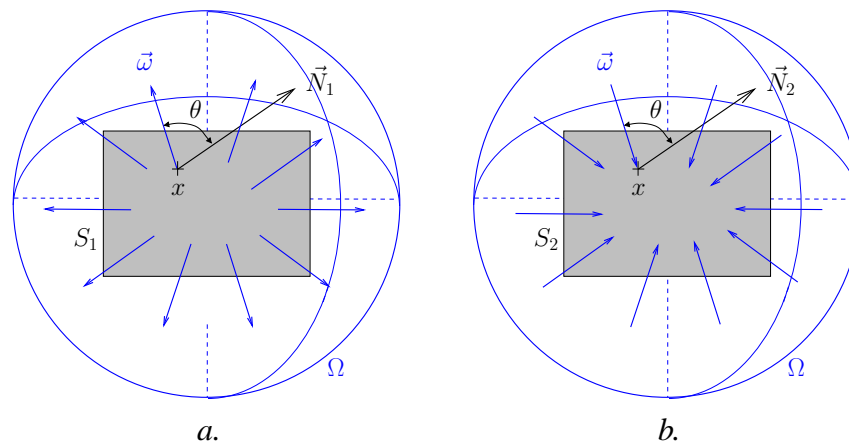


Figure 1.6 – a. Flux  $\Phi_1$  émis par la surface  $S_1$  (vers toutes les directions  $\vec{\omega}$  de l'hémisphère  $\Omega$ ) ; b. flux  $\Phi_2$  reçu par la surface  $S_2$  (depuis toutes les directions  $\vec{\omega}$  de l'hémisphère  $\Omega$ ).

Le flux s'exprime donc comme la dérivée de l'énergie lumineuse en fonction du temps :

$$\Phi = \frac{dQ}{dt} \quad (1.7)$$

### L'intensité

L'intensité  $I$  correspond au flux émis par unité d'angle solide, et est mesurée en watts par stéradian ( $W.sr^{-1}$ ). En anglais : *radiant intensity*.

$$I(\vec{\omega}) = \frac{d\Phi}{d\omega} \quad (1.8)$$

où  $d\omega$  est l'élément d'angle solide dans la direction d'émission  $\vec{\omega}$ .

### La radiosité

La radiosité  $B$  correspond au flux émis par unité de surface, et est mesurée en watts par mètre carré ( $W.m^{-2}$ ). En anglais *radiant exitance* ou *radiosity*.

$$B(x) = \frac{d\Phi}{dS} \quad (1.9)$$

où  $dS$  est l'élément de surface centré au point  $x$  de la surface.

### L'éclairement

L'éclairement  $E$  correspond au flux reçu par unité de surface et est mesuré, comme la radiosité, en watts par mètre carré ( $W.m^{-2}$ ). En anglais : *irradiance*.

$$E(x) = \frac{d\Phi}{dS} \quad (1.10)$$

où  $dS$  est l'élément de surface centré au point  $x$  de la surface.

### La luminance

La luminance  $L$  correspond au flux lumineux émis (ou reçu) par unité d'angle solide et par unité de surface projetée. Elle est mesurée en watts par mètre carré et par stéradian ( $W.m^{-2}.sr^{-1}$ ). En anglais : *radiance*.

$$L(x, \vec{\omega}) = \frac{d^2\Phi}{dS^p.d\omega} = \frac{d^2\Phi}{(\cos\theta.dS).d\omega} \quad (1.11)$$

où  $\theta$  est l'angle formé entre la normale de l'élément de surface  $dS$  centré au point  $x$  et la direction d'émission  $\vec{\omega}$  en ce point,  $dS^p$  est le projeté de l'élément de surface  $dS$  dans la direction d'émission.

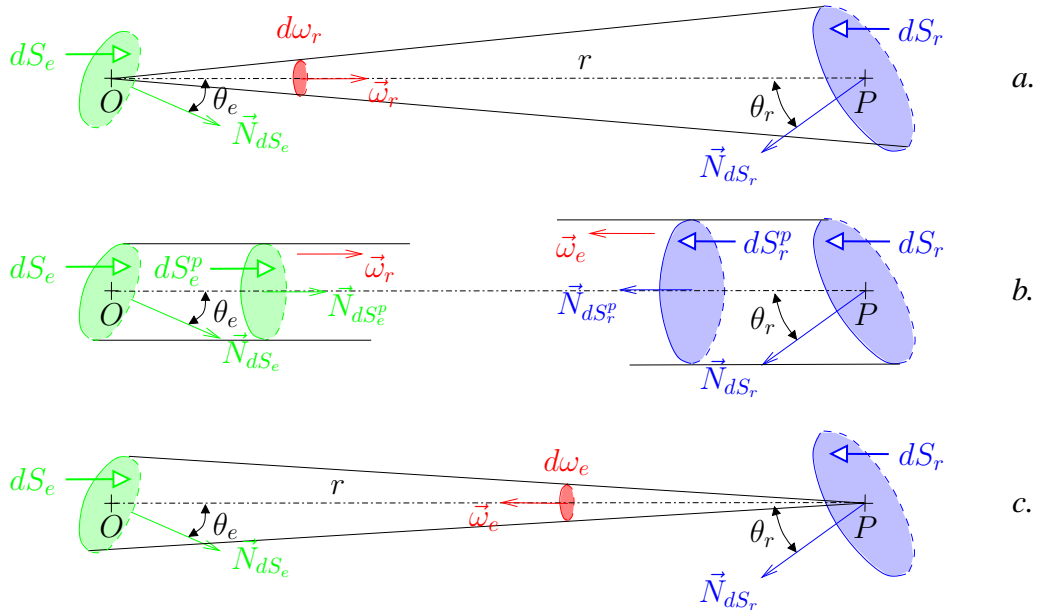


Figure 1.7 – a.  $L_e(O, \vec{\omega}_r)$  est la luminance émise au point  $O$  vers la direction  $\vec{\omega}_r$ ; b.  $dS_e^p$  est le projeté de l'élément de surface  $dS_e$  selon  $\vec{\omega}_r$  ( $dS_e^p = \cos\theta_e.dS_e$ ) et  $dS_r^p$  est le projeté de l'élément de surface  $dS_r$  selon  $\vec{\omega}_e$  ( $dS_r^p = \cos\theta_r.dS_r$ ); c.  $L_i(P, \vec{\omega}_e)$  est la luminance reçue (incidente) au point  $P$  depuis la direction  $\vec{\omega}_e$ .

La figure 1.7 représente un cas d'échanges lumineux entre deux surfaces : soit un émetteur de surface élémentaire  $dS_e$  centré en  $O$  et un récepteur de surface élémentaire  $dS_r$  centré en  $P$ . La luminance émise par l'émetteur en direction du récepteur s'exprime par :

$$L_e(O, \vec{\omega}_r) = \frac{d^2\Phi}{dS_e^p.d\omega_r} = \frac{d^2\Phi}{(\cos\theta_e.dS_e).d\omega_r} \quad (1.12)$$

Et la luminance reçue par le récepteur (luminance incidente) en provenance de l'émetteur s'exprime par :

$$L_i(P, \vec{\omega}_e) = \frac{d^2\Phi}{dS_r^p.d\omega_e} = \frac{d^2\Phi}{(\cos\theta_r.dS_r).d\omega_e} \quad (1.13)$$

### L'invariance de la luminance

D'après la loi de conservation de l'énergie, et en utilisant les notations de la figure 1.7, le flux énergétique émis par un élément de surface  $dS_e$  en direction d'un élément de surface  $dS_r$

est égal au flux reçu par  $dS_r$  en provenance de  $dS_e$  lorsque ces deux éléments de surface sont mutuellement visibles et plongés dans un milieu proche du vide (absence de milieu participant comme les vapeurs, les fumées ou le brouillard). Après ré-écriture,

$$\begin{aligned}
 \Phi(dS_e \rightarrow dS_r) &= \Phi(dS_r \leftarrow dS_e) \\
 \Leftrightarrow L_e(O, \vec{\omega}_r) \cdot \cos \theta_e \cdot dS_e \cdot d\omega_r &= L_i(P, \vec{\omega}_e) \cdot \cos \theta_r \cdot dS_r \cdot d\omega_e \\
 \Leftrightarrow L_e(O, \vec{\omega}_r) \cdot \cos \theta_e \cdot dS_e \cdot \frac{\cos \theta_r \cdot dS_r}{r^2} &= L_i(P, \vec{\omega}_e) \cdot \cos \theta_r \cdot dS_r \cdot \frac{\cos \theta_e \cdot dS_e}{r^2} \\
 \Leftrightarrow L_e(O, \vec{\omega}_r) \cdot \frac{\cos \theta_e \cdot dS_e \cdot \cos \theta_r \cdot dS_r}{r^2} &= L_i(P, \vec{\omega}_e) \cdot \frac{\cos \theta_e \cdot dS_e \cdot \cos \theta_r \cdot dS_r}{r^2}
 \end{aligned} \tag{1.14}$$

nous en déduisons que la luminance émise en  $O$  en direction de  $P$  est égale à la luminance reçue (incidente) en  $P$  en provenance de  $O$  :

$$L_e(O, \vec{\omega}_r) = L_i(P, \vec{\omega}_e) \tag{1.15}$$

La luminance est donc invariante le long de sa direction de propagation. Soit  $r(x, \vec{\omega})$  la fonction de visibilité qui retourne le premier point visible depuis le point  $x$  dans la direction  $\vec{\omega}$  (figure 1.8).

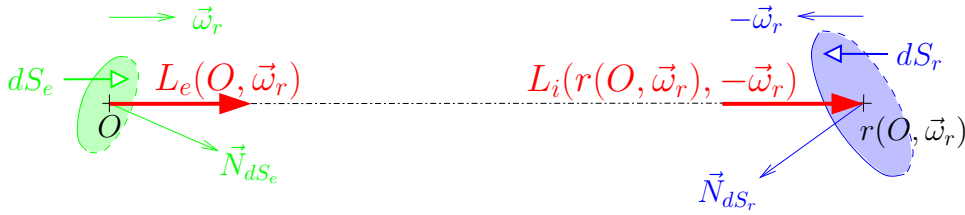


Figure 1.8 – La luminance est invariante le long de sa direction d'émission :  $L_e(O, \vec{\omega}_r) = L_i(r(O, \vec{\omega}_r), -\vec{\omega}_r)$

Avec cette nouvelle notation, l'équation 1.15 peut être réécrite :

$$L_e(O, \vec{\omega}_r) = L_i(r(O, \vec{\omega}_r), -\vec{\omega}_r) \tag{1.16}$$

### Le principe de réciprocité pour le flux

En utilisant toujours les notations de la figure 1.7, si l'émetteur et le récepteur sont échangés (avec une même luminance émise) alors le flux lumineux émis par  $dS_r$  en direction de  $dS_e$  est égal au flux lumineux précédemment émis par  $dS_e$  en direction de  $dS_r$  : c'est le *principe de réciprocité*.

Par conséquent, l'inversion du sens du flux n'a aucun impact sur l'évaluation de la quantité d'énergie transmise entre les deux éléments de surfaces.



### Les principales relations

En supposant, sur l'exemple de la figure 1.6, que l'émission d'énergie lumineuse soit diffuse, ce qui se caractérise par une luminance constante émise par la surface  $S_1$ , le flux lumineux émis par toute la surface et sur toute l'hémisphère peut s'exprimer par :

$$\begin{aligned}
 L_1(x, \vec{\omega}) &= L_1 \\
 B_1(x) &= \int_{\Omega} L_1(x, \vec{\omega}) \cdot \cos\theta(\vec{\omega}) \cdot d\omega \\
 &= L_1 \cdot \int_0^{2\pi} \int_0^{\pi/2} \cos\theta \cdot \sin\theta \cdot d\theta \cdot d\phi \\
 &= L_1 \cdot \pi \\
 \Phi_1 &= \int_{\Omega} \int_{S_1} L_1(x, \vec{\omega}) \cdot \cos\theta(\vec{\omega}) \cdot dS \cdot d\omega \\
 &= \int_{\Omega} I_1(\vec{\omega}) \cdot d\omega \\
 &= \int_{S_1} B_1(x) \cdot dS \\
 &= L_1 \cdot S_1 \cdot \pi
 \end{aligned} \tag{1.17}$$

## C La fonction de distribution de réflectance bidirectionnelle (BRDF)

La fonction de distribution de réflectance bidirectionnelle (FDRB) est définie sur une surface pour caractériser la manière dont un flux lumineux est réfléchi. Cette fonction représente uniquement les *réflexions* avec la surface, elle ne prend pas en compte les éventuelles *réfractions*. Le terme couramment utilisé dans la littérature est *BRDF*, abréviation de *Bidirectional Reflectance Distribution Function*.

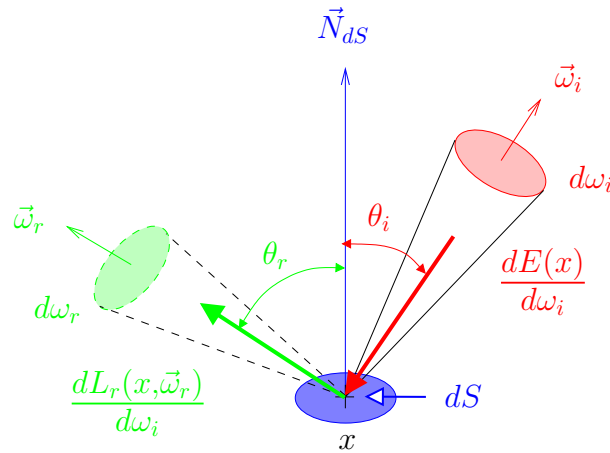


Figure 1.9 – La BRDF exprime le rapport entre la luminance réfléchie  $dL_r/d\omega_i$  et l'éclairement  $dE/d\omega_i$  pour une direction d'incidence  $\vec{\omega}_i$ .

Cette fonction exprime le rapport entre la luminance réfléchie dans une direction  $\vec{\omega}_o$  par un élément de surface  $dS$  centré autour d'un point  $x$  et l'éclairement de  $dS$  pour une direction d'incidence  $\vec{\omega}_i$  (notations sur la figure 1.9). La BRDF, exprimée en  $sr^{-1}$ , est définie par l'équation :

$$f_r(x, \vec{\omega}_r, \vec{\omega}_i) = \frac{dL_r(x, \vec{\omega}_r)/d\omega_i}{dE(x)/d\omega_i} = \frac{dL_r(x, \vec{\omega}_r)/d\omega_i}{L_i(x, \vec{\omega}_i) \cdot \cos\theta_i} = \frac{dL_r(x, \vec{\omega}_r)}{L_i(x, \vec{\omega}_i) \cdot \cos\theta_i \cdot d\omega_i} \tag{1.18}$$

### La réciprocité de Helmholtz

Les propriétés réfléchies d'une surface ne dépendent pas du sens de déplacement de l'énergie lumineuse. Les rôles des directions d'incidence et de réflexion peuvent être échangés, d'où le *principe de réciprocité de Helmholtz* [Hel25], représenté sur la figure 1.10 :

$$f_r(x, \vec{\omega}_r, \vec{\omega}_i) = f_r(x, \vec{\omega}_i, \vec{\omega}_r) \quad (1.19)$$

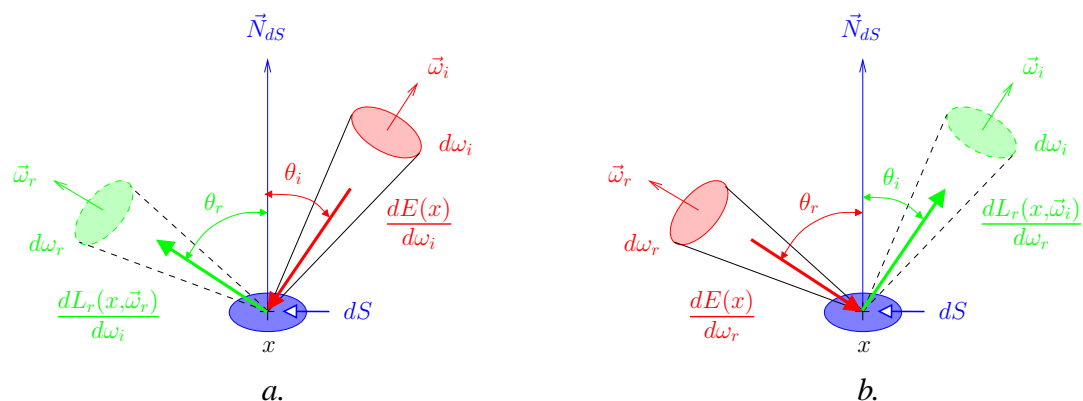


Figure 1.10 – La réciprocité de Helmholtz permet d'écrire :  $f_r(x, \vec{\omega}_r, \vec{\omega}_i) = f_r(x, \vec{\omega}_i, \vec{\omega}_r)$  ; a. représentation de la BRDF  $f_r(x, \vec{\omega}_r, \vec{\omega}_i)$  ; b. représentation de la BRDF  $f_r(x, \vec{\omega}_i, \vec{\omega}_r)$ .

### La réflectivité

La réflectivité exprime le rapport entre la radiosité et l'éclairement d'un élément de surface  $dS$  centré autour d'un point  $x$  pour une direction d'incidence  $\vec{\omega}_i$  (figure 1.11).

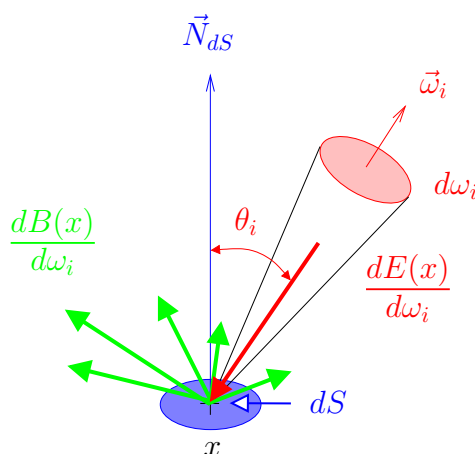


Figure 1.11 – La réflectivité exprime le rapport entre la radiosité  $dB/d\omega_i$  et l'éclairement  $dE/d\omega_i$  pour une direction d'incidence  $\vec{\omega}_i$ .

Ce rapport n'a pas d'unité de mesure, il peut s'exprimer par intégration de la BRDF sur l'ensemble des directions de réflexion :

$$\rho(x, \vec{\omega}_i) = \frac{dB(x)/d\omega_i}{dE(x)/d\omega_i} = \int_{\Omega} \frac{dL_r(x, \vec{\omega}_r) \cdot \cos \theta_r \cdot d\omega_r}{L_i(x, \vec{\omega}_i) \cdot \cos \theta_i \cdot d\omega_i} = \int_{\Omega} f_r(x, \vec{\omega}_r, \vec{\omega}_i) \cdot \cos \theta_r \cdot d\omega_r \quad (1.20)$$

### La conservation de l'énergie

Une surface non émettrice ne peut pas réfléchir plus d'énergie lumineuse qu'elle n'en reçoit. Par conséquent, une BRDF conservatrice d'énergie vérifie l'équation suivante :

$$\rho(x, \vec{\omega}_i) = \int_{\Omega} f_r(x, \vec{\omega}_r, \vec{\omega}_i) \cdot \cos \theta_r \cdot d\omega_r \leq 1, \forall \vec{\omega}_i \quad (1.21)$$

i.e. le pourcentage d'énergie réfléchié dans toutes les directions en provenance d'une même direction d'incidence ne peut dépasser 100%.

Les modèles de BRDF *physiquement corrects* sont des modèles qui respectent les propriétés de *conservation de l'énergie* et de *réciprocité de Helmholtz*. Ces modèles sont étudiés plus en détails dans la section [D.2](#) du chapitre I.

### L'équation de réflectance

La luminance réfléchié par un élément de surface  $dS$  dans une direction de réflexion  $\vec{\omega}_r$  peut s'exprimer en fonction de la BRDF et de la luminance incidente :

$$L_r(x, \vec{\omega}_r) = \int_{\Omega} f_r(x, \vec{\omega}_r, \vec{\omega}_i) \cdot L_i(x, \vec{\omega}_i) \cdot \cos \theta_i \cdot d\omega_i \quad (1.22)$$

## D L'équation de luminance

La luminance totale  $L$  émise par un élément de surface  $dS$  centré en un point  $x$  dans une direction  $\vec{\omega}_r$  est composée de la somme de la luminance auto-émise  $L_e$  (lorsque  $dS$  est un élément de surface d'une source lumineuse) et de la luminance réfléchié  $L_r$  (définie par l'équation [1.22](#)) :

$$\begin{aligned} L(x, \vec{\omega}_r) &= L_e(x, \vec{\omega}_r) + L_r(x, \vec{\omega}_r) \\ &= L_e(x, \vec{\omega}_r) + \int_{\Omega} f_r(x, \vec{\omega}_r, \vec{\omega}_i) \cdot L_i(x, \vec{\omega}_i) \cdot \cos \theta_i \cdot d\omega_i \end{aligned} \quad (1.23)$$

Cette *équation de luminance*, également appelée *équation d'illumination globale*, a été définie par Lafortune en 1996 [[Laf96](#)] : elle se déduit de l'*équation de rendu* définie par Kajiyama en 1986 [[Kaj86](#)] (équation initialement exprimée en termes d'intensité).

En utilisant la propriété d'invariance de la luminance le long d'une direction, définie dans la section [B](#), la luminance totale émise par un point  $x$  de la scène peut s'exprimer en fonction de la luminance totale émise par chaque point de la scène visible depuis le point  $x$  :

$$L(x, \vec{\omega}_r) = L_e(x, \vec{\omega}_r) + \int_{\Omega} f_r(x, \vec{\omega}_r, \vec{\omega}_i) \cdot L(r(x, \vec{\omega}_i), -\vec{\omega}_i) \cdot \cos \theta_i \cdot d\omega_i \quad (1.24)$$



# ORIGINES DU RENDU BASÉ-IMAGE

---

## Sommaire

---

A	Les différents types de textures . . . . .	200
B	Les premiers modèles de Rendu Basé-Image . . . . .	207

---

Les premières méthodes de rendu utilisant des images sont développées à la fin des années 70 avec l'application de textures 2D sur les polygones décrivant un environnement afin de simuler la modélisation de détails géométriques. La définition du terme *placage de texture*, apparaissant pour la première fois dans la thèse de E. Catmull sous le nom de *texture mapping* [Cat74], a permis d'introduire l'utilisation efficace d'images dans le but d'augmenter le réalisme des nouvelles vues synthétisées.

Le modèle géométrique est ensuite progressivement remplacé par des images contenant diverses informations jusqu'à l'apparition des premiers modèles entièrement basés sur des images, permettant la navigation dans des environnements 3D précalculés.

## A Les différents types de textures

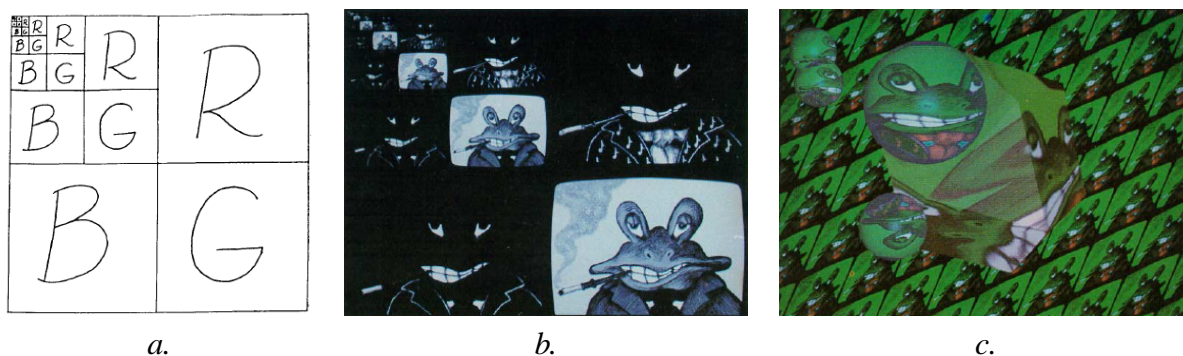
Cette section montre que les textures ont rapidement évolué pour contenir non seulement des informations colorimétriques mais aussi des informations géométriques, d'illumination qui sont facilement utilisables par les cartes graphiques dans le but d'accélérer les temps de rendu.

### Le placage d'images

Le *placage de texture* est la première méthode utilisant des images pour augmenter le réalisme des scènes synthétisées. Le plus souvent, cela consiste à plaquer une image 2D (la texture) sur un polygone afin d'ajouter des détails colorimétriques sur les surfaces sans augmenter la description géométrique de la scène.

Le placage est effectué par une fonction d'interpolation des points de la texture puis une projection des pixels calculés directement sur les coordonnées de l'écran grâce à des opérations matricielles effectuées sur les cartes graphiques. Les textures peuvent être à 1, 2 ou 3 dimensions. Les artefacts visuels comme l'aliassage et les effets de moiré sur les surfaces texturées sont traités par des méthodes de *filtrage* [FLC80, GH86].

### Les textures multi-résolution



La définition de textures à plusieurs niveaux de résolution (*MIP maps*) permet de garder une haute définition de texture pour les surfaces proches de la caméra et d'utiliser des textures de faible résolution pour les surfaces éloignées de la caméra afin d'accélérer les temps de traitement et d'éviter le crénelage / la pixellisation. Ces textures, développées en 1983 par L. Williams [Wil83], sont stockées dans une structure pyramidale facilement compressible (figures 2.1.a et 2.1.b) et sont aujourd'hui directement exploitables par les cartes graphiques. Un filtrage trilineaire permet de supprimer les transitions lors du passage d'une résolution de texture à une autre pour un même polygone (par exemple, lors du déplacement de la caméra). Un filtrage anisotrope permet de lisser les textures vues sur des angles rasants (par exemple, pour les surfaces du sol). De plus, les problèmes récurrents de continuité sur les jointures entre deux polygones adjacents peuvent être traités 2.1.c.

### Les textures procédurales

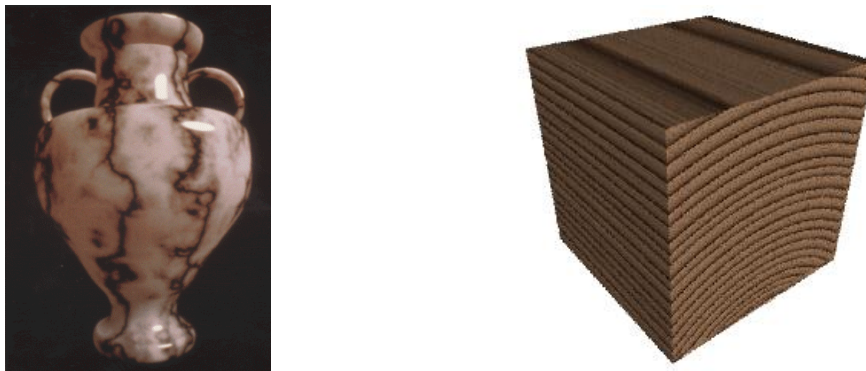


Figure 2.2 – Les textures procédurales permettent un placage continu de la texture sur toute la surface de l'objet. La texture de marbre est présentée dans [Per85] et celle de bois dans [Pea85].

Les textures procédurales sont développées pour permettre la continuité des textures définies sur un même volume et ainsi donner une meilleure impression des matériaux constituant les objets. Ce sont des textures 3D définies par une fonction pseudo-aléatoire de distribution de bruit définissant, par exemple, des textures de marbre [Per85] ou des textures de bois [Pea85] (figure 2.2). Elles sont peu coûteuses en espace de stockage mais requièrent des calculs plus importants lors du rendu des surfaces (le rendu est fait pixel à pixel).

J.-M. Dischler et D. Ghazanfarpour ont travaillé sur la génération automatique de textures procédurales en utilisant une analyse spectrale et spatiale des variations micro-géométriques de la surface [DG97]. Ils analysent les informations provenant de la transformée de Fourier et de l'histogramme afin de réduire les temps de synthèse d'images texturées en définissant chaque texture procédurale comme une somme de fonctions élémentaires aléatoires. Ils proposent également un survol des textures 3D dans [DG01].

### Les cartes de relief / de normale

Cette méthode (*bump mapping* en anglais) est développée par J. Blinn en 1978 [Bli78] pour donner une impression de relief à une surface (simulation de trous et de bosses) sans augmenter pour autant sa description géométrique (la surface est toujours représentée par un polygone).



Une texture contenant des informations sur la direction de la normale est ensuite appliquée sur la surface et utilisée lors des calculs d'illumination : une petite perturbation de la direction de la normale le long de la surface simule ainsi l'effet d'une surface rugueuse (figure 2.3). Une



Figure 2.3 – a. Le bump mapping donne une impression de relief; b. le bump mapping (sphère de gauche) ne permet pas d'afficher correctement la silhouette de l'objet, contrairement au displacement mapping (sphère de droite).

seconde texture, contenant des informations photométriques, peut également être utilisée lors du rendu.

Cette méthode ne permet pas de modifier la silhouette des objets texturés : les contours correspondent à la géométrie initiale et ne peuvent avoir de relief. Les cartes de déplacement, *displacement maps* [Coo84], permettent de résoudre ce problème en perturbant localement la position des points de la surface mais cela augmente de façon significative les calculs d'intersection.

### Les cartes d'environnement / d'illumination

La carte d'illumination est différente des textures traditionnelles car elle n'est pas associée à un objet particulier mais à une sphère imaginaire de rayon infini englobant la scène [Gre86]. En anglais, elle est appelée *illumination mapping* ou *environment mapping*. Des années plus tard, P. Debevec améliore sa capture par un système de sphère miroir appelé *light probe* [Deb98] (plus de détails sont donnés dans la section A). Dans [BN76], J. Blinn et M. Newell définissent une carte de réflexion indexée par la direction du rayon incident pour effectuer les calculs d'illumination des objets de la scène (figure 2.4). Puis dans [MH84], G. Miller et C. Hoffman utilisent une carte d'environnement réel indexée par la direction de la normale de la surface (figure 2.5). Ceci évite le calcul du cosinus de l'angle d'incidence en le stockant directement dans la carte d'environnement (calcul utile pour l'évaluation de la composante de réflexion diffuse). Cette méthode peut être généralisée pour la transparence en indexant la carte par la direction du rayon réfracté [KG79]. Ceci permet de combiner les effets de réflexion et de réfraction sur les surfaces opaques.

Les cartes d'environnement facilitent l'intégration d'environnements lumineux complexes en appliquant directement des textures (la carte d'environnement) sur les polygones à illuminer. Ainsi le calcul d'illumination est indépendant du nombre de sources lumineuses, il dépend de la



a.



b.

Figure 2.4 – Utilisation d’une carte d’environnement par J. Blinn et M. Newell : a. la carte d’environnement d’une pièce fermée ; b. le rendu d’une théière insérée dans la pièce en utilisant la carte d’environnement (images extraites de [BN76]).



a.



b.



c.

Figure 2.5 – Utilisation d’une carte d’environnement réel par G. Miller et C. Hoffman : a. Insertion d’un objet synthétique dans une photographie ; b. la carte d’environnement réel correspondant à la photographie ; c. le rendu de l’objet inséré en utilisant la carte d’environnement (images extraites de [MH84]).

résolution de la carte d’environnement. En contre-partie, ces cartes sont utilisables uniquement dans le cadre d’un éclairage directionnel, où tous les objets de la scène reçoivent le même flux lumineux provenant des mêmes directions d’incidence : pour des environnements lointains comme le ciel, etc.

### Les textures dépendantes du point de vue

P. Debevec et al. ont développé les textures dépendantes du point de vue (*view-dependant texture maps*) afin de générer n’importe quelle vue d’un environnement architectural à partir d’un nombre limité de photographies [DTM96]. Seulement 12 photographies d’un bâtiment sont

utilisées afin de modéliser approximativement sa géométrie (les propriétés architecturales des bâtiments simplifient fortement la modélisation, mais l'intervention de l'utilisateur est tout de même nécessaire). Ensuite, des algorithmes de stéréovision appliqués sur chaque paire de photographies permettent d'éliminer les erreurs importantes sur le modèle géométrique, sans toutefois modéliser les détails : les conditions d'éclairage doivent être les mêmes pour toute la phase d'acquisition (éclairage naturel) et les surfaces de bâtiments doivent être diffuses pour appliquer la stéréovision (voir section B.2.3 du chapitre I pour des détails sur la stéréovision).

Les détails géométriques sont apportés uniquement lors de la visualisation du bâtiment par un plaquage de texture dépendant du point de vue. Chaque photographie est plaquée sur le modèle géométrique. Par conséquent, chaque polygone dispose de plusieurs textures et les textures les plus appropriées sont choisies lors de la génération d'une nouvelle image. La figure 2.6 montre la chaîne complète, de la modélisation au rendu.

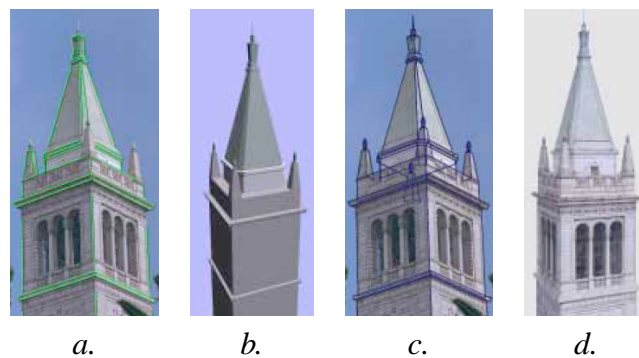


Figure 2.6 – a. Photographie de la tour de Berkeley (les arêtes utilisées pour la modélisation sont représentées en vert); b. modélisation géométrique, après avoir appliqué l'algorithme de stéréovision; c. en bleu, arêtes du modèle géométrique reprojétées sur une photographie; d. nouvelle vue synthétisée (images extraites de [DTM96]).

### *Les textures projectives*

Les textures projectives [DYB98], appelées *projective texture-mapping* en anglais, sont mises en place pour augmenter l'impression de relief sur les surfaces planes et pour accélérer les traitements lors du placage, sur un même polygone, de textures provenant de plusieurs photographies. Cette méthode utilise une structure de données pour organiser les textures de chaque polygone (une carte de vue par polygone). Ainsi, la recherche des 2/3 meilleures vues et l'évaluation de leur pondération sont obtenues plus rapidement sur chaque polygone de la scène. La fusion des différentes vues permet ainsi de synthétiser des images de meilleure qualité et surtout, les textures projectives permettent de limiter les artéfacts visuels entre les différentes images générées : les meilleurs points de vue sont choisis pour un polygone particulier indépendamment des vues choisies pour le reste de la scène.

### *Les cartes de luminance à fortes variations*

Afin d'insérer des objets synthétiques dans les scènes réelles modélisées, P. Debevec capture l'illumination de la scène en plaçant une sphère miroir près de l'endroit d'insertion des objets

[DM97, Deb98]. L'environnement réfléchi par la sphère permet de construire une carte d'environnement similaire à celles présentées dans la section A. Le processus de reconstruction de la carte d'environnement suit le trajet inverse de la lumière pour retrouver les conditions d'illumination directionnelle de la scène. Plusieurs photos sont prises avec des vitesses d'obturation différentes : les faibles valeurs de luminances réfléchies par interrélaxions entre les surfaces ainsi que les fortes valeurs de luminances émises par les sources lumineuses de l'environnement peuvent ainsi être capturées. La carte de luminance est construite sur une grande échelle de valeurs (*high dynamic range radiance map*) après une estimation des réponses des capteurs photographiques.

Lors de l'insertion des objets virtuels, un algorithme d'illumination globale simule les interrélaxions entre les objets virtuels et l'environnement. Pour réduire les temps de calculs, la scène est séparée en deux zones : la scène distante, trop loin de la position d'insertion pour être affectée perceptuellement par les objets insérés, et la scène locale qui au contraire est suffisamment proche pour être significativement affectée par les nouveaux objets. Les luminances émises en direction de la scène distante sont négligées afin d'accélérer les calculs d'illumination. Une estimation des propriétés de réflectance de toutes les surfaces contenues dans la scène locale est également réalisée : cette estimation est étudiée plus en détails dans la section D du chapitre I.

### Les cartes de profondeur

Les cartes de profondeur décrivent une scène par des informations géométriques succinctes (le relief des surfaces) stockées sous forme de textures. Ces cartes de profondeur sont souvent utilisées pour aider la mise en correspondance de pixels provenant de plusieurs points de vue d'une scène et ainsi générer de nouvelles vues par interpolation d'images.

#### *Les déformations d'image*

Les méthodes de déformation d'image, appelées *image warping*, s'appuient sur les informations de profondeur disponibles sur une ou plusieurs images décrivant la scène afin de générer de nouveaux points de vue proches des images disponibles. L'image finale est produite en deux temps : d'abord par projection des pixels d'une image originale sur leur position réelle 3D (en utilisant la carte de profondeur), puis par reprojection des points 3D sur l'image finale. La méthode permet de générer de nouvelles vues réalistes d'un environnement indépendamment de sa complexité géométrique : la description géométrique de la scène est fixe, elle dépend uniquement du nombre d'images de profondeur.

Ces méthodes ont un inconvénient majeur : elles laissent apparaître des trous dans les nouvelles vues visualisées, correspondant à des portions de surfaces non visibles depuis les points de vue originaux mais détectables grâce aux informations de profondeur (les trous sont visibles sur les images de la seconde colonne, figure 2.7). Les trous peuvent également être dûs à la différence d'échantillonnage entre les images originales et les nouvelles vues : si une correspondance est utilisée afin de mettre en relation chaque pixel d'une image originale avec un pixel d'un nouveau point de vue, et que le nouveau point de vue représente les mêmes objets vus de plus près, alors certains pixels de l'image finale n'ont aucune correspondance avec l'image originale.

Pour combler les différences d'échantillonnage, la méthode communément utilisée en *rendu basé-point* est l'écrasement de pixel (*surface splatting*) [ZPBG01]. Cette technique consiste à donner une certaine taille à chaque pixel provenant d'une image originale et ainsi, de les projeter sur plusieurs pixels de l'image finale (chaque pixel est représenté par une surface, une orientation

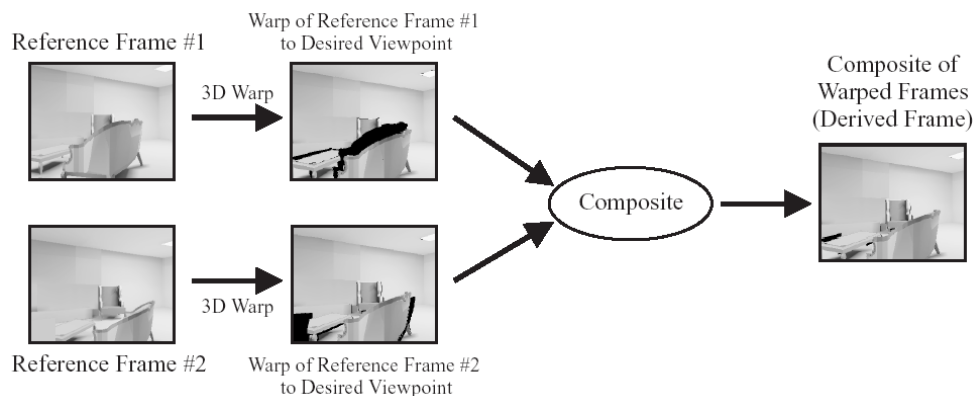


Figure 2.7 – Utilisation de deux images et leur carte de profondeur associée pour régénérer une nouvelle vue (images extraites de [MMB97]).

et des propriétés de transparence). Dans [ZPBG01], cette méthode est également utilisée pour effectuer le rendu d'objets représentés par un nuage de points : le nuage de points provient généralement d'une acquisition géométrique par scanner 3D (voir section B.1 du chapitre I pour des détails sur l'acquisition).

W. Mark et al. proposent une nouvelle méthode appelée *3D warping* [MMB97] afin de combler les manques d'information et de produire de nouvelles vues en temps-réel. Un maillage est construit à partir de la carte de profondeur d'une image d'origine puis est utilisé afin de résoudre les discontinuités dues au sous-échantillonnage. Deux images de référence judicieusement choisies permettent de limiter l'absence d'information dans le nouveau point de vue affiché (image de droite de la figure 2.7). Le choix des vues de référence est effectué en fonction de la position de la caméra lors des deux dernières vues affichées.

### Les cartes de profondeur multi-couche

J. Shade et al. proposent une nouvelle structure, appelée *layered depth image (LDI)*, permettant de représenter une scène par une seule image avec de multiples informations stockées sur chaque pixel [SGHS98]. En effet, chaque pixel de l'image est vu comme un rayon partant du centre de projection de l'image, passant par le pixel et traversant la scène à représenter (figure 2.8.a). En chaque pixel il est possible d'obtenir la liste des points d'intersection (profondeurs) avec les objets de la scène et la valeur colorimétrique correspondant à chaque surface traversée.

La structure permet d'effectuer le rendu de nouveaux points de vue sans faire apparaître de trous dûs aux occlusions de surfaces (figure 2.8.b). L'écrasement de points (*splatting*) est également utilisé sur l'image résultante afin de limiter l'apparition de trous. La structure est essentiellement utilisée pour afficher des objets visualisés sur un angle de caméra restreint. En effet, l'augmentation de l'angle entre la vue synthétisée et la vue utilisée pour le stockage fait apparaître des trous dans l'image produite : sur la figure 2.8a., seulement deux points sont stockés (A et B) alors que le rayon intersecte quatre fois la surface ; lors de la synthèse d'une vue opposée à la caméra LDI, le même rayon est parcouru dans le sens contraire et aucune information ne peut être utilisée.





Figure 2.8 – a. Le pixel traversé par la flèche partant de la caméra dispose de la couleur des points de la surface traversés par cette flèche, à savoir la couleur des points A et B ; b. Rendu d'une poule stockée dans une carte de profondeur multi-couches pour deux points de vues différents (images extraites de [SGHS98]).

## B Les premiers modèles de Rendu Basé-Image

De nouvelles méthodes de rendu sont apparues dans les années 90, permettant la visualisation d'environnement uniquement à partir d'une structure d'images. Ces images sont généralement des vues de la scène prises à des positions bien précises afin de pouvoir déduire toutes les informations nécessaires à la synthèse de nouvelles vues sans construire le modèle géométrique de la scène à visualiser : ce sont les premières applications de Rendu Basé-Image. Suivant la méthode de rendu utilisée et le nombre d'images représentant la scène initiale, des limitations peuvent apparaître sur la liste des points de vues produits.

Ces nouvelles méthodes réduisent de manière drastique la complexité de la description géométrique de la scène. De nouvelles vues d'une scène complexe peuvent être synthétisées en un temps constant : le rendu devient ainsi indépendant de la complexité géométrique de la scène, il dépend uniquement de la résolution de l'image finale produite.

### La navigation 3D précalculée

Les premiers systèmes de navigation développés permettent de se déplacer interactivement dans un environnement 3D mais n'affichent que des séquences d'images préalablement calculées : aucune nouvelle vue n'est synthétisée, l'utilisateur est donc contraint dans ses déplacements à se positionner sur une des vues acquises précédemment.

#### *Les cartes animées*

C'est le premier exemple concret d'utilisation d'images pour la navigation interactive dans une ville. Les cartes animées, appelées *movie-maps*, sont développées en 1980 par A. Lippman pour permettre à l'utilisateur de faire un tour virtuel dans une ville préalablement acquise [Lip80]. L'acquisition des données se fait avec quatre caméras embarquées dans une voiture et quatre photographies sont prises (dans quatre directions orthogonales) tous les dix pieds de distance, dans chacune des rues de la ville. Un capteur optique est utilisé pour évaluer cette distance.

Les photos sont indexées en fonction du plan de la ville et stockées sur des disques laser (ancêtre du DVD). Dans le logiciel de navigation, l'utilisateur se déplace dans la ville grâce à

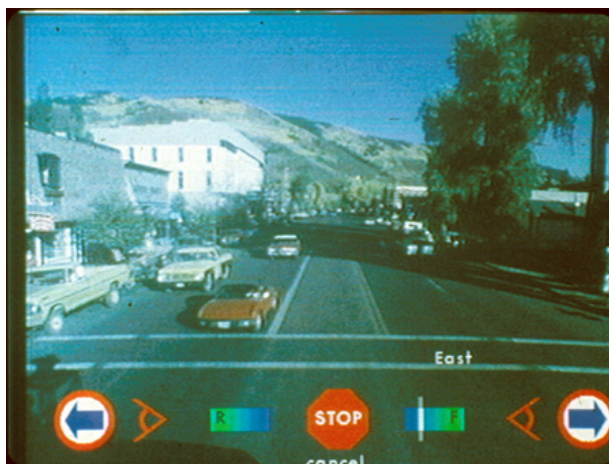


Figure 2.9 – Capture d'écran du logiciel affichant les cartes animées (image extraite du site sur le projet Aspen movie-maps).

des menus apparaissant sur le bas de l'écran (figure 2.9), et les images de la ville s'affichent en suivant le sens de parcours de l'utilisateur.

Aucune nouvelle vue ne peut être générée : l'ordinateur pilote uniquement le lecteur de vidéo-disques, récupère l'image correspondante et l'affiche à l'écran. Toutefois, des informations complémentaires peuvent être affichées sur les façades des bâtiments, comme par exemple, le menu des restaurants.

### *Le musée virtuel*

Le musée virtuel, développé par G. Miller et al. en 1992, est une application permettant de visiter virtuellement un musée [MOC+92]. Des vidéos de déplacement d'une pièce à une autre sont préalablement acquises (une image pour chaque pas lors du déplacement humain) ainsi que des vidéos capturant les objets de collection sur 360 degrés. Ainsi, la visite interactive du musée propose à l'observateur soit de visualiser précisément certains objets en faisant tourner la caméra autour, soit de se déplacer vers une autre pièce en projetant la vidéo de déplacement appropriée. Par conséquent, les vidéos sont directement affichées en fonction des choix de déplacement de l'utilisateur, et aucune image supplémentaire n'est synthétisée.

### **La déformation d'images**

A partir de deux points de vue différents d'un environnement ou d'un objet, les techniques de déformation d'image (*image morphing*) peuvent synthétiser n'importe quelle vue située entre les deux vues originales, sans géométrie additionnelle. Le principal intérêt de ces méthodes est la production d'un ensemble d'images représentant une transition lisse entre les deux vues originales (transition sans changement brusque sur la forme ou la texture des objets composant l'environnement visualisé).

### *L'interpolation de vue*

L'interpolation de vue (*view interpolation*), développée par S. Chen et L. Williams en 1993, permet de générer interactivement n'importe quelle vue entre deux images de référence proches



l'une de l'autre [CW93]. A partir des matrices de transformation de la caméra et des cartes de profondeur de l'ensemble des images acquises, un prétraitement sur toutes les images calcule de manière automatique une carte de correspondance entre chaque paire de vues proches l'une de l'autre.

Ainsi, la génération d'une nouvelle vue dépend uniquement de la résolution de l'image produite et s'effectue en temps constant par interpolation bilinéaire entre les deux cartes de correspondance les plus proches du nouveau point de vue.

### *La déformation de vue*

La déformation de vue (*view morphing*), proposée par S. Seitz et C. Dyer en 1996, est une méthode permettant de calculer des transitions plus réalistes entre deux vues originales [SD96] en utilisant la géométrie projective (figure 2.10). Pour construire les transitions entre les deux vues originales, celles-ci doivent être projetées sur un plan orthogonal à la droite épipolaire passant par le centre optique des deux caméras.

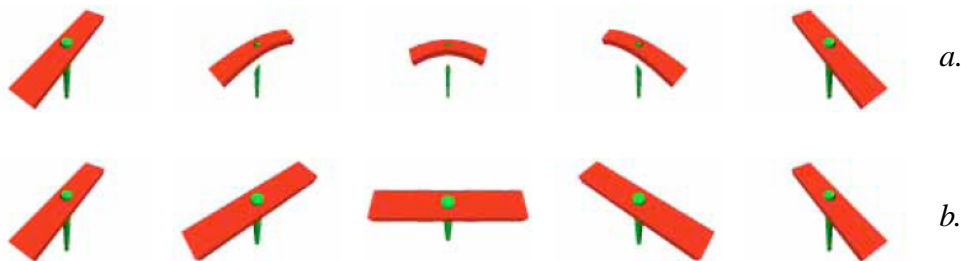


Figure 2.10 – Les 3 vues centrales ont été générées par interpolation des vues sur les bords ; a. avec la méthode de déformation d'image : des déformations géométriques et des discontinuités apparaissent ; b. avec la méthode de déformation de vue : conservation de la forme géométrique grâce à la projection des vues originales sur un même plan avant l'interpolation (images extraites de [SD96]).

Contrairement à l'interpolation de vue, la déformation de vue ne requiert aucune information sur la géométrie de la scène (carte de profondeur non utilisée) mais la carte de correspondance utilisée lors de la déformation d'images requiert l'intervention humaine lors de sa génération.

### **La géométrie projective**

En 1994, S. Laveau et O. Faugeras synthétisent de nouvelles vues d'une scène représentée uniquement par une collection d'images [LF94]. Les images de référence doivent être fortement calibrées (connaissance des paramètres intrinsèques et extrinsèques de la caméra) et des relations de géométrie épipolaire sont utilisées entre les différentes images. Quelques points de correspondance sont définis manuellement entre les paires d'images de référence afin de construire les matrices fondamentales. A partir de ces matrices, chaque pixel d'une image de référence est exprimé dans la base projective de l'image finale. Ainsi, les transformations projectives permettent de mettre en correspondance les pixels des images de référence et de l'image produite sans passer par l'évaluation de coordonnées 3D ou d'un quelconque modèle géométrique de la scène.

### La fonction plénoptique

En 1991, E. Adelson et J. Bergen définissent la *fonction plénoptique* : définie sur 7 paramètres, elle permet de représenter tout échange de flux lumineux à l'intérieur d'un environnement [AB91]. Depuis toutes les méthodes de Rendu Basé-Image sont définies autour de cette fonction, elles échantillonnent le plus fidèlement possible la fonction plénoptique. Plus de détails sont disponibles dans la section A du chapitre I.

---

# BIBLIOGRAPHIE

---

- [AB91] E. H. Adelson and J. R. Bergen. *The Plenoptic Function and the Elements of Early Vision*, chapter 1. Computational Models of Visual Processing, MIT Press, 1991. — cité p. 6, 7, 16, 210
- [BB04] C. Bouganis and M. Brookes. Multiple light source detection. *IEEE PAMI*, 26(4) :509–514, 2004. — cité p. 33
- [BH85] M. J. Brooks and B. K. P. Horn. Shape and source from shading. Technical report, Artificial Intelligence Laboratory, Massachusetts Institute of Technology (MIT), 1985. — cité p. 29
- [Bli78] J. F. Blinn. Simulation of wrinkled surfaces. In *ACM SIGGRAPH Computer Graphics*, volume 12, pages 286–292, 1978. — cité p. 201
- [BMR<sup>+</sup>99] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE TVCG*, 5(4) :349–359, 1999. — cité p. 18
- [BN76] J. F. Blinn and M. E. Newell. Texture and reflection in computer generated images. *Communications of the ACM*, 19(10) :542–547, 1976. — cité p. 202, 203
- [BP92] M. Bichsel and A. P. Pentland. A simple algorithm for shape from shading. In *IEEE CVPR*, pages 459–465, 1992. — cité p. 22
- [CA86] C. H. Chien and J. K. Aggarwal. Volume/surface octrees for the representation of three-dimensional objects. *CVGIP*, 36(1) :100–113, 1986. — cité p. 19
- [Cal04] P. Callet. Rendering of binary alloys. In *ICCVG*, pages 469–476, 2004. — cité p. 1
- [Cat74] E. E. Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces*. PhD thesis, Department of CS, University of Utah, 1974. — cité p. 200
- [CBCG02] W.-C. Chen, J.-Y. Bouguet, M. H. Chu, and R. Grzeszczuk. Light field mapping : Efficient representation and hardware rendering of surface light fields. In *ACM SIGGRAPH Computer Graphics*, pages 447–456, 2002. — cité p. 15
- [CBG94] W. Chojnacki, M. J. Brooks, and D. Gibbins. Revisiting pentland’s estimator of light source direction. *JOSA*, 11(1) :118–124, 1994. — cité p. 29
- [Che95] S. E. Chen. Quicktime vr - an image-based approach to virtual environment navigation. In *ACM SIGGRAPH Computer Graphics*, volume 29, pages 29–38, 1995. — cité p. 7
- [CHLG05] G. Coombe, C. Hantak, A. Lastra, and R. Grzeszczuk. Online construction of surface light fields. In *EGSR*, pages 83–90, 2005. — cité p. 67

- [CJ91] W. Chen and B. Jiang. 3d camera calibration using vanishing points concept. In *IEEE CVPR*, volume 24, pages 57–67, 1991. — cité p. 25
- [CL96] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *ACM SIGGRAPH Computer Graphics*, volume 30, pages 303–312, 1996. — cité p. 18, 19
- [CMS99] W. B. Culbertson, T. Malzbender, and G. G. Slabaugh. Generalized voxel coloring. In *IEEE ICCV*, pages 100–115, 1999. — cité p. 21
- [Coo84] R. L. Cook. Shade trees. In *ACM SIGGRAPH Computer Graphics*, volume 18, pages 223–231, 1984. — cité p. 202
- [CT82] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM TOG*, 1(1) :7–24, January 1982. — cité p. 42
- [CTCS00] J-X. Chai, X. Tong, S-C. Chan, and H-Y. Shum. Plenoptic sampling. In *ACM SIGGRAPH Computer Graphics*, pages 307–318, 2000. — cité p. 13
- [CW93] S. E. Chen and L. Williams. View interpolation for image synthesis. In *ACM SIGGRAPH Computer Graphics*, volume 27, pages 279–288, 1993. — cité p. 209
- [CZ00] G. Cross and A. Zisserman. Surface reconstruction from multiple views using apparent contours and surface texture. In *NATO Advanced Research Workshop on Confluence of Computer Vision and Computer Graphics*, pages 25–47, 2000. — cité p. 26
- [DC01] J. Davis and X. Chen. A laser range scanner designed for minimum calibration complexity. *3DIM*, page 91, 2001. — cité p. 18
- [Deb98] P. E. Debevec. Rendering synthetic objects into real scenes : Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *ACM SIGGRAPH Computer Graphics*, volume 32, pages 189–198, 1998. — cité p. 33, 58, 60, 202, 205
- [Deb04] P. Debevec. Digitizing the parthenon : Estimating surface reflectance properties of a complex scene under captured natural illumination. In *Vision, Modeling and Visualization*, pages 99–109, 2004. — cité p. 1, 48
- [DG97] J.-M. Dischler and D. Ghazanfarpour. A procedural description of geometric textures by spectral and spatial analysis of profiles. *Computer Graphics Forum*, 16(3) :129–140, 1997. — cité p. 201
- [DG01] J.-M. Dischler and D. Ghazanfarpour. A survey of 3d texturing. *Computers and Graphics*, 25(1) :135–151, 2001. — cité p. 201
- [DHT<sup>+</sup>00] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *ACM SIGGRAPH Computer Graphics*, pages 145–156, 2000. — cité p. 16, 45, 46
- [DM97] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH Computer Graphics*, volume 31, pages 369–378, 1997. — cité p. 33, 143, 205
- [DTM96] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs : A hybrid geometry- and image-based approach. In *ACM SIGGRAPH Computer Graphics*, volume 30, pages 11–20, 1996. — cité p. 203, 204

- [DYB98] P. Debevec, Y. Yu, and G. Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. In *Eurographics Workshop on Rendering*, 1998. — cité p. 204
- [ES03] C. H. Esteban and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. In *3DIM*, pages 46–53, 2003. — cité p. 26
- [ESG99] P. Eisert, E. Steinbach, and B. Girod. 3-d shape reconstruction from light fields using voxel back-projection. In *Vision, Modeling and Visualization*, pages 67–74, 1999. — cité p. 26
- [FB95] T. Fromherz and M. Bichsel. Shape from multiple cues : Integrating local brightness information. *ICYCS*, 1995. — cité p. 20
- [FK98] O. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In *ECCV*, volume 1, pages 379–393, 1998. — cité p. 21, 22
- [FL95] P. Fua and Y. G. Leclerc. Object-centered surface reconstruction : Combining multi-image stereo and shading. *IJCV*, 16 :35–56, 1995. — cité p. 26
- [FLC80] E. A. Feibush, M. Levoy, and R. L. Cook. Synthetic texturing using digital filters. In *ACM SIGGRAPH Computer Graphics*, volume 14, pages 294–301, 1980. — cité p. 200
- [Foo97] S.C. Foo. *A Gonioreflectometer for Measuring the Bidirectional Reflectance of Material for Use in Illumination Computation*. PhD thesis, Cornell University, 1997. — cité p. 44
- [Fou94] Alain Fournier. Illumination problems in computer graphics. In *Journée Analyse/Synthèse d'Images (JASI)*, 1994. — cité p. 31
- [Fua93] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. In *Machine Vision and Applications*, volume 6, pages 35–49, 1993. — cité p. 21
- [Fun03] N. Funk. A survey of light source detection methods. Rapports disponibles sur <http://www.cs.ualberta.ca/~funk/>, University of Alberta, Canada, 2003. — cité p. 37
- [Gag94] A. Gagalowicz. Towards a vision system for a domestic robot. In *Journée Analyse/Synthèse d'Images (JASI)*, 1994. — cité p. 31
- [GG92] A. Gersho and R. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic, Boston, 1992. — cité p. 168
- [GGSC96] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *ACM SIGGRAPH Computer Graphics*, volume 30, pages 43–54, 1996. — cité p. 9, 11, 14, 25, 61, 76, 167
- [GH86] N. Greene and P. S. Heckbert. Creating raster omnimax images from multiple perspective views using the elliptical weighted average filter. *IEEE Computer Graphics and Applications*, 6 :21–27, 1986. — cité p. 200
- [Gou71] H. Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers*, C-20(6) :623–629, 1971. — cité p. 54
- [Gre86] N. Greene. Environment mapping and other applications of world projections. In *Graphics Interface*, pages 108–114, 1986. — cité p. 202

- [Gui00] E. Guillou. *Simulation d'environnements complexes non lambertiens à partir d'images : Application à la réalité augmentée*. PhD thesis, IRISA, Université de Rennes 1, 2000. — cité p. 25, 32, 61, 126
- [GW94] A. Van Gelder and J. Wilhelms. Topological considerations in isosurface generation. *ACM TOG*, 13(4) :337–375, 1994. — cité p. 97, 98
- [HA93] D. R. Hougen and N. Ahuja. Estimation of the light source distribution and its use in integrated shape recovery from stereo and shading. In *IEEE ICCV*, pages 148–155, 1993. — cité p. 31
- [HED05] T. Hawkins, P. Einarsson, and P. Debevec. A dual light stage. In *EGSR*, pages 91–98, 2005. — cité p. 46
- [Hel25] H. Von Helmholtz. *Trzatisse on Physiological Optics*. Dover, New York, 1925. — cité p. 196
- [Her92] G. T. Herman. Discrete multidimensional jordan surfaces. In *CVGIP*, pages 507–515, 1992. — cité p. 85
- [HLGB03] J. M. Hasenfratz, M. Lapiere, J.-D. Gascuel, and E. Boyer. Real-time capture, reconstruction and insertion into virtual world of human actors. In *Vision, Video and Graphics*, pages 49–56, 2003. — cité p. 20, 76, 94
- [HNI05] K. Hara, K. Nishino, and K. Ikeuchi. Light source position and reflectance estimation from a single view without the distant illumination assumption. *IEEE PAMI*, 27(4) :493–505, 2005. — cité p. 48, 155
- [HSKS99] W. Heidrich, H. Schirmacher, H. Kück, and H.P. Seidel. A warping-based refinement of lumigraphs. *WSCG*, pages 102–109, 1999. — cité p. 13
- [HTSG91] X. D. He, K. E. Torrance, F. X. Sillion, and D. P. Greenberg. A comprehensive physical model for light reflection. In *ACM SIGGRAPH Computer Graphics*, volume 25, pages 175–186, 1991. — cité p. 44
- [IMG00] A. Isaksen, L. McMillan, and S. Gortler. Dynamically reparameterized light fields. In *ACM SIGGRAPH Computer Graphics*, pages 297–306, 2000. — cité p. 14
- [IS91] K. Ikeuchi and K. Sato. Determining reflectance properties of an object using range and brightness images. *IEEE PAMI*, 13(11) :1139–1153, 1991. — cité p. 48
- [Kaj86] J. T. Kajiya. The rendering equation. In *ACM SIGGRAPH Computer Graphics*, pages 143–150, 1986. — cité p. 197
- [KG79] D. S. Kay and D. P. Greenberg. Transparency for computer synthesized images. In *ACM SIGGRAPH Computer Graphics*, volume 13, pages 158–164, 1979. — cité p. 202
- [KS00] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *IJCV*, 38(3) :199–218, 2000. — cité p. 20, 184
- [Laf96] E. P. Lafortune. *Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering*. PhD thesis, Department of Computer Science, Katholieke Universiteit, 1996. — cité p. 197
- [Lam60] J. H. Lambert. *Photometria sive de mensura de gratibus luminis, colorum umbrae*. Eberhard Klett, 1760. — cité p. 40
- [LBG80] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. In *IEEE Trans. on Communications*, volume 28, pages 84–95, 1980. — cité p. 170

- [LC87] W. E. Lorensen and H. E. Cline. Marching cubes : A high resolution 3d surface construction algorithm. In *ACM SIGGRAPH Computer Graphics*, volume 21, pages 163–169, 1987. — cité p. 11, 20, 76, 85, 93, 94, 97, 98
- [Ler06] R. Lerbour. Compression de lumigraphes. Mémoire de stage recherche m2, Université de Poitiers, 2006. — cité p. 171
- [Lew94] R. Lewis. Making shaders more physically plausible. *Computer Graphics Forum*, 13(2), 1994. — cité p. 41, 131
- [LF94] S. Laveau and O. Faugeras. 3-d scene representation as a collection of images and fundamental matrices. In *ICPR*, pages 689–691, 1994. — cité p. 209
- [LFTG97] E. Lafortune, S.-C. Foo, K. E. Torrance, and D. P. Greenberg. Non-linear approximation of reflectance functions. In *ACM SIGGRAPH Computer Graphics*, volume 31, pages 117–126, 1997. — cité p. 43
- [LFTW06] H. Li, S.C. Foo, K. Torrance, and S. Westin. Automated three-axis gonioreflectometer for computer graphics applications. *Optical Engineering*, 45(4), 2006. — cité p. 44, 45, 52
- [LH96] M. Levoy and P. Hanrahan. Light field rendering. In *ACM SIGGRAPH Computer Graphics*, volume 30, pages 31–42, 1996. — cité p. 9, 10, 15, 17, 65, 67, 75, 169, 170, 171
- [Lip80] A. Lippman. Movie-maps : An application of the optical videodisc to computer graphics. In *ACM SIGGRAPH Computer Graphics*, 1980. — cité p. 207
- [LK93] K. M. Lee and C. Kuo. Shape from shading with a linear triangular element surface model. *IEEE PAMI*, 15 :815–822, 1993. — cité p. 22
- [LKG<sup>+</sup>03] H. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.P. Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM TOG*, 22(2) :234–257, 2003. — cité p. 49, 55, 136, 176
- [LLS03] Y. Li, S. Lin, H. Lu, and H. Shum. Multiple-cue illumination estimation in textured scenes. In *IEEE ICCV*, pages 1366–1373, 2003. — cité p. 34
- [Llo82] S. P. Lloyd. Least squares quantization in pcm. In *IEEE Trans. Inform. Theory*, volume 28, pages 129–137, 1982. — cité p. 170
- [LPC<sup>+</sup>00] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital michelangelo project : 3D scanning of large statues. In *ACM SIGGRAPH Computer Graphics*, pages 131–144, 2000. — cité p. 1, 18
- [LR85] C. H. Lee and A. Rosenfeld. Improved methods of estimating shape from shading using the light source coordinate system. *Artificial Intelligence*, 26 :125–143, 1985. — cité p. 22
- [LV03] J.-O. Lachaud and A. Vialard. Geometric measures on arbitrary dimensional digital surfaces. In *DGCI*, pages 434–443, 2003. — cité p. 24, 111, 113, 119
- [LW94] E. P. Lafortune and Y. D. Willems. Using the modified phong reflectance model for physically based rendering. Technical Report CW 197, Department of Computing Science, K.U. Leuven, 1994. — cité p. 41, 131
- [MA83] W. N. Martin and J. K. Aggarwal. Volumetric descriptions of objects from multiple views. *IEEE PAMI*, 5(2) :150–158, 1983. — cité p. 19



- [Mar03] C. Marchessoux. *Système d'acquisition 3D pour le pied humain*. PhD thesis, Laboratoire SIC, Université de Poitiers, 2003. — cité p. 76
- [MB95] L. McMillan and G. Bishop. Plenoptic modeling : An image-based rendering system. In *ACM SIGGRAPH Computer Graphics*, volume 29, pages 39–46, 1995. — cité p. 8, 9
- [MEM01] T. H. Meyer, M. Eriksson, and R. C. Maggio. Gradient estimation from irregularly spaced data sets. *Mathematical Geology*, 33 :693–717, 2001. — cité p. 23
- [Mer02] B. Mercier. Extraction de sources lumineuses en rendu basé-image. Master's thesis, Laboratoire SIC, Université de Poitiers, 2002. — cité p. 146
- [MFK99] Y. Matsumoto, K. Fujimura, and T. Kitamura. Shape-from-silhouette/stereo and its application to 3-d digitizer. In *DGCI*, pages 177–190, 1999. — cité p. 26
- [MGW01] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. In *ACM SIGGRAPH Computer Graphics*, pages 519–528, 2001. — cité p. 55, 56
- [MH84] G. S. Miller and C. R. Hoffman. Illumination and reflection maps : Simulated objects in simulated and real environments. In *ACM SIGGRAPH Computer Graphics*, 1984. — cité p. 202, 203
- [MLH02] D. McAllister, A. Lastra, and W. Heidrich. Efficient rendering of spatial bidirectional reflectance distribution functions. In *Graphics Hardware*, pages 79–88, 2002. — cité p. 55, 168, 173
- [MM04] B. Mercier and D. Meneveaux. Joint estimation of multiple light sources and reflectance from images. In *ICCVG*, pages 66–71, 2004. — cité p. 156
- [MM05] B. Mercier and D. Meneveaux. Shape from silhouette : Image pixels for marching cubes. *Journal of WSCG*, 13 :112–118, 2005. — cité p. 114
- [MMB97] W. R. Mark, L. McMillan, and G. Bishop. Post-rendering 3d warping. In *Symposium on Interactive 3D graphics*, pages 7–16, 1997. — cité p. 206
- [MMF06] B. Mercier, D. Meneveaux, and A. Fournier. A framework for automatically recovering object shape, reflectance and light sources from calibrated images. *IJCV*, 2006. — cité p. 156
- [MMS<sup>+</sup>04] G. Müller, J. Meseth, M. Sattler, R. Sarlette, and R. Klein. Acquisition, synthesis and rendering of bidirectional texture functions. In *EGSR*, pages 69–94, 2004. — cité p. 16
- [MN03] N. J. Mitra and A. Nguyen. Estimating surface normals in noisy point cloud data. In *Symposium on Computational Geometry*, pages 322–328, 2003. — cité p. 18
- [MOC<sup>+</sup>92] G. E. Miller, S. E. Offert, E. Chen, D. Patterson, S. Blacketter, S. A. Rubin, J. Yim, and J. Hanan. The virtual museum : Interactive 3d navigation of a multimedia database. *Visualization and Computer Animation*, 3(3) :183–197, 1992. — cité p. 208
- [MPDW03] V. Masselus, P. Peers, P. Dutré, and Y. D. Willems. Relighting with 4d incident light fields. In *ACM SIGGRAPH Computer Graphics*, volume 22, pages 613–620, 2003. — cité p. 45, 47, 52, 53, 57
- [MPN<sup>+</sup>02] W. Matusik, H. Pfister, A. Ngan, P. Beardsley, R. Ziegler, and L. McMillan. Image-based 3d photography using opacity hulls. In *ACM SIGGRAPH Computer Graphics*, pages 127–137, 2002. — cité p. 45, 46, 52, 57, 166

- [MRP98] G. Miller, S. Rubin, and D. Poncelon. Lazy decompression of surface light fields for precomputed global illumination. In *Eurographics Workshop on Rendering*, 1998. — cité p. 15
- [MWLT00] S. Marschner, S. Westin, E. Lafortune, and K. Torrance. Image-based brdf measurement. In *Applied Optics*, 2000. — cité p. 44, 45
- [NE01] P. Nillius and J.-O. Eklundh. Automatic estimation of the projected light source direction. In *IEEE CVPR*, pages 1076–1083, 2001. — cité p. 30, 31
- [NH91] G. M. Nielson and B. Hamann. The asymptotic decider : Resolving the ambiguity in marching cubes. In *IEEE Visualization*, pages 83–93, 1991. — cité p. 20, 97, 98
- [NZI01] K. Nishino, Z. Zhang, and K. Ikeuchi. Determining reflectance parameters and illumination distribution from a sparse set of images for view-dependant image synthesis. In *IEEE ICCV*, pages 599–606, 2001. — cité p. 49, 176
- [ON94] M. Oren and S. K. Nayar. Generalization of Lambert’s reflectance model. In *ACM SIGGRAPH Computer Graphics*, volume 28, pages 239–246, 1994. — cité p. 43
- [Pea85] D. R. Peachey. Solid texturing of complex surfaces. In *ACM SIGGRAPH Computer Graphics*, volume 19, pages 279–286, 1985. — cité p. 201
- [Pen82] A.P. Pentland. Finding the illuminant direction. *JOSA*, 72 :448–455, 1982. — cité p. 29
- [Pen88] A. Pentland. Shape information from shading : a theory about human perception. In *ICCV*, pages 404–413, 1988. — cité p. 22
- [Per85] K. Perlin. An image synthesizer. In *ACM SIGGRAPH Computer Graphics*, volume 19, pages 287–296, 1985. — cité p. 201
- [PF98] L. Papier and J. Françon. Evaluation de la normale au bord d’un objet discret 3d. *Revue Internationale de CFAO et d’Infographie*, 13(2) :205–226, 1998. — cité p. 23
- [Pho75] B. T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6) :311–317, 1975. — cité p. 41
- [PSG01] M. W. Powell, S. Sarkar, and D. Goldgof. A simple strategy for calibrating the geometry of light sources. *IEEE PAMI, 0-1 versus OM, Feb. 17*, 23(9) :1022–1027, 2001. — cité p. 35, 36, 37
- [PTVF99] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C, The Art of Scientific Computing*. Ed. Cambridge, 1999. — cité p. 30, 47, 146, 151
- [RH01] R. Ramamoorthi and P. Hanrahan. A signal-processing framework for inverse rendering. In *ACM SIGGRAPH Computer Graphics*, volume 35, pages 117 – 128, 2001. — cité p. 51
- [RHHL02] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. In *ACM SIGGRAPH Computer Graphics*, pages 438–446, 2002. — cité p. 18
- [SA93] P. Sinha and E. Adelson. Recovering reflectance and illumination in a world of painted polyhedra. In *ICCV*, pages 156–163, 1993. — cité p. 29

- [SAR<sup>+</sup>06] S. Savarese, M. Andreetto, H. Rushmeier, F. Bernardini, and P. Perona. 3d reconstruction by shadow carving : Theory and practical evaluation. *IJCV*, 2006. — cité p. 120
- [Sch94] C. Schlick. A survey of shading and reflectance models. *Computer Graphics Forum*, 13(2) :121–131, 1994. — cité p. 40
- [SD96] S. M. Seitz and C. R. Dyer. View morphing. In *ACM SIGGRAPH Computer Graphics*, 1996. — cité p. 209
- [SD97] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *IEEE CVPR*, pages 1067–1073, 1997. — cité p. 20
- [Set96] J. A. Sethian. *Level Set Methods*. Cambridge University Press, 1996. — cité p. 22
- [SGHS98] J. Shade, S. Gortler, L. W. He, and R. Szeliski. Layered depth images. In *ACM SIGGRAPH Computer Graphics*, volume 32, pages 231–242, 1998. — cité p. 206, 207
- [SH99] H-Y. Shum and L-W. He. Rendering with concentric mosaics. In *ACM SIGGRAPH Computer Graphics*, pages 299–306, 1999. — cité p. 12, 13
- [She05] R. Shewchuk. Star splaying : an algorithm for repairing delaunay triangulations and convex hulls. In *Symposium on Computational Geometry*, pages 237–246, 2005. — cité p. 18
- [SHS99] H. Schirmacher, W. Heidrich, and H. P. Seidel. Adaptive acquisition of lumigraphs from synthetic scenes. *Computer Graphics Forum*, 18(3), 1999. — cité p. 13
- [SKS02] P.-P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM TOG*, 21(3) :527–536, 2002. — cité p. 54, 55, 168, 173
- [SM99] D. Samaras and D. Metaxas. Coupled lighting direction and shape estimation from single images. In *IEEE ICCV*, pages 868–874, 1999. — cité p. 30
- [SS97] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. In *ACM SIGGRAPH Computer Graphics*, 1997. — cité p. 8
- [SSI99a] I. Sato, Y. Sato, and K. Ikeuchi. Acquiring a radiance distribution to superimpose virtual objects onto a real scene. *IEEE TVCG*, 5(1) :1–12, 1999. — cité p. 25
- [SSI99b] I. Sato, Y. Sato, and K. Ikeuchi. Illumination distribution from brightness in shadows : Adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions. In *IEEE ICCV*, pages 875–883, 1999. — cité p. 32, 34
- [SSI01] I. Sato, Y. Sato, and K. Ikeuchi. Stability issues in recovering illumination distribution from brightness in shadows. In *IEEE CVPR*, pages 400–407, 2001. — cité p. 32, 34
- [SWI97] Y. Sato, M. D. Wheeler, and K. Ikeuchi. Object shape and reflectance modeling from observation. In *ACM SIGGRAPH Computer Graphics*, volume 31, pages 379–388, 1997. — cité p. 18, 48, 53, 54, 55, 176
- [Sze90] R. Szeliski. Real-time octree generation from rotating objects. Technical Report CRL-90-12, Cambridge Research Laboratory, 1990. — cité p. 20
- [Sze93] R. Szeliski. Rapid octree construction from image sequences. In *CVGIP*, volume 1, pages 23–32, 1993. — cité p. 11, 19, 77

- [TDR99] P. Tellier and I. Debled-Rensson. 3d discrete normal vectors. In *DCGI*, pages 447–457, 1999. — cité p. 24, 111, 113
- [Thu01] G. Thurmer. Smoothing normal vectors on discrete surfaces while preserving slope discontinuities. *Computer Graphics Forum*, 20(2) :103–114, 2001. — cité p. 119
- [TNMM04] T. Takai, K. Niinuma, A. Maki, and T. Matsuyama. Difference sphere : An approach to near light source estimation. In *IEEE CVPR*, volume 1, pages 98–105, 2004. — cité p. 35, 36, 37
- [TS67] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surface. *JOSA*, 57 :1105–1114, 1967. — cité p. 32, 42
- [Tsa87] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Robotics and Automation*, RA-3(4) :323–344, 1987. — cité p. 24
- [TW97] G. Thurmer and C. A. Wuthrich. Normal computation for discrete surfaces in 3d space. *Computer Graphics Forum*, 16(3) :15–26, 1997. — cité p. 23, 109, 111
- [TW98] G. Thurmer and C. A. Wuthrich. Computing vertex normals from polygonal facets. *Graphics Tools*, 3(1) :43–46, 1998. — cité p. 23
- [War92] G. Ward. Measuring and modeling anisotropic reflection. In *ACM SIGGRAPH Computer Graphics*, pages 265–272, 1992. — cité p. 43, 47
- [WAT92] S. Westin, J. Arvo, and K. Torrance. Predicting reflectance functions from complex surfaces. In *ACM SIGGRAPH Computer Graphics*, pages 255–264, 1992. — cité p. 51
- [WG90] J. Wilhelms and A. Van Gelder. Topological considerations in isosurface generation extended abstract. *Computer Graphics*, 24(5) :79–86, 1990. — cité p. 20, 97
- [Wil83] L. Williams. Pyramidal parametrics. In *ACM SIGGRAPH Computer Graphics*, volume 17, pages 1–11, 1983. — cité p. 200, 201
- [WS03] Y. Wang and D. Samaras. Estimation of multiple directional light sources for synthesis of augmented reality images. *Graphical Models*, 65(4) :185–205, 2003. — cité p. 34
- [YDMH99] Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse global illumination : Recovering reflectance models of real scenes from photographs. In *ACM SIGGRAPH Computer Graphics*, pages 215–224, 1999. — cité p. 49, 53, 176, 184
- [YS04] Y. Yemez and F. Schmitt. 3d reconstruction of real objets with high resolution shape and texture. *Image and Vision Computing*, 22(13) :1137–1153, 2004. — cité p. 106
- [YSCS02] A. Yezzi, G. Slabaugh, R. Cipolla, and R. Schafer. A surface evolution approach of probabilistic space carving. In *3DPVT*, pages 618–621, 2002. — cité p. 26
- [YXA04] T. Yu, N. Xu, and N. Ahuja. Recovering shape and reflectance model of non-lambertian objects from multiple views. In *IEEE CVPR*, volume 2, pages 226–233, 2004. — cité p. 48
- [YY91] Y. Yang and A. Yuille. Source from shading. In *IEEE CVPR*, pages 534–539, 1991. — cité p. 30
- [ZK02] W. Zhou and C. Kambhamettu. Estimation of illuminant direction and intensity of multiple light sources. In *ECCV*, page 206, 2002. — cité p. 34, 35, 37

## BIBLIOGRAPHIE

---

- [ZPBG01] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross. Surface splatting. In *ACM SIGGRAPH Computer Graphics*, pages 371–378, 2001. — cité p. 205, 206
- [ZTCS99] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape from shading : A survey. *IEEE PAMI*, 21(8) :690–706, 1999. — cité p. 22
- [ZWGS02] Z. Zhang, L. Wang, B. Guo, and H.-Y. Shum. Feature-based light field morphing. In *ACM SIGGRAPH Computer Graphics*, pages 457–464, 2002. — cité p. 14
- [ZY00] Y. Zhang and Y. H. Yang. Illuminant direction determination for multiple light sources. In *IEEE CVPR*, pages 269–276, 2000. — cité p. 33, 34

---

## Résumé

Le rééclairage d'objets réels à partir d'images est un problème difficile à traiter car il implique l'estimation d'un nombre important de paramètres. Néanmoins le rééclairage de tels objets est indispensable pour une intégration de manière réaliste dans un environnement comportant des conditions d'illumination différentes de celles présentes lors de l'acquisition. Le travail proposé dans ce mémoire décrit une méthodologie de reconstruction automatique de toute la chaîne d'analyse : il couvre tous les aspects allant de la reconstruction géométrique à l'estimation de la réflectance de la surface en passant par la détection de sources lumineuses. Les seules données nécessaires à l'application de cette méthode sont de multiples points de vue de l'objet acquis sous des conditions d'illumination fixes et les paramètres de la caméra. Aucun objet additionnel n'est présent sur les images et la surface de l'objet basé-image peut être diffuse, spéculaire et/ou texturée. Nous avons mis en place un système d'acquisition robuste constitué d'un appareil photographique grand marché et d'un plateau tournant. Notre modèle géométrique reconstruit permet de passer d'un modèle discret à un modèle triangulé sans perte d'information topologique (propriétés de fermeture et nombre de composantes connexes conservés). De multiples sources ponctuelles et directionnelles sont détectées et la réflectance des objets est estimée sur chaque maille par un modèle de BRDF paramétrique. Nous utilisons les données reconstruites pour visualiser les objets avec de nouvelles conditions d'illumination. La version actuelle permet de générer de nouvelles images en quelques secondes avec une méthode de lancer de rayons.

**Mots-clés :** rendu basé-image - algorithme de sculpture - maillage - détection de multiples sources lumineuses - estimation de propriétés de réflectance - rééclairage.

---

## Abstract

Relighting real objects from images is a difficult problem since a lot of parameters have to be estimated. Nevertheless such an operation is essential for integrating real objects into virtual scenes, for instance in the context of movies, advertisements or video games. The work proposed in this report describes a methodology for automatically reconstructing all stages of real objects estimation : object acquisition, geometry reconstruction, light sources detection, reflectance recovery and relighting. The objects we use are only represented by a series of calibrated images, they can be composed by diffuse, glossy surfaces and textures. The acquisition system, composed by a digital camera and a turntable, is robust and no additional object or pattern is used during the acquisition phase. We reconstruct a discrete geometric model and a triangular mesh having similar topological properties (closed surface, number of connected components). Multiple punctual and directionnal light sources can be detected. Surface reflectance properties are estimated on each mesh by a parametric BRDF model. The estimated information is used for visualizing objects with new illumination conditions. New images are produced in a few seconds with a ray-tracing method.

**Keywords:** image-based rendering - shape from silhouette - marching cubes - multiple light sources detection - reflectance properties recovery - relighting.

---