



HAL
open science

Détourage interactif de photographies d'Intérêt général

Marc Bertier

► **To cite this version:**

Marc Bertier. Détourage interactif de photographies d'Intérêt général. Génie logiciel [cs.SE]. Ecole Nationale Supérieure des Mines de Saint-Etienne, 1983. Français. NNT: . tel-00807932

HAL Id: tel-00807932

<https://theses.hal.science/tel-00807932>

Submitted on 4 Apr 2013

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.

THESE

présentée par

Marc BERTIER

pour obtenir le diplôme de

DOCTEUR-INGENIEUR

EN INFORMATIQUE

DETOURAGE INTERACTIF DE PHOTOGRAPHIES D'INTERET GENERAL

Soutenue à Saint-Etienne le 1er Juillet 1983 devant

la commission d'examen composée de

Président M. M. NOUGARET

Examineurs MM. Ph. COUEIGNOUX

M. HABIB

R. MAHL

J. SERRA

M. GANGNET

Rapporteurs

THESE

présentée par

Marc BERTIER

pour obtenir le diplôme de

DOCTEUR-INGENIEUR

EN INFORMATIQUE

DETOURAGE INTERACTIF DE PHOTOGRAPHIES D'INTERET GENERAL

Soutenu à Saint-Etienne le 1er Juillet 1983 devant

la commission d'examen composée de

Président M. M. NOUGARET

Examineurs MM. Ph. COUEIGNOUX

M. HABIB

R. MAHL

J. SERRA

M. GANGNET

} Rapporteurs

ECOLE NATIONALE SUPERIEURE DES MINES DE SAINT-ETIENNE

Directeur : M. M. MERMET
Directeur des Etudes et de la Formation : M. J. LEVASSEUR
Directeur des Recherches : M. J. LEVY
Secrétaire Général : Melle M. CLERGUE

PROFESSEURS DE 1ère CATEGORIE

MM. COINDE	Alexandre	Gestion
GOUX	Claude	Métallurgie
LEVY	Jacques	Métallurgie
LOWYS	Jean-Pierre	Physique
MATHON	Albert	Gestion
RIEU	Jean	Mécanique - Résistance des Matériaux
SOUSTELLE	Michel	Chimie
FORMERY	Philippe	Mathématiques Appliquées

PROFESSEURS DE 2ème CATEGORIE

MM. HABIB	Michel	Informatique
PERRIN	Michel	Géologie
VERCHERY	Georges	Matériaux
TOUCHARD	Bernard	Physique Industrielle

DIRECTEUR DE RECHERCHE

M. LESBATS	Pierre	Métallurgie
------------	--------	-------------

MAITRES DE RECHERCHE

MM. BISCONDI	Michel	Métallurgie
DAVOINE	Philippe	Géologie
Mle FOURDEUX	Angeline	Métallurgie
MM. KOBYLANSKI	André	Métallurgie
LALAUZE	René	Chimie
LANCELOT	Francis	Chimie
LE COZE	Jean	Métallurgie
THEVENOT	François	Chimie
TRAN MINH	Canh	Chimie

PERSONNALITES HABILITEES A DIRIGER DES TRAVAUX DE RECHERCHE

MM. DRIVER	Julian	Métallurgie
GUILHOT	Bernard	Chimie
THOMAS	Gérard	Chimie

REMERCIEMENTS.

=====

Je tiens à remercier ici toutes les personnes qui ont contribué à l'achèvement de ce travail :

Je pense en premier lieu à M. Philippe COUEIGNOUX qui -après avoir marqué de son empreinte l'essentiel de cette recherche- a continué après son départ de l'école des mines, malgré ses lourdes tâches et en dépit de son éloignement, à m'encourager et à me soutenir jusque dans la mise au point du texte.

Je remercie tout aussi chaleureusement M. Michel HABIB qui n'a pas hésité à s'investir personnellement dans un travail pourtant déjà largement ébauché, pour lui donner un nouveau souffle et l'enrichir de toute une réflexion théorique qui sans cela, lui aurait fait défaut.

Je remercie chacune des personnalités qui ont accepté de juger ce travail. C'est un honneur pour moi de voir M. Marcel NOUGARET, président, et messieurs Robert MAHL, Jean SERRA et Michel GANGNET, réunis dans ce jury.

Je remercie enfin mesdames AVONDO, BONNEFOY et BARNEOUD, et messieurs LOUBET, BROSSARD, VELAY et DARLHES qui ont mené à bien la réalisation matérielle du rapport, et tout le personnel présent et passé du département informatique de l'école, dont la camaraderie a été une aide quotidienne.

Quant à Nicole et Noémie, leur patience dans les moments difficiles et le réconfort qu'elles m'ont prodigué alors, sont au-dessus de tout remerciement. Je leur dédie cet ouvrage.

AVANT PROPOS



Dans le premier chapitre de ce texte, nous présenterons et mettrons en situation longuement l'objet du travail de recherche rapporté dans la suite.

Pour une présentation du plan général du document, le lecteur devra se reporter au dernier paragraphe (I.4.) du chapitre d'introduction.

TABLE DES CHAPITRES.

=====

I.	Introduction	5
II.	Extraction de Contour par Déformations Successives	41
III.	Recherche de Contour comme Chemin Optimal dans un Graphe	81
IV.	Modélisation	119
V.	Le Modèle du Pantin	139
VI.	Analyse d'Images de Batiments	159
VII.	Conclusion : Vers un Système Intégré de Détourage	189

CHAPITRE 1

=====

Chapitre I : INTRODUCTION.

=====

I.1. Production de Documents Assistée par Ordinateur.

I.1.1. Traitement et Analyse de Textes.

I.1.2. Informatique Graphique et Imprimerie.

I.1.3. Informatique et Images.

I.1.3.1. Amélioration d'Images.

I.1.3.2. Codage/Décodage.

I.1.3.3. Synthèse d'Images.

I.1.3.4. Analyse d'Images.

I.2. Montage de Documents Mixtes.

I.2.1. Photomontage et Bureautique.

I.2.2. Détourage.

I.3. Le Détourage et la Segmentation d'Images.

I.3.1. Techniques de Base de Segmentation d'Images.

I.3.2. Segmentation d'Images et Interprétation.

I.3.3. Le Corpus des Photographies d'Intérêt Général.

I.4. Présentation des Chapitres suivants.

Bibliographie.

Chapitre I : INTRODUCTION.

=====

Dans ce chapitre d'introduction, nous allons présenter, sous deux points de vue très différents, le sujet du travail rapporté dans la suite.

Dans un premier temps, nous mettrons en valeur le **contexte technologique matériel et surtout logiciel** qui permet aujourd'hui le développement de **systèmes de mise en page pour les métiers de l'édition** (§ I.1).

Dans cette perspective, apparaîtront d'eux mêmes, la motivation et l'intérêt pratique présentés par l'étude détaillée du **détourage**, opération fondamentale qui consiste à extraire d'une photographie l'image d'un objet y figurant (§ I.2).

Nous aborderons, dans la suite de ce chapitre une présentation plus technique des difficultés soulevées par l'automatisation du détourage. Plus précisément, nous présenterons le problème comme relevant à la fois de la **segmentation** et de **l'interprétation d'image** (§ I.3).

Nous donnerons enfin (§ I.4), pour conclure cette longue introduction, le **plan général du texte**.

I.1. La production de documents assistée par ordinateur.

Nous appellerons ici "document", un support d'information, bidimensionnel, statique et destiné à une exploitation visuelle : Une page de magazine, une image télévision fixe, un schéma de circuit électronique en sont de bons exemples; sont par contre exclus par cette définition volontairement restrictive, les séquences d'images animées, les enregistrements du langage parlé...

En ce sens, un document contient essentiellement trois composantes : des textes (manuscripts, dactylographiés ou imprimés); des graphismes (dessin industriel, gravure...); et des images (photographies, peinture d'art...).

Pour chacune de ces composantes, l'informatique s'est révélée la source d'un gain de productivité utile sinon indispensable.

I.1.1. Traitement et Analyse des textes.

Il y a déjà longtemps que l'outil informatique a été introduit dans les métiers de la production de documents textuels.

Dans le domaine de la presse, les premières applications ont concerné la saisie, la correction et la mise en page des textes; puis se sont développés en amont, l'usage de réseaux de transmission de données numériques (télématique), et en aval, les procédés de composition automatique des textes.

Le développement accéléré de la micro-informatique favorise aujourd'hui une très large diffusion des techniques de traitement de textes dans les entreprises, voire dans le grand public. Sur la

bureautique, le lecteur disposera avec /SEYBOLD REPORT/ d'une abondante source de documentation.

Les travaux d'analyse automatique de textes ont une audience bien plus restreinte. Ils sont encore dans bien des cas, au stade de la recherche universitaire. Dans les deux domaines principaux d'application, la documentation et la traduction automatiques, des résultats partiels ont été obtenus. Voir par exemple /BOITET 83/.

I.1.2. Informatique Graphique et Imprimerie.

L'informatique graphique est née de la conception assistée par ordinateur. A l'origine, la C.A.O. est un outil qui permet d'accroître la productivité des bureaux d'études parce qu'il automatise et relie entre elles les tâches de calcul, de représentation et d'établissement d'un programme de fabrication.

Dans la mesure où le produit ultime est dans la majorité des cas, le programme de fabrication, l'étape de représentation aurait pu s'avérer superflue; cependant les documents restent encore le plus sûr moyen d'archiver et de transmettre l'information finale. De plus, le document peut constituer lui-même, dans certains cas, le produit final de la conception : masques pour circuits intégrés par exemple.

La C.A.O. a donc eu un certain impact sur les techniques de production de documents : elle a suscité un important développement des matériels et des logiciels de dessin automatique (technologies de tracé à haute résolution; logiciels de représentation graphique).

Accessoirement, ces développements nécessaires à la C.A.O. ont permis l'apparition de nouveaux domaines de production de

documents graphiques :

-la présentation graphique de tableaux de chiffres sous forme de diagrammes spectaculaires est même devenu un service commercial offert par quelques sociétés spécialisées (ISSCO, ...). La concision et la clarté de ces modes de représentation est remarquable et leur permettront peut-être de s'imposer.

- l'introduction des techniques de **l'informatique graphique dans le domaine de l'imprimerie**, est liée à une mutation plus ancienne : Depuis vingt ans, la composition des textes au plomb est progressivement abandonnée au profit de la composition sur film (photocomposition). Dès lors, la représentation digitale des caractères d'imprimerie a toutes les chances de s'imposer dans la mesure où des techniques de codage dense et de décodage rapide ont été développées /COUEIGNOUX 81-a/.

Les documents graphiques sont aussi l'objet de travaux d'**analyse** :

Plusieurs générations de lecteurs optiques de caractères dactylographiés ou imprimés, se sont succédées aux stades de prototype et de produits commerciaux /COUEIGNOUX 81-b/.

Dans plusieurs centres de recherche ont été et sont encore développés des programmes d'**analyse de documents graphiques** divers : codage de schéma électroniques ou cartes météo, contrôle de fabrication de circuits imprimés, saisie de plans d'architecture par analyse optique du tracé /COQUILLARD et al. 82/.

I.1.3. Informatique et Images.

L'utilisation des techniques informatiques dans les domaines où les informations à traiter ont intrinsèquement une nature iconique, date de plus de 20 ans. Une documentation volumineuse

s'est constituée dans laquelle les ouvrages de synthèse qui suivent font autorité : /ROSENFELD et KAK 76/ /PRATT 78/ /BALLARD 82/ /FOLEY ET VAN DAM 82/; les revues spécialisées les plus importantes sont : /CGIP/ /PAMI/ /PR/ /CGW/.

Toutefois, jusqu'à ces dernières années, cela restait une application un peu en marge de l'informatique "classique", et surtout complètement déconnectée des filières traditionnelles de production et de traitement d'images.

Le traitement d'images par ordinateur s'est principalement imposé jusqu'ici, dans le domaine des applications militaires, spatiales et médicales.

Une raison fondamentale de ce développement limité, tient avant tout aux lourdes exigences du traitement et de la synthèse d'images en moyens matériels et logiciels (puissances de calcul et de mémorisation).

Typiquement, un texte se présente vis à vis d'un programme comme une suite monodimensionnelle de caractères pris dans un jeu de 200 signes environ; et il s'agit de les disposer séquentiellement sur des lignes consécutives. Le texte d'une page dactylographiée de format courant représente de l'ordre de 4 k-octets d'information.

Un document graphique au trait se présente aussi comme une liste : liste de segments à tracer et de déplacements du curseur dans le plan de l'image. On peut évaluer à 10 k-octets, l'information contenue dans un plan d'architecture.

Une image fixe au contraire, est fondamentalement de nature bidimensionnelle : à tout point de la fenêtre de visualisation est associée une valeur d'intensité lumineuse et, dans le cas d'une image en couleurs, une valeur de chrominance et de saturation (ou encore une valeur d'intensité par couleur : rouge, vert, bleu).

Pratiquement pourtant, les programmes n'ont pas accès à la valeur de l'intensité en tout point de l'image; ils ne disposent de celle-ci qu'à travers une représentation numérique approchée, approximation obtenue par **échantillonnage et quantification** de l'image réelle.

On peut noter à cette occasion que la "synthèse" d'images sous forme échantillonnée et quantifiée existe de puis très longtemps, puisqu'elle est à la base de techniques telles que la mosaïque et la broderie (cf. fig. I.1).

On obtient une approximation d'autant plus fidèle que les pas d'échantillonnage et de quantification sont petits. Pour un rendu visuellement acceptable, on estime généralement nécessaire une grille d'échantillonnage d'au moins plusieurs centaines de points de côté, et une quantification sur une centaine de niveaux par couleur (cf. fig. I.1).

Il s'en suit immédiatement que la quantité d'information mémorisée dans une image est d'un ordre de grandeur largement supérieur aux chiffres précédents; soit environ un méga-octet pour une image couleur de format A4.

Malgré cette difficulté inhérente aux images, chacun des grands thèmes du traitement d'images par ordinateur, a fini par dégager des résultats susceptibles d'être introduits dans la production assistée de documents.

I.1.3.1. Amélioration d'image.

Disposant d'un organe de numérisation d'image (caméra électronique à tube et convertisseur analogique/numérique...) et d'un organe de restitution (imprimante point à point, sortie sur film ou visualisation sur moniteur télévision...), il est possible de modifier par programme la représentation numérique de l'image dans le but d'en améliorer le rendu, l'aspect /SCHREIBER 78/ /TROXEL et LYNN 78/.

Figure I.1. : Echantillonnage et Quantification d'Images.

a. Extrait d'un catalogue de broderie

Photographie numérisée et restituée
sur moniteur télévision,

b. 32x32 pixels, 8niveaux;

c. 512x512 pixels, 8 niveaux;

d. 32x32 pixels, 64 niveaux;

e. 512x512 pixels, 64 niveaux.



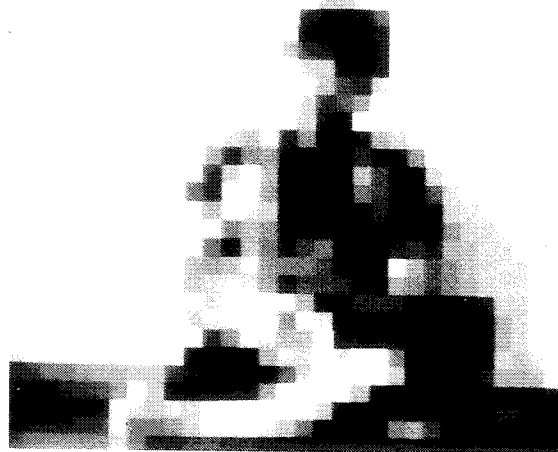
-a-



-b-



-c-



-d-



-e-

On distingue les méthodes "directes" consistant à appliquer des transformations numériques aux valeurs représentant le niveau de gris des pixels ; et les méthodes dérivées de la théorie du signal où la notion de signal fonction d'une variable unique, le temps, est étendue à la notion de signal fonction de deux variables d'espace (filtrage à 2 indices).

Il est ainsi possible de réaliser numériquement non seulement la plupart des traitements réalisés couramment en laboratoire, fig. I.2, (chambre noire) mais même éventuellement d'autres traitements réputés impossibles par les photographes : réduction du flou de mise au point, de bougé... /GRANLUND et KNUTSSON 82/

I.1.3.2. Codage/Décodage

Le stockage "tel quel" d'images numérisées est coûteux vu le volume d'information impliqué. Il en est de même de leur transmission. Aussi les problèmes de compression de l'information d'image ont-ils été étudiés très tôt : de nombreuses méthodes, adaptées ou non à des types d'images particuliers, basées sur des techniques de codage et/ou des transformations linéaires, ont été mises au point.

On parle généralement de méthodes sans perte d'information d'une part avec lesquelles on peut atteindre des taux de compression proches de 10:1 pour des photographies naturelles ; et de méthodes avec perte d'information qui permettent des taux de compression supérieurs (jusqu'à 50:1) et que l'on arrive à combiner avec une amélioration subjective du rendu de l'image! /KOCHER et KUNT 83/ /KUNT 82/ /TROXEL et al. 80/.

Dans la pratique, le développement de ces techniques a conduit, aux USA et dès 75, à la mise en place, pour les deux plus importantes agences de photographie de presse, de réseaux de transmission par fac-similé numérique /SCHREIBER 74/.

Figure I.2. : Traitements numériques d'images.

-a-
négatif du
cliché de la
figure I.1.e.

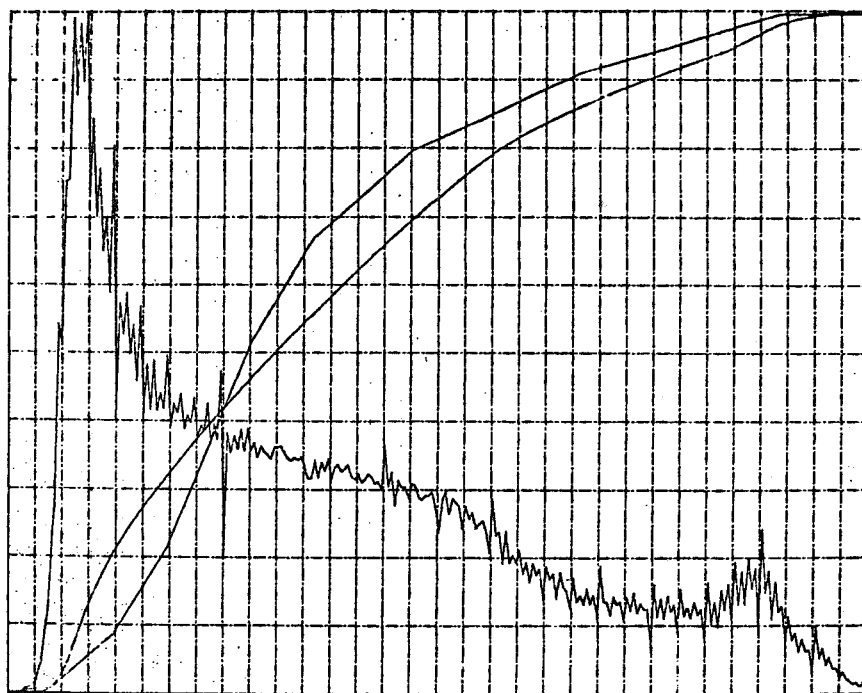


Ci-dessous,
(c) rehaussement du
contraste du cliché
(b), par modification
de l'histogramme des
teintes. (d).



-b-

-c-



-d-

I.1.3.3. Synthèse

Il s'agit là d'un domaine où l'on cherche à faire construire par programme une image, non pas ex nihilo, mais à partir d'une description codée et concise. Celle-ci peut avoir été fournie au programme par un opérateur ou élaborée par des programmes exécutés précédemment (CAO et en particulier tout le domaine du "solid modeling" /REQUICHA 80/). Le programme de synthèse, construit à partir de la description d'un objet et sous un angle à choisir, la représentation numérique de son image.

Ces techniques commencent à s'imposer dans le dessin animé, mais la finesse de résolution recherchée 7000x4000x24 impose une puissance et un temps de traitement considérables.

Sur la synthèse d'image, voir les actes des congrès de l'ACM /SIGGRAPH ...82/.

I.1.3.4. Analyse

La représentation numérique d'une image réelle étant disponible dans la mémoire de l'ordinateur, les problèmes d'analyse consistent à en **extraire** des informations globales.

Des programmes peuvent par exemple opérer des **mesures statistiques** plus ou moins simples (teinte moyenne, histogramme des teintes, fig. I.2(d), ou des **comparaisons** complexes entre l'image et des patrons prototypes de formes connus à l'avance. C'est le vaste domaine de la reconnaissance des formes appliquée à l'image /DUDA et HART 73/.

I.2. Le montage de documents mixtes.

I.2.1. Photomontage et bureautique.

Le texte, le dessin au trait et l'image sont donc des domaines où l'informatique s'est d'ores et déjà imposée; mais jusqu'à présent, les applications concernent en général l'un des trois domaines à l'exclusion des deux autres. Textes, graphiques et images restent traités par des voies distinctes et ne se trouvent mêlés qu'au dernier moment, le plus souvent par simple collage.

Il existe pourtant un point nécessaire de convergence, un ensemble d'activités où doivent être mis en oeuvre conjointement le traitement de texte, le graphique, le traitement et la synthèse d'images : la **production des documents** où des informations des trois natures coexistent.

On pense en premier lieu aux tâches de photomontage, de constitution de maquettes et de mise en page dans les métiers de la presse, de l'édition et de la publicité /BORREL 82/.

Il peut également s'agir - à plus long terme - de tout le domaine des travaux de bureau : Un outil sophistiqué de mise en page texte/image constitue une extension prévisible des systèmes de bureautique (Réalisation dans l'entreprise même, de rapports et documents de haute qualité visuelle ...).

Pour exploiter cette convergence, il est d'une efficacité douteuse d'appliquer une technique à un domaine pour lequel elle n'a pas été initialement conçue :

l'impression des images sur une imprimante disposant d'un jeu standard de caractères, est possible mais donne des résultats de piètre qualité;

l'impression de graphiques en photocomposition (APS-5 de Autologic) ou sur machine de traitement de textes, à l'aide d'une police spéciale de caractères (on parle de "semi-graphique"), est

un autre exemple de cette médiocre adaptation a posteriori.

L'intérêt qu'il y a, à faire assister le travail de montage par un ordinateur est le même que dans bien d'autres cas : Obtention plus rapide du résultat ; Appréciation du rendu par visualisation sur moniteur vidéo ; pas de manipulation matérielle ; possibilité de retour en arrière ; comparaisons d'options différentes ; Automatisation de tâches fastidieuses ; fonctionnalités nouvelles, traitements numériques simulant les traitements photographiques.

Du point de vue de l'importance du matériel à mettre en jeu, deux remarques peuvent être faites :

Dans les deux cas, de la presse et de la bureautique, la structure informatique préexiste ; l'augmentation de la puissance de traitement (traitements spécialisés "cablés") et de mémorisation (mémoire(s) d'image) peut s'avérer indispensable, mais ne constitue pas, et constituera de moins en moins, un obstacle économique important.

Par contre, les périphériques spécialisés nécessaires à la saisie des documents originaux et à la restitution des résultats, sont encore très coûteux ou de qualité souvent encore insuffisante pour concurrencer les techniques photographiques classiques.

Quant au logiciel, un important travail d'intégration sera nécessaire, pour tirer profit de cette convergence. Parmi les fonctions nouvelles à mettre au point, la fonction de détournage occupe une place centrale et c'est à elle que nous nous sommes intéressé.

I.2.2. Détourage

En photomontage, on appelle **"détourage"**, l'opération qui consiste à extraire d'une photographie, l'image d'un objet y figurant ; par exemple, pour l'intégrer ensuite dans une autre scène, ou plus simplement, pour la mettre en valeur sur un fond uni, ou encore pour la faire apparaître en surimpression d'un texte, etc...

Le caractère central du détourage peut s'illustrer par analogie avec le traitement de texte : le détourage joue vis à vis des images, un rôle équivalent à celui de la localisation d'un mot dans un texte.

Le détourage est rarement réalisé par découpe aux ciseaux (résultats très médiocres), mais plutôt par application d'une gouache opaque ou d'une pellicule inactinique, sur les parties à éliminer d'une épreuve positive ou négative sur film transparent.

C'est une opération très courante et assez délicate. Elle n'est pas vraiment réversible et ne donne pas toujours des résultats satisfaisants à cause de la limite nécessairement franche entre une zone gouachée et une zone non gouachée, ou à cause de découpes trop complexes.

C'est à une tentative d'automatisation de cette opération que nous avons travaillé. On verra plus loin qu'une automatisation complète n'est pas envisageable mais ceci n'est pas rédhibitoire :

Dans le domaine de la production de documents, à l'inverse de la plupart des tâches en reconnaissance des formes et en particulier en robotique, la présence d'un opérateur est de toutes façons, nécessaire. Simplement, au lieu d'être absorbé à des tâches manuelles, il pourra se consacrer plus librement à l'aspect créatif de son travail.

Le premier objectif qu'il est donc intéressant d'atteindre est d'alléger et d'assouplir le travail de montage tel qu'il se déroule avec des moyens classiques : possibilité pour l'opérateur

et sans aucune manipulation matérielle effective, de faire des tentatives multiples, de visualiser les résultats, de revenir sur des choix précédents, etc...

Il est aussi envisageable, dès lors que le détournage est assisté par un ordinateur, d'offrir des **fonctionnalités nouvelles** :

- fondu (*) de l'objet rapporté sur le nouveau fond, dans certaines parties du contour;

- justification de texte d'après le contour d'un objet;

- traitements numériques reproduisant les traitements en chambre noire, dans des zones délimitées par détournage;

- Améliorations impossibles à obtenir par procédé photographique (réduction des flous par "déconvolution" numérique).

(*) extension du terme utilisé en technique cinématographique pour indiquer la transition progressive dans le temps entre 2 plans, pour indiquer ici la transition progressive dans l'espace entre 2 images fixes.

I.3. Le détournage, un problème de segmentation numérique d'image.

L'opération de détournage évoquée plus haut, rappelle évidemment la notion omniprésente en analyse d'image par ordinateur de **segmentation**. Le terme de segmentation a toutefois une signification beaucoup plus large et recouvre des techniques et des buts très divers.

Après avoir donné un aperçu synthétique de la segmentation des images, nous allons redéfinir le détournage dans cette perspective (I.3.1). Puis nous établirons les liens qui peuvent exister entre segmentation et interprétation d'images (I.3.2). Enfin nous caractériserons le corpus photographique que nous considérerons (I.3.3).

I.3.1. Les techniques de base de segmentation d'image.

D'une façon tout à fait générale, on appelle segmentation d'une image toute opération qui aboutit à une **partition** de celle-ci.

A l'intérieur des zones (ou "segments") ainsi obtenues, est vérifiée une condition d'uniformité sur un certain critère calculable en tout point; et l'uniformité n'est réalisée sur aucun autre sous-ensemble de l'image.

Le cas où la segmentation est totalement arbitraire (i.e. n'a aucune **signification**), correspond au cas où le seul critère est formé des fonctions caractéristiques des parties. Ce cas ne présente bien entendu, aucun intérêt pratique.

Dès lors, il est commode et classique de séparer les méthodes en deux grandes classes :

- les méthodes de croissance de zones uniformes au sens du critère /ZUCKER 76/ /ROSENFELD 82/.
- les méthodes d'extraction des points de variation importante du

critère. On parle de méthodes d'extraction de frontières /ROSENFELD et KAK 76/.

Idéalement, il est clair que les approches des deux types sont strictement équivalentes dans la mesure où l'uniformité est rigoureusement respectée dans chaque zone. Dans la pratique par contre, des méthodes des deux types sont utilisées, car pour chaque critère, une seule des deux approches est commodément applicable.

Le critère le plus élémentaire et le plus fréquemment utilisé, est la **valeur de gris** en chaque point. On aboutit ainsi :

- d'une part, aux méthodes de croissance de zones de gris uniforme.
- et d'autre part, aux méthodes d'extraction de contours.

Bien d'autres critères ont été proposés; le plus important n'est pas de les décrire mais de les classer selon les éléments qu'ils font intervenir.

Pour un algorithme quelconque de segmentation, le critère sur lequel la condition d'uniformité est testée, fait intervenir :

- la valeur de gris du pixel (ou sa couleur); (a)
- la localisation du pixel dans l'image; (b)
- la valeur des pixels du voisinage de celui considéré; (c)
- la valeur et la localisation par rapport au pixel considéré, de pixels voisins; (d)
- la valeur de tous les pixels de l'image; (e)
- la valeur et la localisation par rapport au pixel considéré, de tous les pixels. (f)

On peut reprocher à cette classification sa redondance : tous les critères de (a) jusqu'à (e), sont inclus dans le critère (f). Elle est toutefois intéressante pour comparer en terme de "localité/globalité", quelques procédés classiques de segmentation et le détournage réalisé par un opérateur humain.

Seuillage :

Le seuillage est un exemple de procédé numérique élémentaire de segmentation d'une image en deux zones. Chaque pixel est classé selon que sa valeur est inférieure ou non à un seuil. Si celui-ci est fixé indépendamment de l'image, le seul critère de classement est de type (a).

On peut au contraire, calculer le seuil à partir de la distribution des valeurs de gris dans toute l'image (histogramme). Le classement fait alors intervenir une influence de type (a) (seuillage) et une influence de type (e) (dans le calcul préalable du seuil).

Enfin le "seuillage adaptatif" correspond au cas où le calcul du seuil est réalisé dans une portion réduite d'image (influence de type (c)).

Fenêtrage :

C'est un cas dégénéré de segmentation où seul intervient le critère (b).

Condition de Connexité :

Une importante variante de la segmentation par seuillage consiste à imposer de surcroît, la connexité des segments. On peut analyser cette contrainte supplémentaire comme influences de types (b) et (c) agissant sur le résultat d'une 1^o segmentation : le classement d'un pixel dépend de son classement à la 1^o segmentation, de celui de ses voisins, mais aussi de sa propre localisation puisque deux pixels de même valeur, dans deux voisinages strictement identiques, peuvent être classés différemment.

Relaxation :

La nécessité de prendre en compte de façon explicite et très précise, l'influence du voisinage dans la segmentation, est l'idée fondamentale qui est à l'origine des techniques dites "par relaxation".

L'influence sur le classement d'un pixel, de chacun des pixels du voisinage, est fonction des valeurs des deux points (valeur de gris ou amplitude et direction de bord) et de leur localisation relativement l'un à l'autre (influence de type (d)).

Il s'agit de techniques itératives où le souci d'obtenir une convergence rapide limite à une taille réduite, le voisinage dont l'influence est prise en compte. /ROSENFELD 77/ /PRAGER 80/

Détourage :

Le détourage effectué de façon entièrement "manuelle", constitue bien au sens donné plus haut, une segmentation, mais il est très difficile d'identifier avec précision tous les critères qui guident l'opérateur humain dans son travail. On peut toutefois énumérer certains "sous-critères" qui interviennent dans le choix d'un contour.

- critères locaux :

- . variation importante (ou uniformité) de la luminosité;
- . variation importante (ou uniformité) de la distribution locale des niveaux de gris;

- critères régionaux :

- . régularité, longueur du contour,
- . ou forme, aire des segments;

- critères globaux :

- . connexité des segments,
- . ou fermeture des contours;
- . vraisemblance du résultat.

A ce dernier titre (usage de critères globaux), le détourage "manuel" relève de la catégorie (f) définie ci-dessus et diffère notablement des procédés classiques de segmentation. Il pose le problème crucial des relations entre segmentation et interprétation en reconnaissance des formes.

I.3.2. Relations entre segmentation et interprétation.

Dans la littérature de l'analyse d'images, la segmentation apparaît dans deux optiques bien différentes : elle peut constituer elle-même l'objectif ultime de l'analyse, ou se présenter au contraire comme une étape préliminaire à une analyse proprement dite.

Dans le 1° cas, et pour de nombreux domaines d'application, des procédés efficaces ont été mis au point; soit qu'une méthode simple et très générale de segmentation donne le résultat souhaité mais ceci est rare; soit qu'ait été développées des méthodes particulières à la fois à un corpus et à un problème précis sur ce corpus. Ces procédés s'appuient plus ou moins implicitement sur une certaine modélisation de la classe particulière d'images associée à un domaine. Cette démarche qui consiste à adapter la segmentation au corpus et au problème considérés, a été très tôt et très largement utilisée. Son efficacité est bonne pour des **classes d'images à la fois étroites et bien typées** (standardisées) et pour lesquelles une modélisation implicite simple (le standard) s'impose donc d'elle même. C'est pourquoi ces techniques ont été utilisées fréquemment sur l'imagerie scientifique : microphotographies, radiographies ... dans le domaine médical /FU et MUI 81/; analyse de photos de chambres à bulles ou à étincelles en physique des particules.

Dans le 2° cas au contraire, face à l'insuffisance des renseignements a priori sur l'image à analyser, la segmentation ne peut aboutir directement au résultat souhaité; cependant une étape préliminaire de segmentation opérant sur des critères très généraux, présente deux intérêts :

- le 1° est de réduire l'information à manipuler;
- l'autre est de permettre de poser des hypothèses d'identification de segments dont il faudra alors tester la compatibilité, dans le but d'aboutir à une interprétation globale cohérente. On peut dire alors que la segmentation joue le rôle

d'extraction de primitives .

Le détournage présente la particularité de s'apparenter aux deux démarches à la fois.

Il se rapproche de la première dans la mesure où la segmentation de l'image est bien ici par définition, le but ultime de l'opération.

/KELLY 72/ présente un travail qui correspond exactement à cette démarche : il s'agit de retrouver la silhouette d'un personnage sur un portrait photographique. Pour cela, le programme agit conformément à un modèle anatomique extrêmement simple et rigide codé dans ses instructions : son comportement heuristique consiste à tester l'acceptabilité de triplets d'éléments de bords comme appartenant au sommet et aux cotés de la tête du personnage; si leurs localisations et leurs positions relatives les rendent admissibles, on cherche un contour qui les relie; si le triplet ne peut être accepté ou si ses 3 points ne peuvent être reliés, on envisage le triplet suivant. L'auteur ajoute que cette recherche ne pouvant être réalisée à la résolution originale, elle est effectuée à une résolution plus grossière et le 1° résultat ainsi obtenu servira alors de guide pour la segmentation finale (c'est ce que l'auteur appelle "planning", terme issu du domaine de l'intelligence artificielle).

On peut rattacher à cette démarche le travail de /BAIRD et KELLY 74/, bien que cette fois ci et sur le mêmes portraits photographiques, le but ne soit plus la segmentation personnage/fond, mais l'identification des tâches correspondant aux yeux, à la bouche etc..., pour opérer des mesures anthropométriques.

Il est clair que si la façon précédente d'envisager le détournage est importante, elle est quand même bien insuffisante : il n'est évidemment pas possible de réduire le détournage sur un large ensemble de photos peu standardisées, à la recherche d'éléments de bords vérifiant une configuration exactement prédéterminée. Il est au minimum nécessaire que l'opérateur puisse

préciser interactivement le modèle précis qui convient pour l'image à traiter.

Cette nécessité est à rapprocher de l'expérience psycho-visuelle citée dans /PAVLIDIS 77/ et qui consiste à présenter à un observateur, une image binaire (N et B strict) qui en première vision, semble ne rien représenter d'identifiable. Dans cette situation, l'observateur s'avère incapable de détourner cette image autrement qu'en isolant les parties noires des parties blanches. Si l'on indique alors à l'observateur que la photo représente un chien dalmatien sur un fond de feuilles mortes, il identifie en général instantanément le contour de l'animal et est alors capable de détourner l'image.

Cette expérience prouve bien la quasi-nécessité, au moins dans les cas extrêmes, de l'apport d'une information en plus de celle contenue dans l'image. On parlera d'information a priori.

Enfin, le détournement peut aussi relever de la deuxième des démarches décrites plus haut.

Le détournement présente alors une grande similitude avec les expériences menées en compréhension d'images (voir par exemple /RISEMAN et HANSON edtrs 78/) : comme le suggère l'expérience du dalmatien, l'identification des éléments de la scène est parfois une étape nécessaire pour aboutir au détournement; cependant dans les cas moins difficiles, l'identification complète n'est pas indispensable -voire inutile dans des cas très particuliers comme ceux d'objets apparaissant en silhouette-.

D'une façon générale, une interprétation partielle de l'image peut alléger la tâche de l'opérateur, car le détournement peut s'appuyer sur l'identification de certains éléments et sur une interaction limitée avec l'opérateur, pour reconstituer une occurrence particulière d'un modèle plus général -on parle alors de mécanisme d'inférence ou d'induction-. Enfin de l'occurrence particulière peuvent être extraites des informations grossières sur le contour dans des zones où la segmentation finale rencontrera des difficultés.

1.3.3 Le corpus des photographies d'intérêt général.

Nous avons déjà établi l'importance du type des photos à traiter lors du choix d'une méthode de segmentation.

Nous devons donc essayer de caractériser l'ensemble des photos auquel nous nous intéresserons et que nous appellerons "corpus des photos d'intérêt général". Ceci est d'autant plus délicat que la première caractéristique est justement que cet ensemble de photos présente une variabilité extrême (pas de "standard")

- Contrairement à ce qui est souvent le cas pour l'imagerie scientifique par exemple, les images sont ici, complexes ; c'est à dire qu'il n'y a qu'une corrélation très imparfaite entre la segmentation sur critère local et le détournage cherché : contours manquants et contours intempestifs. **Toute tentative systématique cherchant à réduire le nombre des premiers augmentera le nombre des seconds et inversement.**

- Les photos que nous désirons traiter sont de bonne qualité. Cela implique qu'elles présentent peu de bruit, au sens donné à l'expression en théorie du signal. Beaucoup d'études ont été faites sur les problèmes de segmentation d'images bruitées, mais il ne faut donc pas s'attendre à ce que l'utilisation des techniques mises au point réduise dans une importante mesure les difficultés dues à la structure complexe des images.

- Le système sur lequel nous avons réalisé nos expérimentations nous a imposé de travailler sur des images N et B. Cela ne signifie pourtant pas que l'information de couleur est sans importance pour établir une segmentation, elle est au contraire classiquement reconnue comme assez primordiale /ROSENFELD 82/.

Cependant, au delà des contraintes matérielles, deux remarques permettent de relativiser son importance :

- pour une bonne part du corpus, elle n'est pas disponible !

- elle permettrait assurément de réduire le nombre des erreurs de segmentation sur critère local, mais absolument pas de les éliminer toutes ; de plus le traitement des difficultés en N et B s'étendrait immédiatement au cas de la couleur.

- Les photographies à traiter sont réelles et non synthétiques ; ceci a une grande importance car les zones homogènes n'auront pas une teinte strictement fixée et les limites entre elles ne seront pas parfaitement abruptes mais légèrement progressives.

I.4. Présentation des chapitres suivants .

De multiples aspects du problème posé par l'automatisation même partielle de l'opération de détournage de photographies d'intérêt général, ont été évoqués jusque là.

Dans le chapitre II, nous aborderons les problèmes liés à la segmentation dans le corpus considéré et à la manière de prendre en compte une information de contexte. Nous présenterons un algorithme itératif d'extraction de contour à partir d'une hypothèse supposée extraite du voisinage.

Puis nous présenterons dans le chapitre III une autre approche de la détection de contours : la recherche optimale dans un graphe.

Le chapitre IV est relatif à la modélisation des objets à détourner. On présentera les fonctionnalités que devront remplir les modèles utilisés : représentation des informations a priori ; analyse de l'image pour identifier une occurrence du modèle ; mécanisme d'inférence ou niveau sémantique ; interaction avec l'opérateur.

Nous décrirons dans les chapitres V et VI, deux expérimentations relatives à la modélisation que nous avons réalisées :

. Dans le chapitre IV, on montre comment le "modèle du pantin", permet de représenter l'information a priori sur la

forme du contour d'un personnage.

. Le chapitre VI illustre l'opération d'extraction de primitives pour le modèle de bâtiments.

Le chapitre VII présentera en conclusion, une synthèse de l'ensemble et suggèrera une architecture logicielle pour réaliser un système de détournage.

Bibliographie du chapitre I.

Ch. Boitet

Informatisation de la traduction et traduction automatisée.
"la jaune et la rouge" n°380, Ecole Polytechnique, 1982.

J. Borrel

Pagination and the Printing Industry.
Computer Graphic World, vol 5, n°4, 1982.

M.L. Baird et M.D. Kelly

Recognizing Objects by Rules of Inference on Sequentially
thresholded Gray-level Pictures.
CGIP 3, 1974.

C.G.I.P. Computer Graphics and Image Processing

A. Rosenfeld editr, Academic Press

C.G.W. Computer Graphic World

Publishing Company Inc, San Francisco.

S. Coquillard, M. Gangnet et Haiat

Numérisation par caméra de plans de batiments.
Proceedings MICAD 82.

Ph. Coueignoux

a- Character Generation by Computer.
CGIP 16, 1981.

b- La Reconnaissance des Caractères.
La Recherche, n° 126, Oct 1981.

J.D. Foley et Van Dam

Fundamentals of Interactive Computer Graphics.
Adisson Wesley, 1982.

R.O. Duda et P.E. Hart

Pattern Classification and Scene Analysis.
Wiley-Interscience publications, 1973.

G.H. Granlund et H. Knutsson

Image Enhancement.
in /I.N.R.I.A. 82/

I.N.R.I.A. 82

Support du Cours "Vision par Ordinateur".
Rocquencourt, juin 1982.

- M.D. Kelly
Edge Detection in Picture using Planning.
Machine Intelligence, Tome 6, 1972.
- M. Kocher et M. Kunt
Image Data Compression by Contour-Texture Modeling.
Actes de la conf. Intern. Genève avril 83, SPIE n°397,
à paraître en aout 1983.
- M. Kunt
Coding.
in /I.N.R.I.A. 82/
P.A.M.I. :
- I.E.E.E. Transactions on Pattern Analysis and Machine
Intelligence.
- P.R. : Pattern Recognition,
Pergamon Press.
- T. Pavlidis
Structural Pattern Recognition.
Springer Verlag, 1977.
- J.M. Prager
Extracting and Labeling Boundary Segments in Natural Scenes.
I.E.E.E. PAMI n°1, 1980.
- W.K. Pratt
Digital Image Processing.
Wiley and sons, 1978.
- A.A.G. Requicha
Representation for Rigid Solids: Theory, Methods and Systems.
A.C.M. Computing Surveys, vol12 n° 4, déc. 1980.
- Riseman et Hanson, éditeurs
Computer Vision Systems.
Academic Press, 1978.
- A. Rosenfeld et Kak
Picture Processing by Computer.
Academic Press, 1976.
- A. Rosenfeld
Iterative Methods in Image Analysis.
PR vol 10, 1978.

A. Rosenfeld

Segmentation : Pixel Based Methods.

in /I.N.R.I.A. 82/

The SEYBOLD REPORT on Publishing Systems,
Media, Pennsylvania, U.S.A.

SIGGRAPH

Proceedings of the congress of the ACM Special Interest Group
on Graphics, 1975 ... 1982.

W.F. Schreiber

1974.

W.F. Schreiber

Image Processing for Quality Improvement.

Proc. I.E.E.E. vol 66 n° 12, 1978.

Troxel et Lynn

1978.

Troxel et al.

1980.

S.W. Zucker

Region Growing : Childhood and Adolescence.

CGIP vol 5, 1976.

CHAPITRE 2

=====

Chapitre II : EXTRACTION DE CONTOUR PAR DEFORMATIONS SUCCESSIVES.

=====

II.1. Problèmes rencontrés lors de la segmentation locale.

II.1.1. Carrefour de Bords.

II.1.2. Fuite.

II.1.3. Absence Totale de Bord.

II.1.4. Représentation de l'Information de Contexte.

II.2. Mise en Oeuvre Séquentielle d'une Segmentation sous Guide.

II.3. Méthode des Déformations Successives.

II.3.1. Définition de l'Optimalité.

II.3.1.1. Optimalité dans le Domaine Continu.

II.3.1.2. Contrainte de Continuité.

II.3.1.3. Choix de la fonction de Coût.

II.3.2. Algorithme Itératif de Recherche d'un Optimum Local.

II.3.2.1. Principe d'Itération.

II.3.2.2. Test d'Arrêt.

II.3.2.3. Evaluation du Nombre de Chaînes.

II.3.2.4. Choix d'une Solution Initiale.

II.3.2.5. Premiers Résultats.

II.3.3. Algorithme Modifié.

II.3.3.1. Principe d'Itération.

II.3.3.2. Fonction de Coût.

II.4. Résultats et Discussion.

II.4.1. Détourage semi-automatique.

II.4.2. Limites et Echecs.

II.4.3. Solutions Envisageables.

II.4.3.1. Améliorations des Informations Utilisées.

II.4.3.2. Modifications de l'Algorithme.

II.4.3.3. Modèle de Bord.

Annexe : Un problème de mise en Oeuvre, la Rotation.

Bibliographie.

CHAPITRE II :

EXTRACTION DE CONTOURS PAR DEFORMATIONS SUCCESSIVES.

On a montré dans le chapitre précédent que l'opération de détournage impose d'analyser à plusieurs niveaux de localité, l'image à détourner ; et qu'à un niveau donné, il est nécessaire de **prendre en compte le contexte**, c'est-à-dire une information plus globale.

Dans ce chapitre nous présentons une approche de la segmentation ultime qui permet de tenir compte d'une information de contexte : à savoir, la direction grossière du contour dans un domaine de l'image.

Les diverses techniques proposées ont en commun d'aborder le problème en termes d'optimisation :

Maximiser, le long d'une chaîne de pixels voisins, la probabilité -pour chacun des pixels de cette chaîne- d'appartenir à un contour compatible avec le contexte.

Deux types de méthodes seront proposés : le 1° consiste à procéder par **déformations successives** à partir d'une solution initiale déduite du contexte; cette méthode fera l'objet de ce chapitre :

avant de décrire dans le détail, les mécanismes auxquels nous sommes parvenus (§ II.3), nous présentons les situations-clé qu'ils rencontreront (§ II.1), ainsi qu'une 1° tentative faite dans l'équipe (§ II.2) dont les faiblesses nous ont conduit à la méthode par déformations successives. Le chapitre se terminera par la présentation et la discussion des résultats obtenus (§ II.4).

Dans le chapitre suivant, nous montrerons comment l'extraction de contours peut se ramener au problème de recherche dans un graphe, d'un **chemin optimal** reliant 2 points donnés par le contexte.

II.1. Problèmes rencontrés lors de la segmentation locale.

Les difficultés sur lesquelles bute toute tentative de découpe purement locale, et que la prise en compte du contexte devra permettre de résoudre, se ramènent toutes à l'un des trois types suivants de problèmes :

II.1.1 Carrefour de bords = trop de régions.

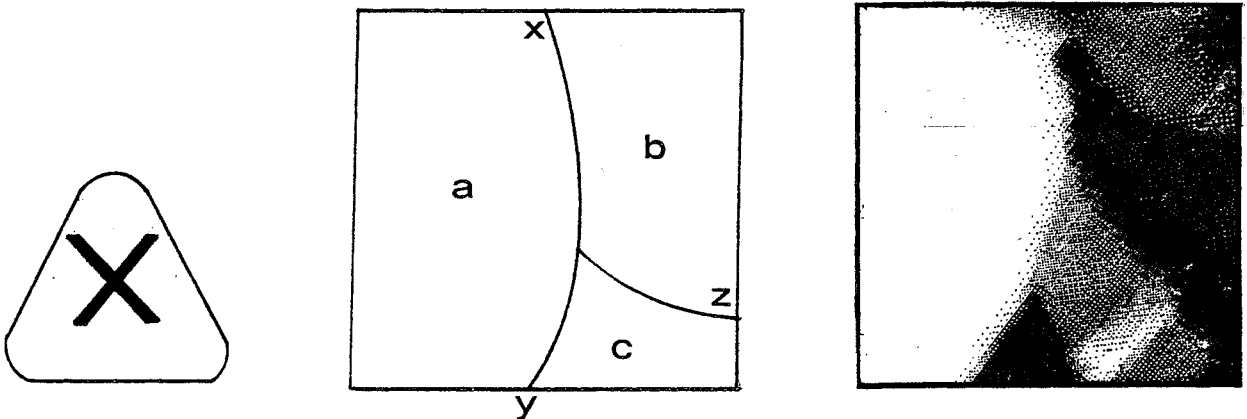


Figure II.1

C'est un cas où la segmentation est satisfaisante (pas d'erreur), mais produit 3 régions ou plus (plusieurs contours) alors que seules 2 étiquettes de régions ('intérieur' et 'extérieur') doivent être attribuées (1 seul contour).

Il est toujours possible de repousser à plus tard la résolution de ce problème, mais pas de le faire disparaître.

Deux indications globales peuvent servir à résoudre cette difficulté :

- la connaissance - a priori ou par référence au contexte - du fait que l'intérieur ou l'extérieur est homogène
- et la direction régionale globale du contour.

A l'aide de l'une de ces deux informations, on pourra sélectionner le contour xy au lieu du contour xz, ou réunir les régions B et C et non pas A et C.

On remarque que le recours à une interprétation de l'image permet de résoudre en général, ce type de difficultés. Dans le cas

plus simple de scènes de blocs polyhédriques, /GUZMAN 68/ montre que l'analyse du graphe reliant les sommets de polygones de l'image permet en général d'interpréter spatialement ceux-ci et donc de déterminer à chaque carrefour les régions "fond" et "objet". Cf. aussi /HUFFMAN 71/

II.1.2. Fuite = pas assez de régions :

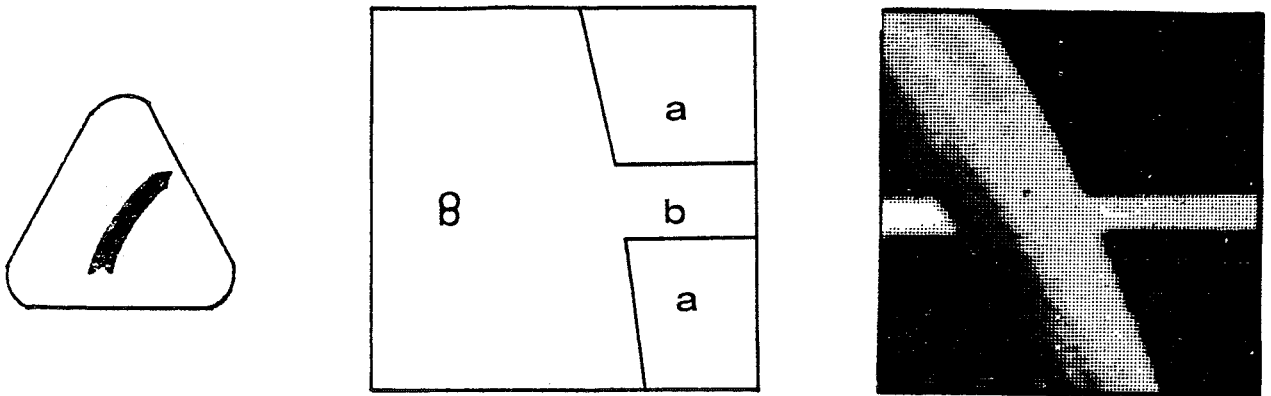


Figure II.2

Contrairement au cas du carrefour, il y a vraiment, ici, une erreur dans la découpe obtenue par segmentation : la région b reçoit à tort la même étiquette que B au lieu de recevoir celle de A.

Une solution peut être d'appliquer un algorithme de dilatation puis d'érosion de zone :

Une telle méthode consisterait ici, à étendre la zone A dans un 1^o temps, en repoussant sa frontière de d (dilatation). On devra ensuite imposer sur la zone dilatée, un retrait de ses frontières de la même valeur d (érosion).

Pour la Dilatation-Erosion, l'application "d'opérateurs morphologiques" est particulièrement bien adaptée. Pour l'application de la morphologie mathématique au traitement des images, voir /SERRA 83/ et /SERRA 82/.

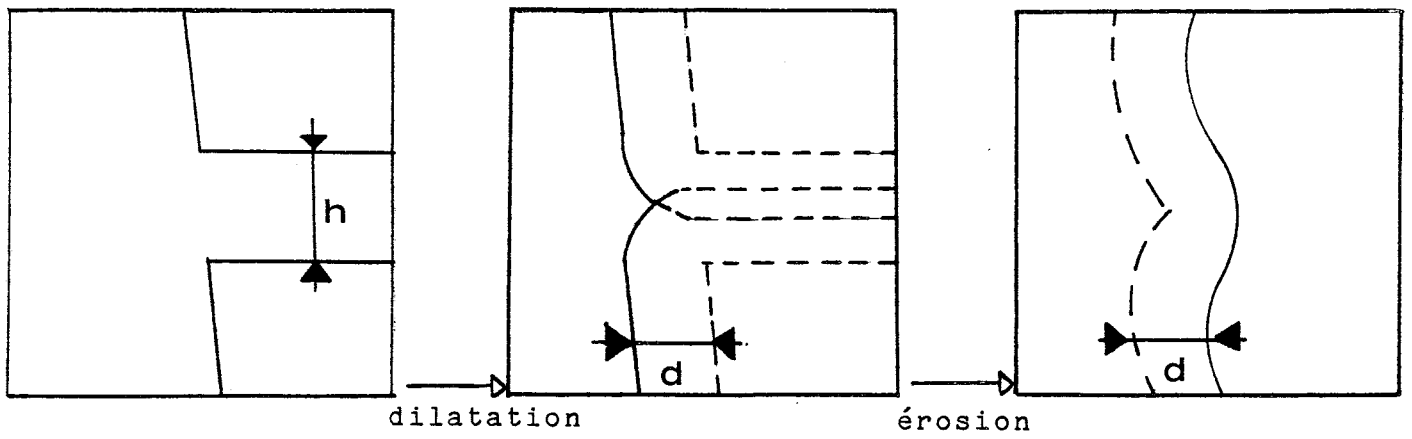


Fig II.2.bis

Il apparait clairement que pour résoudre une fuite de largeur h par un tel moyen, la valeur de d doit être supérieure à $h/2$; mais d ne peut pas augmenter sans risque pour l'une des régions de disparaître.

Exemple :

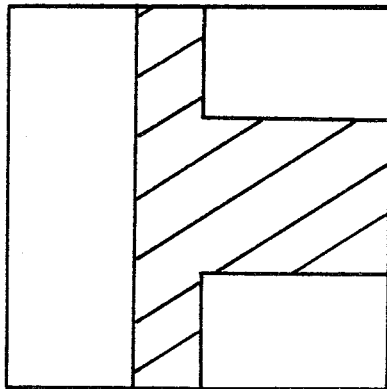


Fig II.2.ter

II.1.3. Absence locale de bord :

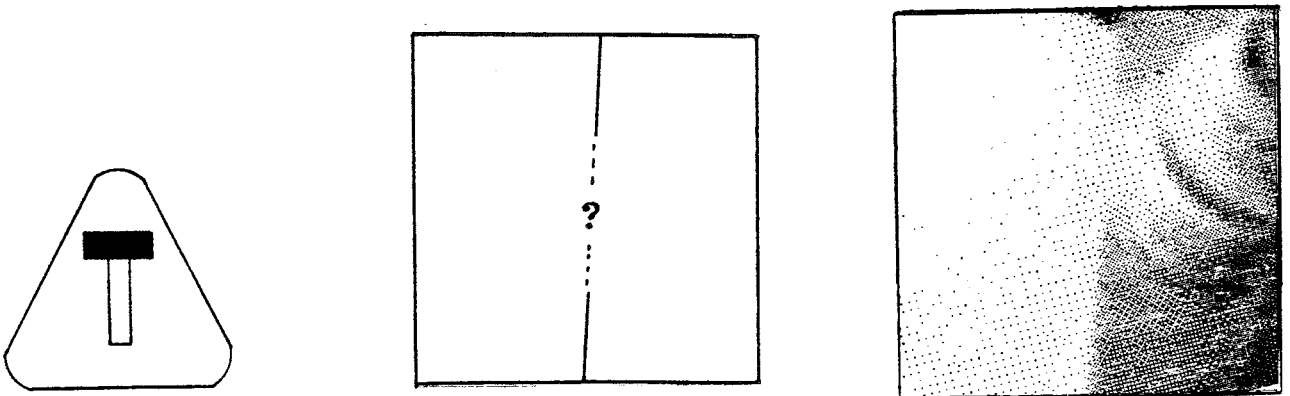


Figure II.3

Cette difficulté est le cas limite de la précédente : la segmentation, au lieu de donner un résultat même incompatible avec le contexte, n'aboutit à **aucune découpe**. La dilatation-érosion n'est donc pas praticable. Reste encore la possibilité d'accepter telle quelle, l'indication de direction fournie par le contexte.

II.1.4. Représentation de l'information de contexte.

Les trois difficultés précédentes sont intrinsèques à l'image en ce sens qu'elles apparaissent quelle que soit l'approche utilisée, contours ou régions, séquentielle ou parallèle. Dans ces trois cas la notion de direction guide est primordiale pour résoudre les difficultés.

Le moyen le plus adapté pour représenter l'information issue d'une analyse globale de l'image, qu'elle soit automatique, interactive, ou totalement laissée à l'opérateur, est d'indiquer la segmentation grossière par une ligne polygonale.

Chaque segment détermine :

- l'étendue de la région où appliquer la segmentation locale,
- et la direction régionale approximative du contour à extraire, dans cette région.

Le domaine de segmentation est un rectangle dont le segment donné - que nous appellerons 'segment-guide' - est une médiane. L'autre dimension du domaine peut être fixée par référence à l'échelle de représentation de l'objet à détourer.

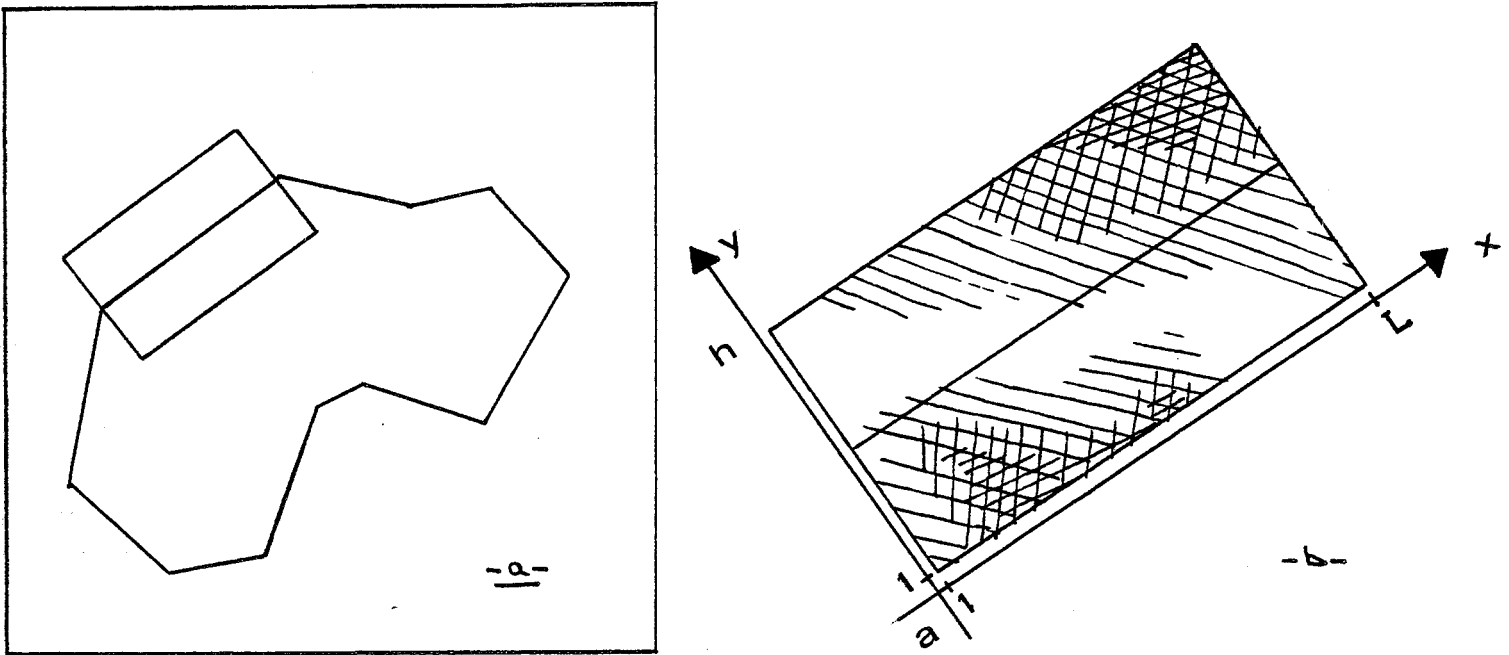


Figure II.4 - Polygone des segments guides

II.2. Mise en oeuvre séquentielle d'une segmentation sous guide.

L'information la plus indispensable à extraire du contexte pour résoudre les problèmes locaux, est donc la **direction globale approximative du contour** dans chaque domaine.

Jusqu'ici rien ne disqualifie "l'extraction de contour" ou "la croissance de zones", mais il semble plus direct de manipuler l'information de direction en terme de contour plutôt qu'en terme de zones et donc de limites de celles-ci.

Disposant dans chaque domaine d'un segment pour guider la segmentation, on est assez naturellement conduit, à tenter d'utiliser un algorithme de **suiti de contour** à partir d'une extrémité du segment guide jusqu'à atteindre l'autre extrémité.

A partir d'un point de départ donné par le contexte, on passe de pixel en pixel voisin en sélectionnant dans le voisinage le point le meilleur au sens d'un critère faisant intervenir la direction et l'intensité de bord des deux pixels.

Dans cette optique, une tentative à été faite dans notre équipe par Didier Perny en 80 qui proposait l'algorithme suivant :

(* on dispose d'une pile d'enregistrements
composés d'un numéro d'étape, de l'identification
d'un pixel, et d'une évaluation du pixel *)

début

```
    fini := faux
    n°étape := 1
    éval := 0
    pixel := 1° extrémité du segment-guide
    empiler (n°étape, pixel, éval)
    tant que pas fini, faire
        dépiler (n°étape, pixel, éval)
        si pixel dans ensemble des points-cible
            alors fini := vrai
        sinon
            n°étape := n°étape + 1
            pour i de 1 à 7
                nouvpix := i°voisin (pixel)
                nouvéal := f (éval, nouvpix)
                empiler (n°étape, nouvpix, nouvéal)
            finpour
        finsi
    finfaire
fin.
```

Bien entendu cette recherche arborescente ne peut pas être menée exhaustivement, et impose de limiter à une dizaine le nombre de niveaux de récursion, et ainsi, les problèmes mentionnés plus haut ne manquent pas d'apparaître à nouveau : par exemple, la fuite

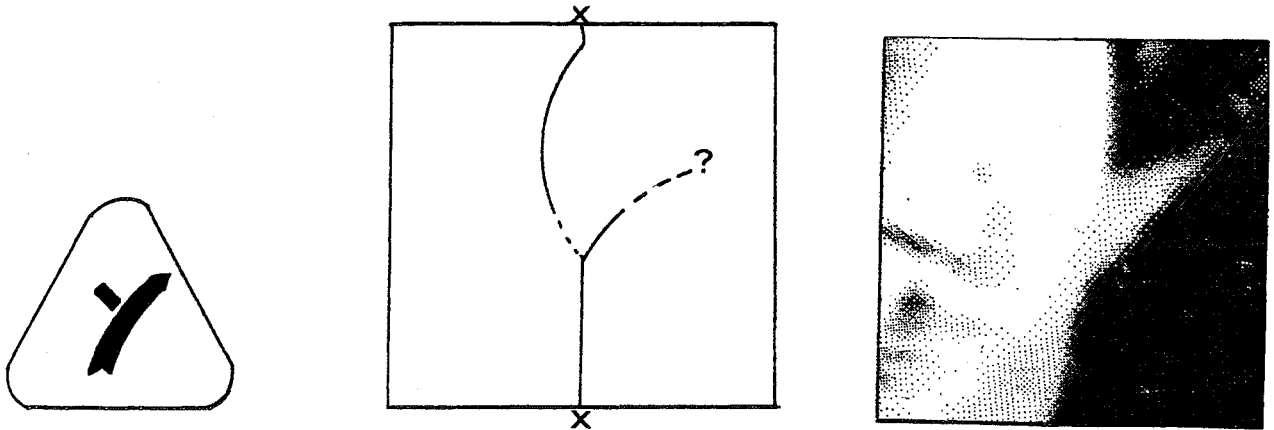


Figure II.5

Il est théoriquement possible de palier à cette difficulté en travaillant successivement à des résolutions de moins en moins grossières.

II.3. Extraction de contour en parallèle avec prise en compte du contexte : méthode par déformations successives .

II.3.1. Définition de l'optimalité.

II.3.1.1. Optimalité dans le domaine continu.

Dès le début de ce chapitre, nous avons formulé le problème en termes d'optimisation dans un domaine discret (l'ensemble des pixels). Avant de revenir en détail sur cette formulation, il est intéressant de voir comment se formulerait le problème dans le cas continu.

En effet, l'image réelle n'est connue qu'aux sommets d'une grille d'échantillonnage de pas fini, mais on peut trouver par une méthode d'interpolation, une fonction F régulière, ayant les mêmes valeurs sur les points de la grille et qui représente tout aussi bien l'image réelle que la représentation numérique.

F étant au moins dérivable, son gradient est défini en tout point de R^2 et constitue un champ vectoriel.

$$\begin{array}{l} F \quad : \quad R^2 \longrightarrow R^+ \\ \overrightarrow{\text{grad}} F \quad : \quad R^2 \longrightarrow R^2 \end{array}$$

Dans cette formulation, une portion de contour joignant 2 points A et B , est un arc de R^2 , sans point double, que l'on peut représenter paramétriquement :

$$\text{GAMMA} = \left\{ (x(s), y(s)), s \text{ abscisse curviligne le long de GAMMA} \right\}$$

En chaque point d'un tel arc, le produit scalaire du vecteur (unitaire) normal à l'arc et du gradient, mesure la manière de laquelle l'arc épouse localement un contour.

Or par hypothèse de régularité sur F , son gradient est continu, d'où la possibilité de définir la valeur de coût associée à un arc comme suit :

$$C(\text{GAMMA}) = \int_{\text{GAMMA}} \overrightarrow{n} \cdot \overrightarrow{\text{grad}} F \, ds$$

Dans ce cadre, le problème d'optimisation posé initialement consiste à rechercher parmi tous les arcs du domaine, celui pour lequel C est maximum.

Pour poursuivre dans cette voie, les outils mathématiques de la géométrie différentielle, nous font défaut.

Intuitivement, on imagine que l'application à ce problème d'une méthode itérative d'optimisation, de type Newton par exemple, nécessiterait de définir mathématiquement la déformation d'un arc. Il faudrait alors pouvoir évaluer l'effet sur la

fonction de coût d'une déformation élémentaire.

C'est cette intuition qui est sous-jacente à la méthode proposée dans la suite.

II.3.1.2. Contrainte de continuité

Nous formulons ici en termes d'optimisation, la recherche d'un contour reliant l'un à l'autre les deux côtés du domaine rectangulaire discret, situés aux extrémités du segment-guide.

L'extraction du contour consiste à chercher la liste de points du domaine qui maximise la probabilité pour chacun d'eux d'appartenir à un contour, sous la contrainte de continuité suivante :

Nous appelons 'chaîne de pixels', une liste de points admissible comme solution ; elle répond aux propriétés suivantes :

- elle est formée de L points, 1 pour chaque abscisse dans le repère xAy , (fig. II.4.b)
- toutes les ordonnées dans ce même repère sont comprises entre 1 et h .
- deux pixels d'abscisses consécutives appartiennent au voisinage l'un de l'autre.

Il est clair qu'une telle contrainte d'optimisation, introduit dans la solution que l'on obtiendra, une partie de l'information de contexte ; mais elle aura également pour effet de limiter la complexité au niveau local des circonvolutions des contours.

De plus, cette formulation de l'extraction de contour nécessite que l'on accède aux pixels par leurs coordonnées dans un repère précis lié au segment-guide. Le problème de rotation d'une fenêtre d'image que cela pose, sera abordé en annexe à ce chapitre (p. 77).

II.3.1.3. Choix de la fonction de coût.

La probabilité pour qu'un point du domaine appartienne à un contour, est très directement liée à l'intensité locale de bord,

ou amplitude locale du gradient de luminosité. Cette grandeur peut être classiquement calculée par application d'un opérateur de bord(*). On obtient même ainsi la direction locale du bord, dont il pourrait être intéressant de tenir compte ici.

En fait il nous a semblé suffisant et économique en temps de calcul d'utiliser comme probabilité en chaque point, la valeur absolue de la composante transversale du gradient. Cette simplification est en plus, particulièrement adaptée puisqu'elle fait intervenir implicitement la direction locale et tend à favoriser la direction à suivre globalement.

A partir de là, la fonction de coût la plus facile à mettre en oeuvre, est le cumul le long d'une chaîne de pixels de l'intensité latérale de bord.

$$(1) \quad F = \sum_i p_b(i, j(i))$$

où $(i, j(i))$ pour $i=1, L$ sont les pixels d'une chaîne.

et $p_b(x,y)$ la probabilité que (x,y) soit sur le contour.

II.3.2. Algorithme itératif de recherche d'un optimum local.

Diverses méthodes itératives d'optimisation pouvaient être employées ici :

- * Soit, trouver pour chaque abscisse, l'ordonnée du point de plus forte intensité de bord ; puis modifier progressivement la liste de points obtenue jusqu'à rétablir la continuité, tout en diminuant le moins possible, la valeur de la fonction.
- * Soit, inversement, s'assurer au départ de ce que la contrainte de continuité soit vérifiée, puis modifier itérativement la chaîne des pixels pour augmenter le cumul des intensités de bord tout le long.

(*) Les méthodes de calcul de gradient d'une image en teintes de gris seront abordés plus loin : cf. chapitre VI. et la bibliographie que nous citerons alors.

Cette dernière approche nous a paru plus simple puisqu'elle assure le caractère admissible sinon optimal d'une solution partielle et c'est elle que nous avons retenue.

II.3.2.1. Principe de l'itération

Dans une chaîne de pixels qui vérifie la contrainte de continuité, chaque point peut ou non être déplacé seul, sans rompre la chaîne, suivant la position relative de ses 2 voisins ; Par symétrie on ramène à 4, les cas qui peuvent se présenter :

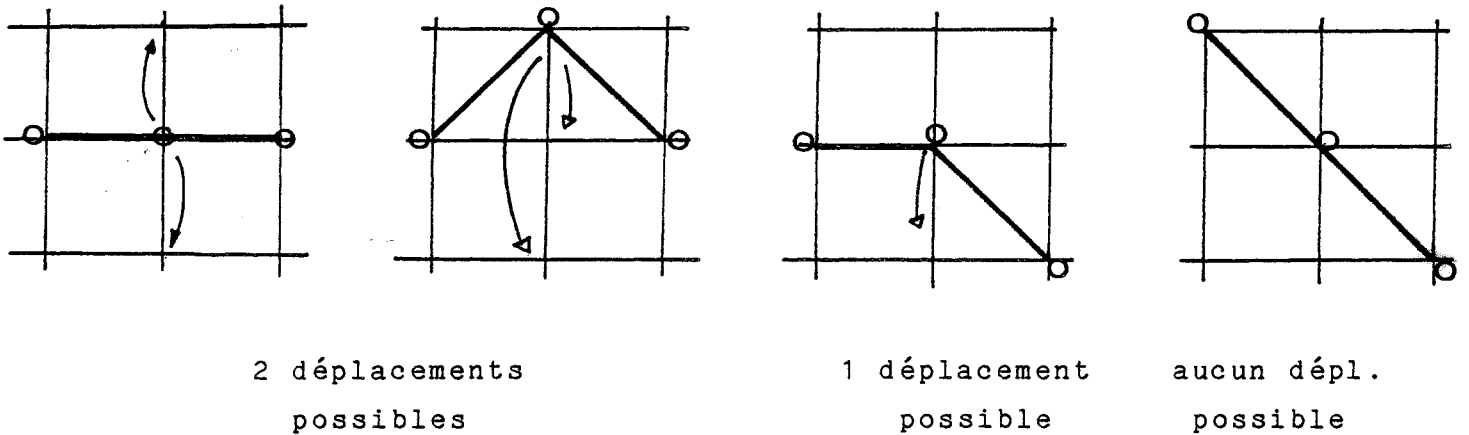


Figure II.6.

Pour chacun des déplacements individuels possibles à partir d'une situation donnée, on peut calculer l'amélioration apportée à la fonction à optimiser.

Dès lors, une itération consiste à trouver l'amélioration maximale qu'il est possible d'obtenir par un déplacement unique, et à effectuer ce déplacement.

On constate alors qu'à chaque itération, la liste des déplacements autorisés et des gains associés, en chaque point, est très légèrement modifiée. Plus précisément, à chaque itération, la première exceptée, la mise à jour ne concerne que trois points.

ALGORITHME D'EXTRACTION DE CONTOUR PAR DEFORMATIONS SUCCESSIVES

version 1

```
début
(* SOLUTION INITIALE *)
  pour i de 1 à L, faire
    sol(i) := n/2
  finpour
(* VALEUR INITIALE DE LA FONCTION DE COUT *)
  coût := 0
  pour i de 1 à L, faire
    coût := coût + bord (i,sol(i))
  finpour
(* CALCUL DES DEPLACEMENTS QUI AMELIORENT LA FONCTION DE COUT *)
  pour i de 1 à L, faire
    (* calcul du meilleur déplacement en i *)
    déplac(i) := ...; gain(i) := ...;
  finpour

(* ITERATION JUSQU'A UN OPTIMUM LOCAL *)
  fini := faux
  tant que pas fini, faire
    imax := max (gain(1..n))
    si imax = 0
      alors fini := vrai
    sinon
      (* EFFECTUER LE MEILLEUR DEPLACEMENT *)
      sol(imax) := sol(imax) + déplac(imax)
      coût := coût + gain(imax)
      (* MISE A JOUR DES DEPLACEMENTS POSSIBLES *)
      déplac(imax-1) := ...; gain(imax-1) := ...;
      déplac(imax) := ...; gain(imax) := ...;
      déplac(imax+1) := ...; gain(imax+1) := ...;
    finsi
  fintantque
fin.
```

II.3.2.2. Test d'arrêt

Il est clair qu'il est nécessaire de terminer l'itération dès qu'à tous les déplacements possibles, sont associées des **diminutions du critère** en chaque point : Si l'on ne l'interrompait pas explicitement, le processus itératif "bouclerait" à partir de ce point, en effectuant successivement deux mouvements opposés.

En aucun cas, on est assuré en fin d'itération, d'avoir atteint la situation donnant la valeur maximale de la fonction économique définie pour toutes les "chaînes" du domaine rectangulaire ; on a simplement atteint un **optimum local** en ce sens que toutes les chaînes voisines - i.e. pouvant être atteintes par déplacement d'un seul pixel sans provoquer de rupture -, donnent de moins bonnes valeurs à la fonction.

Deux considérations peuvent en pareil cas, intervenir : soit l'on prend soin de partir d'une solution initiale bien choisie, soit l'on entreprend une recherche arborescente de l'optimum.

La recherche exhaustive de la meilleure chaîne dans un domaine $L \times h$ pourrait conduire à évaluer la fonction de coût sur toutes les chaînes. Ce cas constitue aussi le plus mauvais cas si l'on s'arrête sur un optimum local.

II.3.2.3. Evaluation du nombre de chaînes.

Soit $Ch(h,L)$, le nombre de chaînes d'une grille $h \times L$, et soit $P(i,L)$, le nombre de chaînes d'origine i ,

on a :

$$Ch(h,L) = \sum_{i=1}^h P(i,L)$$

et on peut établir sur P et Ch , la relation de récurrence suivante:

$$Ch(h,L) = 3.Ch(h,L-1) - 2.P(1,L-1) \quad (1)$$

d'où

$$Ch(h,L) = 3^{L-1}.Ch(h,1) - 2(P(1,L-1) + 3.P(1,L-2) + \dots + 3^{L-2}.P(1,1))$$

Or,

$$\text{Ch}(h,1) = 3.(h-2) + 4 = 3.h-2$$

d'où l'expression donnant le nombre de chaînes :

$$\text{Ch}(h,L) = 3^{L-1}.(3h-2) - 2.(P(1,p-1)+3.P(1,p-2)+\dots+3^{L-2}.P(1,1))$$

On obtient ainsi en minorant le 2° terme ($P(1,i) \gg P(1,1)$), une majoration de $\text{Ch}(h,L)$:

$$\text{Ch}(h,L) \ll M = 3^{L-1}.(3h-4) + 2$$

Pour une minoration grossière, il suffit de compter le nombre de chaînes que l'on aurait si chaque sommet n'avait que 2 successeurs (au lieu de 3 pour la plupart) :

$$\text{Ch}(h,L) \gg m = h.2^L$$

$$h.2^L \ll \text{Nombre de chaînes} \ll 3^{L-1}.(3h-4) + 2$$

Vu que L et h prennent typiquement des valeurs de l'ordre de 100 et 30, une recherche exhaustive semble tout à fait irréaliste.

La difficulté à résoudre est l'analogie dans le cas parallèle de celle signalée au § II.2. La même remarque s'applique dans la mesure où l'on peut songer à des échelles de plus en plus fines.

En fait nous montrerons dans le chapitre III, que la complexité du problème n'est qu'apparente dans la formulation précise que nous retenons ici.

II.3.2.4. Choix d'une solution initiale.

Le choix qui vient en premier à l'esprit, pour construire une solution initiale est de partir du segment guide :

$$\{ (x, y(x)) / x=1 \text{ à } 1, \text{ et pour tout } x, y(x)=h/2 \}$$

Une autre solution qui tient mieux compte d'une possible imprécision latérale dans la localisation du segment guide consiste à réaliser une première recherche optimale parmi tous les segments rectilignes du domaine, parallèles au segment guide.

Le choix du 'segment-optimal' obtenu dans cette première étape comme point de départ de l'étape principale, assure d'atteindre en fin de compte une assez bonne solution sous-optimale mais toujours pas la meilleure.

II.3.2.5. Premiers résultats :

Les premiers résultats obtenus en programmant cet algorithme, ont été assez décevants : les situations sous-optimales auxquelles le programme aboutissait, s'éloignaient, en général beaucoup du contour que l'on aurait souhaité atteindre. Plus précisément, l'algorithme semblait s'arrêter trop tôt : l'itération s'interrompait très souvent, bien avant d'épouser un contour marqué.

L'analyse minutieuse de ces résultats déficients en a fait apparaître la raison :

Pour chaque abscisse, lorsque le point correspondant dans la chaîne des pixels, a atteint un maximum local d'intensité de bord, le processus est **définitivement bloqué en ce point**, même si ce maximum est très local et très peu accentué. Rapidement à cause de la contrainte de continuité, le blocage en un point se propage de part et d'autre de celui-ci même si les points voisins sont susceptibles d'améliorer beaucoup la valeur globale de la fonction de coût.

II.3.3. Algorithme modifié.

II.3.3.1. Principe de l'itération.

Une solution pouvant être adoptée, consistait à accepter une petite diminution de la valeur globale de la fonction pendant une ou quelques itérations; mais ceci revenait à mettre en oeuvre une recherche en arbre partielle et risquait de devenir coûteux.

La solution choisie a consisté à modifier et à élargir la notion de voisinage introduire dans l'ensemble des 'chaînes' d'un domaine rectangulaire (chaînes pouvant être atteinte en 1 itération).

Dans une situation donnée, on s'autorise à déplacer tous les points d'une case vers le haut ou d'une case vers le bas, quelle que soit la position des voisins. On permet donc les déplacements qui rompent la continuité, mais on rétablit aussitôt celle-ci en déplaçant autant de voisins successifs que nécessaire (figure II.8.).

Le calcul de la fonction de coût cumule évidemment l'effet des déplacements multiples associés au déplacement initial.

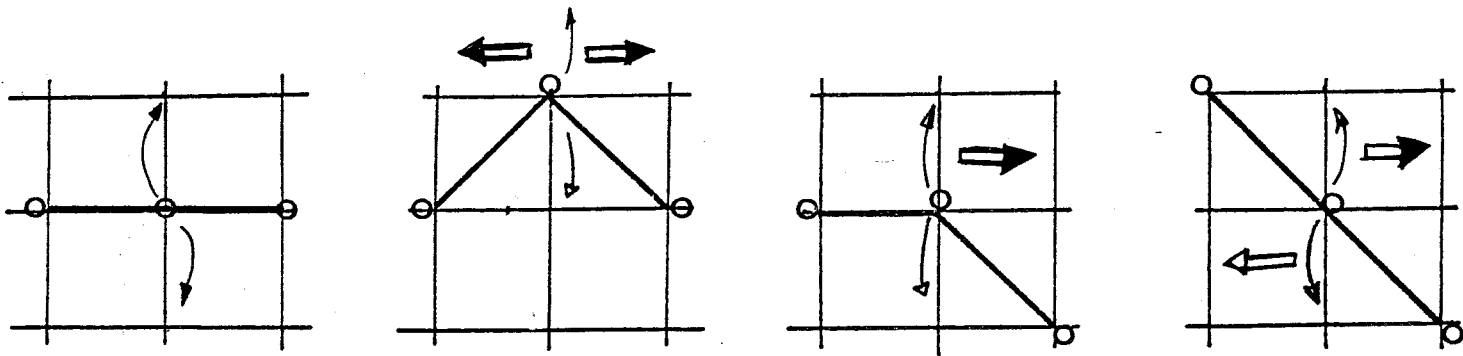
A chaque itération, la mise à jour des gains des déplacements possibles et des gains associés à chaque point, doit concerner tous les points déplacés + 2 (1 à chaque extrémité du segment modifié).

On obtient ainsi l'algorithme présenté page suivante.

ALGORITHME D'EXTRACTION DE CONTOUR PAR DEFORMATIONS SUCCESSIVES
version 2

```
début
  pour i de 1 à L, faire                                (* SOLUTION INITIALE *)
    sol(i) := n/2
  finpour
  coût := 0      (* VALEUR INITIALE DE LA FONCTION DE COUT *)
  pour i de 1 à L, faire
    coût := coût + bord (i,sol(i))
  finpour
(* CALCUL DES DEPLACEMENTS QUI AMELIORENT LA FONCTION DE COUT *)
  pour i de 1 à L, faire
    (* calcul du meilleur déplacement en i *)
    déplac(i) := ...; gain(i) := ...;
    début(i) := ...; fin(i) := ...;
  finpour

  fini := faux    (* ITERATION JUSQU'A UN OPTIMUM LOCAL *)
  tant que pas fini, faire
    imax := max (gain(1..n))
    si imax = 0
      alors fini := vrai
    sinon    (* effectuer le meilleur déplacement *)
      pour i de début(imax) à fin(imax), faire
        sol(i) := sol(i) + deplac(imax)
      finpour
      coût := coût + gain(imax)
      (* mise à jour *)
      pour i de début(imax) à fin(imax), faire
        déplac(i) := ...; gain(i) := ...;
        début(i) := ...; fin(i) := ...;
      finpour;
    finsi
  fintantque
fin.
```



aucune
répercussion

répercussion à
droite et à gau
che dans l'un
des cas, pas de
répercussion
dans l'autre.

répercussion
d'un côté dans
un cas pas de
répercussion
dans l'autre

répercussion
d'un côté dans
un cas, de l'au
tre côté dans
l'autre cas

Figure II.8

II.3.3.2. Fonction de coût.

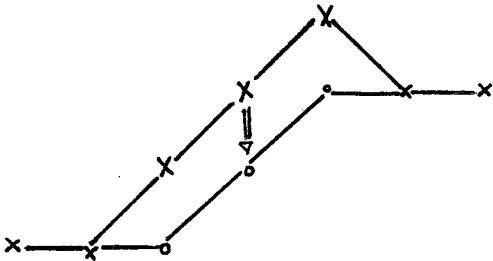
On a d'abord utilisé l'expression (1) du §II.2.1. pour la fonction économique. Elle a l'inconvénient d'aboutir dans certaines conditions à des **contours bruités, c'est-à-dire, inutilement irréguliers** : Les dernières itérations n'améliorent que faiblement la fonction et peuvent le faire au détriment de l'aspect global du contour. Ceci est particulièrement sensible dans les zones de contour peu marqué.

La solution à ce problème consiste à introduire dans l'expression de la fonction, la longueur de la chaîne. Celle-ci varie pour chaque déplacement possible, de l'une des 3 valeurs suivantes :

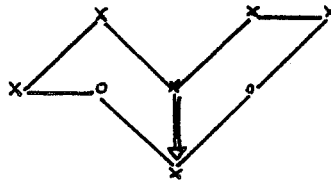
$$2.(\sqrt{2}-1) , 0 , -2.(\sqrt{2}-1)$$

par exemple :

$$\Delta l = 2 \cdot (\sqrt{2}-1)$$



$$\Delta l = 0$$



$$\Delta l = 2 \cdot (\sqrt{2}-1)$$

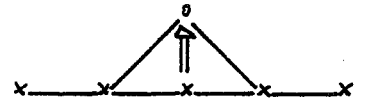


Figure II.9

L'expression
$$\sum p_b (i, j(i))/\text{Long}(\text{chaîne}) \quad (2)$$
 exprime la probabilité moyenne de présence de bord par unité de longueur, le long de la chaîne. Cette expression semble a priori une mesure judicieuse de la précision avec laquelle une chaîne de pixels épouse un contour. Cependant, elle ne peut pas être utilisée ici comme le montre le calcul suivant.

Soient :

- L la longueur de la chaîne avant l'itération considérée,
- C le cumul des intensités latérales de bord le long de cette chaîne et
- dL ,dC les variations de ces 2 grandeurs pour un déplacement.

Le déplacement effectué doit vérifier :

$$(3) \quad \frac{C+dC}{L+dL} > \frac{C}{L} = \bar{C} \iff dC > \frac{C \cdot dL}{L} = \bar{C} \cdot dL$$

Or cette condition pénalise beaucoup trop fortement les déplacements qui rallongent la chaîne puisque C doit augmenter de presque autant que \bar{C} (pour $dL=2\sqrt{2}-1 \approx 0.8$).

L'expression $\sum_i p_b(i, j(i)) - k \times \text{long}(\text{chaîne})$ (4)

donne des résultats plus satisfaisants, mais nécessite de régler le coefficient de pondération k .



$k = 0.1$



$k = 10$



$k = 30$

Figure II.10

II.4. Résultats et discussion

II.4.1. Détourage semi-automatique.

Dans la plupart des 'bonnes situations", les résultats obtenus sont satisfaisants. A partir de l'indication de la segmentation grossière, le programme extrait avec une bonne précision, des contours moyennement complexes et intenses (cf. fig. II.11).

L'utilisation la plus immédiate de l'algorithme présenté consiste à l'introduire à l'intérieur d'un programme de détourage 'manuel', c'est-à-dire sous le contrôle duquel un opérateur introduit point à point le contour qu'il désire définir ; un tel programme travaille de façon totalement 'aveugle' puisqu'il ne consulte pas les données de l'image, et réalise une simple interpolation entre les points introduits.

L'apport de notre algorithme en lieu et place de cette interpolation, est double :

- Il permet un important allègement de l'interaction entre l'opérateur et le programme puisque le pas entre deux points successifs n'est pas limité par la distance en-dessus de laquelle l'interpolant et le contour vrai différent sensiblement, mais par la taille des régions dans lesquelles la direction grossière du contour est constante.

Il est difficile de chiffrer la réduction du nombre de points à introduire car elle varie beaucoup selon les images, mais le facteur de réduction est - dans la majorité des cas - bien supérieur à 2 (fig. II.12).

- Le deuxième bénéfice concerne la qualité du résultat : notre algorithme apporte une amélioration notable en supprimant l'effet désagréable de 'coup de ciseaux', qui apparaît dans certaines zones d'images présentant un bord franc mais trop finement découpé pour pouvoir être suivi manuellement.

Extraction de Contours par Déformations successives.

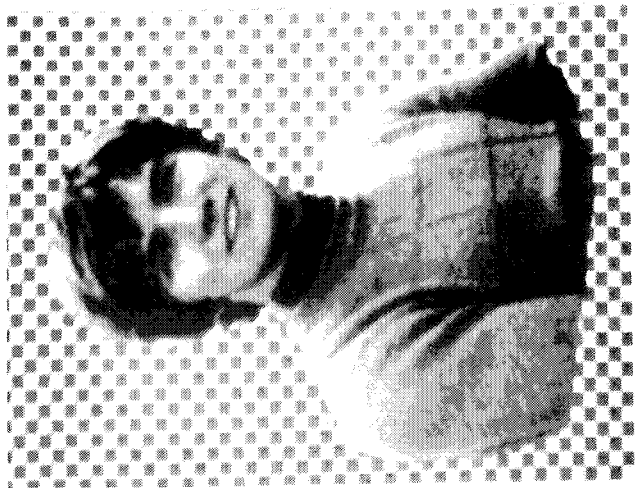
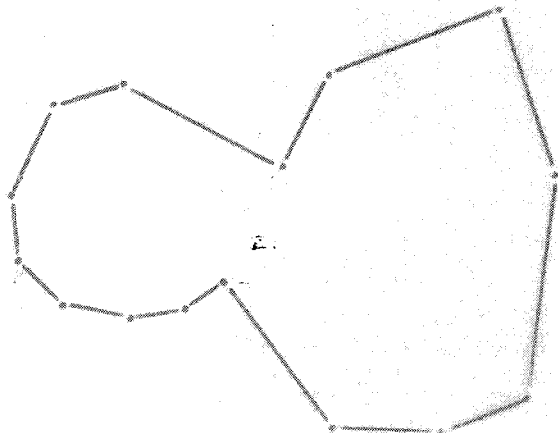
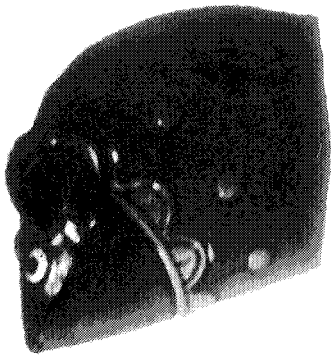
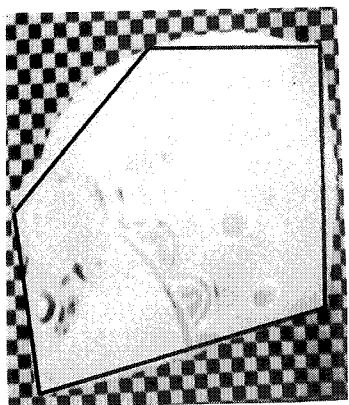
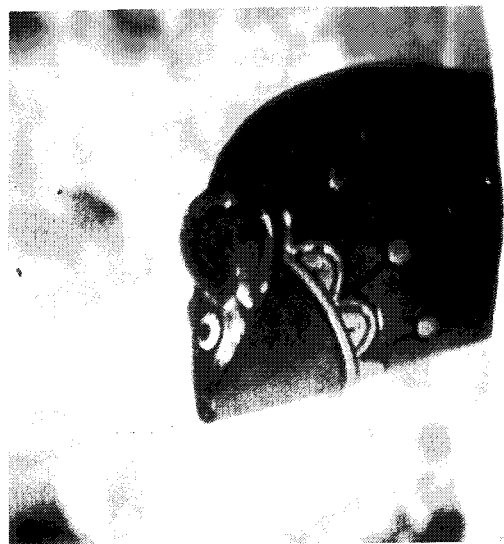
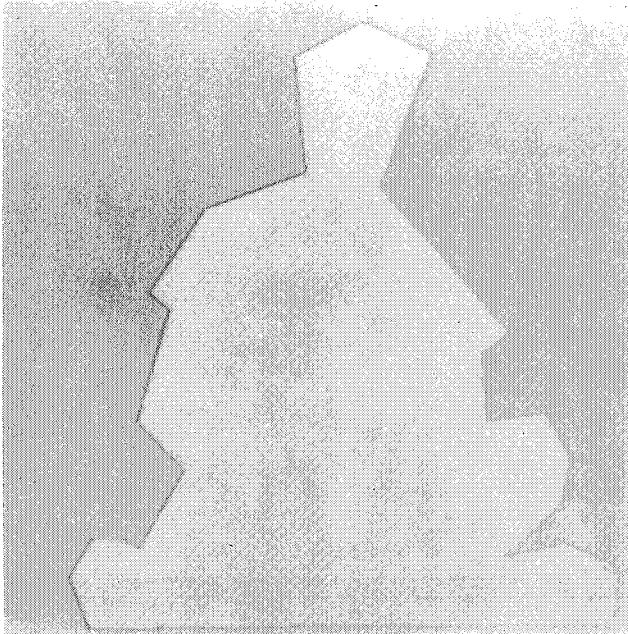
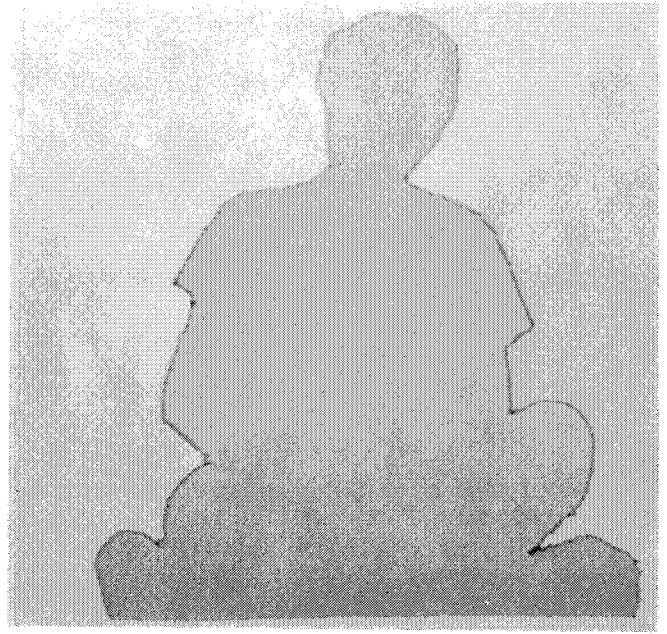


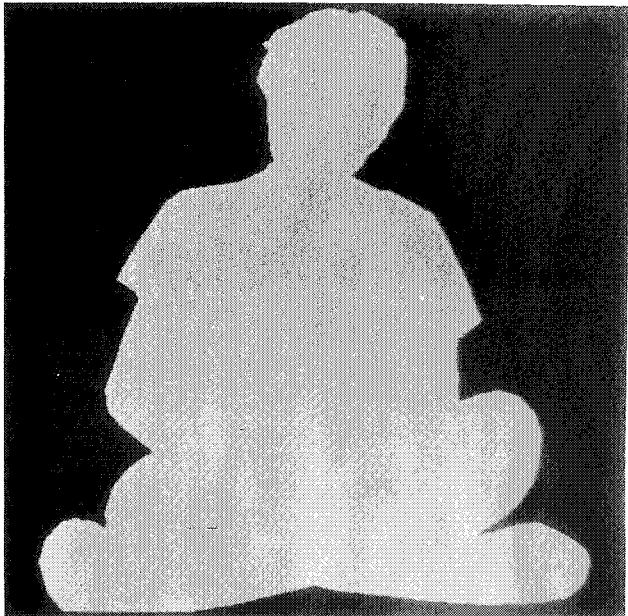
Figure II.12.



-a-



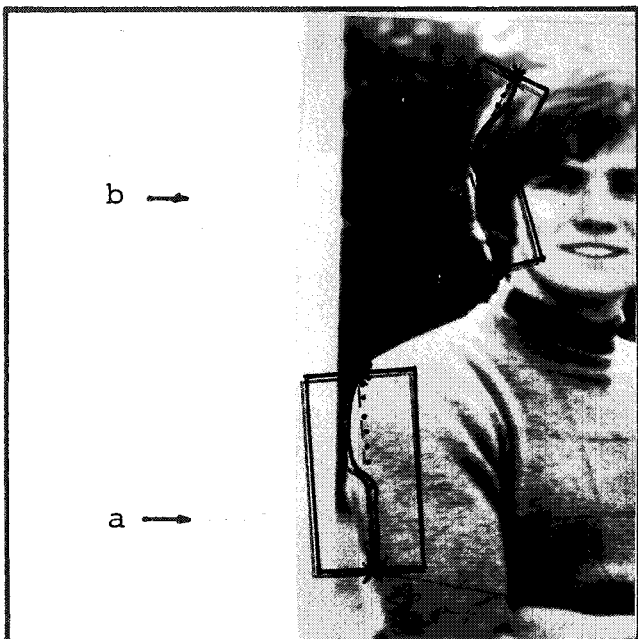
-b-



-c-



-d-



-e-

Figure II.13.

A ce stade l'algorithme est bien adapté au détournage semi-manuel de photographies prises **en prenant soin qu'elles constituent de 'bonnes conditions'** . Cette contrainte n'est pas aberrante et au contraire, elle correspond fréquemment au souci général des photographes de mettre en valeur leur sujet.

II.4.2. Limites et échecs

La segmentation 'semi-manuelle' permet évidemment le retour en arrière sur un pas de détournage jugé non satisfaisant par l'opérateur. Il est cependant important de citer et d'analyser les erreurs qui peuvent apparaître avec l'algorithme décrit, pour certaines situations.

Il n'est pas surprenant que ces situations soient en général, d'une part celles où le contour à suivre est très peu marqué et d'autre part celles où le segment guide s'en éloigne trop.

Mais si l'on analyse en détail, les cas d'erreurs, le comportement de l'algorithme par déformations successives, trois types d'échecs fondamentalement différents peuvent se présenter :

- 1°) L'algorithme itératif s'arrête sur un 'optimum local' très différent de l'optimum global (fig. 13.a). Ce type d'erreur se produit en particulier lorsque le segment guide est la corde d'un contour souhaité largement incurvé.
- 2°) Le contour que l'on désirerait obtenir, ne peut pas l'être car il ne constitue pas une 'chaîne de pixel' dans le sens où nous l'avons entendu. Ce type de problème est irrémédiablement lié à la nature de la méthode, puisque c'est la double contrainte du mécanisme d'optimisation qui est en cause : contrainte de continuité par 8-voisinage et solution de type $(i, j(i))$ pour tout $i=1,1$ (nécessité que $j(i)$ soit une fonction dans le repère lié au segment-guide).
- 3°) Le troisième cas d'erreur semble lui aussi remettre en cause gravement la méthode : L'optimum global ayant été correctement extrait dans le domaine, il s'avère parfois que celui-ci ne correspond pas à ce que l'opérateur reconnaît comme contour (fig. 13.b.).

II.4.3. Solutions envisageables

L'énoncé des points d'achoppement de la méthode par déformations successives, suggère plusieurs manières d'y remédier :

- Il est possible d'améliorer les résultats obtenus,
- soit en s'assurant d'une meilleure fiabilité des informations sur lesquelles l'algorithme se fonde,
 - soit en modifiant l'algorithme.

II.4.3.1. Améliorations des informations prises en compte.

La première de ces améliorations consiste à modifier le calcul de la probabilité de présence d'un bord, en chaque point, en substituant au calcul de bord d'intensité lumineuse, un calcul de **gradient de texture**. Des méthodes ont été proposées /DAVIS et MITICHE 80/ /LOWITZ 81/, et devraient permettre, mais au prix d'une augmentation notable du temps de calcul, une réduction des cas d'absence de bord.

Les autres améliorations consisteraient à augmenter l'information extraite du contexte.

- la réduction de la dimension latérale du domaine de segmentation en est un exemple puisqu'elle limite les variations autour du segment guide c'est-à-dire assure une plus grande fiabilité dans celui-ci. On peut résoudre les problèmes de type 3) ci-dessus - ou plus exactement les reporter au niveau de l'exploitation plus fine du contexte -, en réduisant la fenêtre à un domaine où le bord cherché est effectivement l'optimum global de la fonction économique.
- La principale information extraite du contexte est matérialisée par ce que nous avons appelé le segment-guide ; En fait, cette notion pourrait être étendue, et le segment guide extrapolé en arc-guide pour résoudre de façon simple certaines erreurs de type 1, en partant d'une solution initiale plus proche du segment guide.

- Enfin, on a introduit plus haut (II.2.3.2.) un coefficient de pondération dans la fonction de coût ; il devrait être possible - c'est en tous cas ce dont est capable l'opérateur - de modifier ce coefficient par analyse globale de l'image... et **interpolation** de celle-ci.

II.4.3.2. Modifications de l'algorithme par déformations successives.

Une première modification décrite précédemment (II.2.3.1), a consisté à élargir la notion de voisinage dans l'ensemble des 'chaînes de pixels' définies sur un domaine.

D'autres modifications de ce type seraient peut être souhaitables pour résoudre les problèmes de type 2).

On peut tout simplement envisager de réduire la contrainte de continuité, c'est-à-dire par exemple, autoriser que deux points d'abscisses consécutives aient des ordonnées pouvant différer de deux et non un pixel(s) ; mais cette modification n'assurerait pas une solution générale au problème.

C'est l'autre aspect de la contrainte d'optimisation qui est le plus gênant ($j(i)$ est une **fonction** au sens mathématique du terme). Sa suppression permettrait de conserver comme seule contrainte le 8-voisinage, mais nécessiterait la refonte complète de l'algorithme. Des algorithmes totalement différents seront envisagés et analysés dans le chapitre suivant.

II.4.3.3. Modèle de bord

On peut proposer une autre approche, pour résoudre les problèmes de type 3), que celle consistant à supposer le bord mieux localisé (II.3.3.1). Ce serait de baser l'algorithme sur une **idée moins simple de ce que sont les contours dans une image naturelle numérisée** .

En effet, nous avons supposé qu'une **portion de contour maximise** le long de lui-même le **cumul du gradient** de luminosité ou de texture, et que cette **fonction de coût croît progressivement**

lorsque la chaîne se rapproche du contour exact.

Ceci est vrai dans la plupart des cas pour des **images naturelles** ; cependant, sous certains éclairages, pour certaines formes de l'objet, pour certaines orientations de sa surface et selon même la matière dont il est constitué, des variations importantes de la luminosité ou de la texture, masquent une transition faible entre l'objet et le fond.

La figure II.14. extraite d'un manuel de dessin d'art en présente un cas particulier que la technique du clair-obscur permet de bien restituer.

Toute une étude reste à faire pour savoir si le dilemme peut être résolu localement ou si l'on doit nécessairement recourir à une information de contexte.

Plus précisément, est-il possible de caractériser plus finement les contours pour intégrer dans une description plus large le cas général et le contre exemple de la figure. Des indices existent qui peuvent peut être permettre de choisir un vrai contour peu prononcé au lieu d'un contour erroné mais intense : analyse des zones séparées, étendue latérale du bord d'intensité.

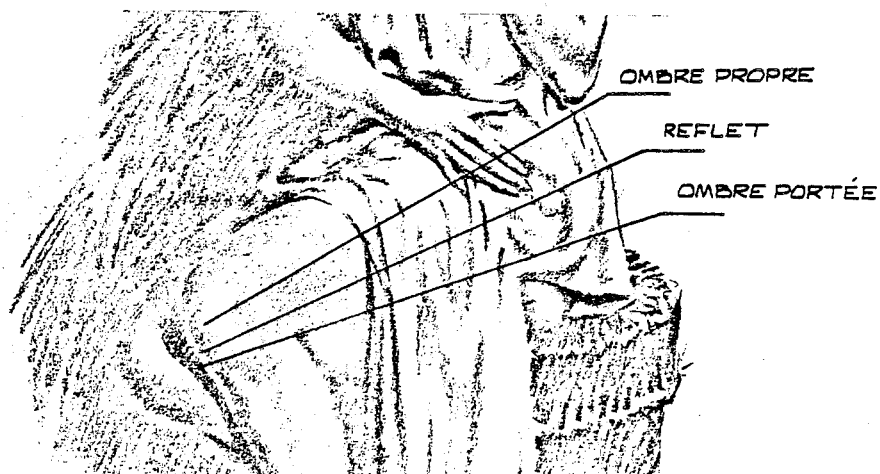


Figure II.14. : la technique du clair-obscur.
"les bases du dessin" - éditions Vinciana.

Il est de toutes façons théoriquement possible d'inférer ce choix d'informations sur la nature de l'objet, son volume, sa disposition dans la scène, l'éclairage, le fond...

ANNEXE

Un problème de mise en oeuvre de l'extraction
de contours : la rotation.

Jusqu'ici, les algorithmes d'extraction de contours dans une fenêtre rectangulaire, ont été décrits en faisant implicitement référence à un repère lié au segment guide. Cela suppose que pour chaque domaine, on puisse effectuer une rotation de la partie d'image précisée à l'aide de ce segment-guide.

La rotation d'une image numérisée pose des problèmes qui ont été largement étudiés et dont l'essentiel tient au fait que les points d'une grille d'échantillonnage ayant subie une certaine rotation, ne tombent en général, en concordance exacte avec aucun

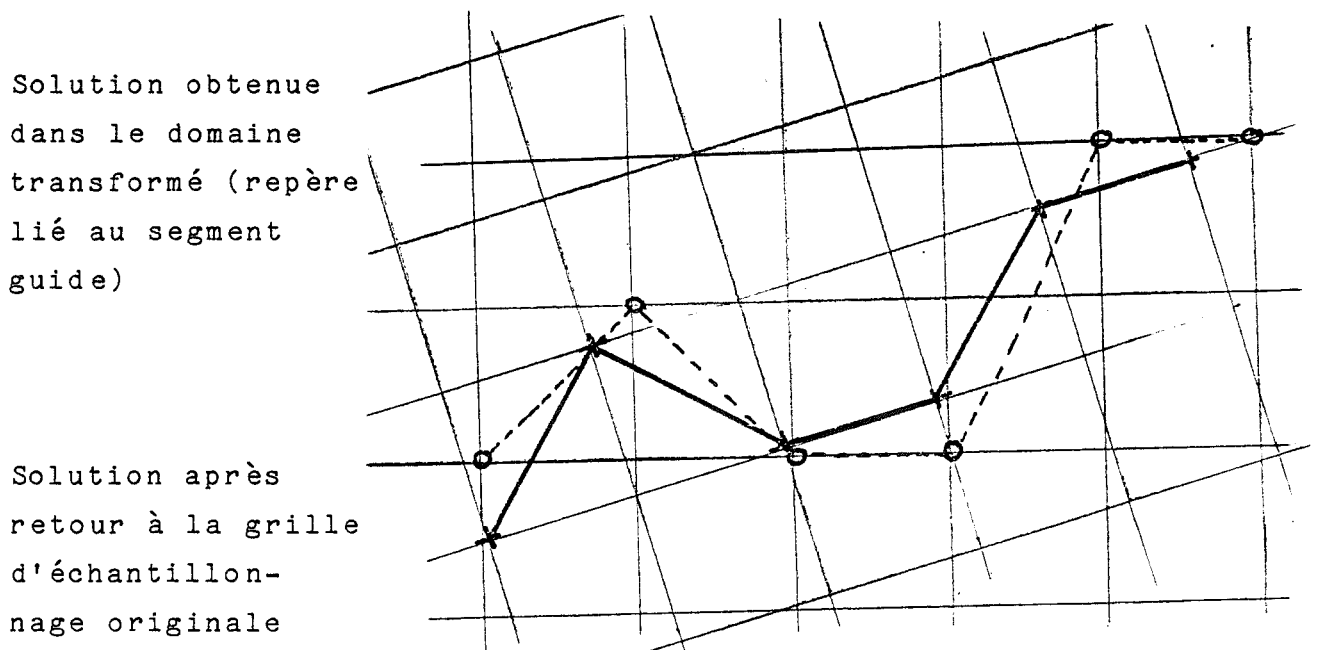


Figure II.11

point de la grille originale. La solution la plus satisfaisante en principe, consiste à attribuer aux points qui ne sont pas sur la grille, une valeur obtenue par interpolation des valeurs des points voisins de la grille (interpolation bilinéaire par exemple). Cette transformation est coûteuse en temps de calcul, et si l'on applique un algorithme d'extraction de contours, on obtiendra pour solution, une liste de points à coordonnées non-entières dans la grille d'échantillonnage originale.

Si l'on juge alors utile de revenir à des coordonnées entières - pour tracer le contour en mémoire d'image par exemple -, on peut approximer chaque point par le point de la grille le plus proche mais d'une part, on perd l'assurance que tout point est "8-voisin" de son prédécesseur, et d'autre part, le contour subit une distorsion non négligeable qui remet en cause le coût payé pour une rotation rigoureuse.

Ceci nous a conduit à substituer à la rotation, une transformation approchée, bijective et plus rapide, que nous appellerons 'pseudo-rotation'. C'est une transformation inspirée des méthodes de rotation rapide proposées par /BRACCINI et MARINO 80/, et qui, comme celles-ci, utilise abondamment l'algorithme incrémental de tracé de segment dû à Bresenham /BRESENHAM 65/ /FOLEY et VAN DAMME 82/.

La bijection entre les domaines transformé et original, est assurée par le report du même tracé de segment, obtenu avec l'algorithme incrémental, perpendiculairement au segment guide. Cette transformation conserve approximativement le 8-voisinage.

La déformation qu'elle induit est une compression perpendiculairement au segment-guide, ce qui ne nuit pas pour l'algorithme d'extraction de contour.

Bibliographie du chapitre II.

- L.S. Davis et A. Mitiche
Edge Detection in Textures.
CGIP 12, 1980.
- A. Guzman
Decomposition of Usual Scenes into 3-Dimensional Bodies.
Fall Joint Conference, 1968.
- D.A. Huffman
Impossible Objects as Nonsense Sentences.
in Machine Intelligence, Meltzer et Michies Edtrs, vol 6,
Edimburg 1971.
- T. Kanade
Recovery of 3-D Shape of an Object from Single View.
in Computer Vision - Brady éditeur, Notrh-Holland 1981.
- G.E. Lowitz
Extraction de Caricatures et Texture de puis un Histogramme
Local.
3° Cong. AFCET Rec. des Formes et Intel. Artif., Nancy, 1981.
- J. Serra
Image Analysis and Mathematical Morphology.
Academic Press, 1982.
- J. Serra
Images et Morphologie Mathématique.
La Recherche n° 144, mai 1983.

Bibliographie de l'annexe.

- C. Braccini et G. Marino
Fast Geometrical Manipulations of Digital Images.
CGIP 13, 1980.
- J.E. Bresenham
Algorithm for Computer Control of Digital Plotter.
IBM Syst. J. 4 n°1, 1965.
- Foley et Van Dam
Fundamentals of Interactive Computer Graphics.
Reading, Mass., Addison Wesley, 1982.

CHAPITRE 3

=====

Chapitre III : RECHERCHE DE CONTOUR COMME CHEMIN
OPTIMAL DANS UN GRAPHE.

III.1. Modélisation.

III.2. Complexité du Problème 2.

III.3. Restrictions sur les Graphes Considérés.

III.3.1. Grilles Simples.

III.3.2. Grilles Pseudo-Symétriques.

III.4. Méthodes Classiques de Recherche de plus Court Chemin.

III.4.1. Algorithme de Dijkstra.

III.4.2. Méthode de Parcours d'un Graphe de Nilsson (A*).

III.4.2.1. Algorithme A*.

III.4.2.2. Analyse de A*.

III.4.2.3. Comportement Heuristique de A*.

III.5. Choix d'une Modélisation.

III.6. Conclusion.

CHAPITRE III :

RECHERCHE DE CONTOURS COMME CHEMIN
OPTIMAL DANS UN GRAPHE.

Nous explorons ici une 2^e classe de méthodes pour résoudre le problème déjà étudié plus haut et pour lequel nous avons décrit dans le chapitre précédent une heuristique : les déformations successives.

Nous ramenant à un domaine formalisé depuis bien longtemps, nous essaierons d'en adopter la rigueur, ce qui explique le changement de ton dans la rédaction de ce chapitre.

Le problème à résoudre est la recherche d'un contour qui relie deux cotés opposés d'une fenêtre rectangulaire prise dans une image.

En chaque point de la fenêtre est évaluée la probabilité pour le point d'appartenir à un contour. On cherche la liste de pixels voisins qui maximise la somme pour tous ses pixels, de la probabilité d'appartenir à un contour.

Nous proposons de modéliser ce problème d'analyse d'image par un problème d'optimisation dans un graphe.

Au cours de notre étude, nous avons eu connaissance de travaux effectués sur la même idée : /BALLARD et SKLANSKY 76/ /MARTELLI 71 et 76/ /LESTER et al. 78/ /ASHKAR et MODESTINO 78/ /RAMER 75/; voir /BALLARD et BROWN 82/ pour un récapitulatif sur le sujet.

Après avoir présenté la modélisation adoptée (III.1), nous étudierons la complexité du problème (III.2), et de relaxations de celui-ci (III.3). Nous analyserons ensuite (III.4), les références bibliographiques se rapportant à cette partie. Pour terminer, nous justifierons le choix de notre modélisation (III.5).

III.1. Modélisation.

A un quadrillage rectangulaire de $n \times m$ pixels, nous associons un graphe noté $G_{n \times m}$ et appelé la "grille symétrique d'ordre $n \times m$ ".

Sommets

A chaque point de coordonnées (i, j) du quadrillage, on associe un sommet du graphe. Nous ajoutons deux sommets "fictifs" : une source et un puits notés s et p .

Arcs

Deux sommets correspondants à des points "8-voisins" du quadrillage, sont reliés par un arc dans chaque sens.

Aucun arc ne relie deux sommets non voisins.

De la source part un arc vers chaque sommet de la 1^o colonne du quadrillage.

Au puits, aboutit un arc depuis chaque sommet de la dernière colonne du quadrillage.

$$G_{n \times m} = (X, U)$$

$$\text{avec } X = \{s, p\} \cup \{(i, j) \mid 1 \leq i \leq n \text{ et } 1 \leq j \leq m\}$$

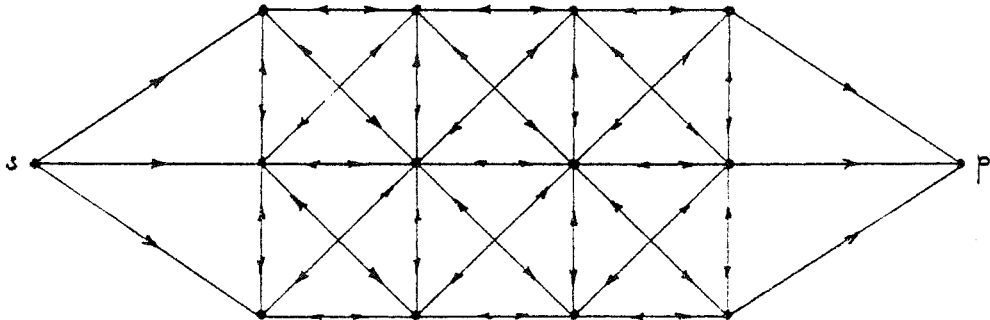
et

$$U = \{(s, (i, 1)) \mid 1 \leq i \leq n\} \cup \{((i, m), p) \mid 1 \leq i \leq n\} \cup \\ \{(i, j), (i, k)\} \mid i-1 \leq j \leq i+1, j-1 \leq k \leq j+1 \text{ et } (i, j) / (k, l)\}$$

(cf note p. suivante)

Exemple

$G_{3 \times 4}$



Valuation

Nous supposerons $G_{n \times m}$ muni d'une valuation strictement positive,

$$v : U \longrightarrow \mathbb{R}^+,$$

sauf aux arcs partant de s ou aboutissant à p , où nous pouvons supposer

$$v((s, (i, 1))) = 0 = v(((i, m), p)) \text{ pour tout } i \text{ de } 1 \text{ à } n.$$

Nous pouvons maintenant exprimer le problème central que nous allons étudier :

Problème 1

Chercher le chemin ch de $G_{n \times m}$ de s à p et de valuation maximale, avec

$$v(ch) = \sum_i v((x_i, x_{i+1})),$$

pour

$$ch = \{x_0=s, x_1, \dots, x_{k+1}=p\}.$$

Ce problème central n'admet évidemment pas de solution de valeur finie, lorsque le graphe considéré possède des circuits à valuations strictement positives.

La première idée, pour éliminer cette difficulté, consiste à interdire les circuits dans les chemins de s à p :

Problème 2

Chercher le chemin de $G_{n \times m}$ élémentaire (i.e. sans circuit), et de valuation maximale.

Remarque : si l'on néglige le cas particulier de la source et du puits, on a pour une grille à N sommets, de l'ordre de $M=4.N$ arcs. Ceci aura d'importantes conséquences lors du calcul de la complexité des algorithmes.

III.2. Complexité du problème 2.

Si l'on ne particularise pas la structure du graphe, le problème 2 est NP-difficile.

Pour la définition des termes "NP-difficile" et "NP-complet", voir par exemple /GAREY et JOHNSON 79/.

Grossièrement cela signifie que le problème est équivalent à des problèmes bien connus et pour lesquels il y a très peu de chances (*) qu'il existe un algorithme polynomial; NP-complet s'applique à un problème de décision et NP-difficile à un problème d'optimisation.

Propriété 1 :

Le problème 2, pour un graphe quelconque, est NP-difficile, car le problème de décision qui lui est associé (problème 2') est équivalent au problème classique ci-dessous ("Elém.") lui-même NP-complet :

Problème 2'.

données : $G=(X,U)$, $v: U \rightarrow R^+$, k dans R^+ , 2 sommets s et p ;
question : Existe-t-il un chemin élémentaire de s à p dans G
et de valuation **supérieure à k** ?

"Elém."

données : $G=(X,U)$, $v: U \rightarrow R^-$, k dans R^- , 2 sommets s et p ;
question : Existe-t-il un chemin élémentaire de s à p dans G
et de valuation **inférieure à k** ?

preuve de la propriété 1 :

Si l'on introduit la valuation $v'=-v$ dans le problème 2', la question devient "existe-t-il un chemin élémentaire de s à p de valuation inférieure à $-k$ " où la valuation est négative et $-k$ aussi. Ce qui est bien le problème "Elém.". La réciproque s'obtient de la même manière. #

On montre que le problème "Elém." est lui-même NP-complet en

(*) voir note p. 91.

le ramenant par une transformation polynomiale au problème suivant de recherche d'un circuit hamiltonien (i.e. passant 1 fois et une seule par chaque sommet).

"Ham."

donnée : $K_n = (X, V)$ complet, $v: V \rightarrow R$, h dans R

question : existe-t-il un circuit hamiltonien de valeur inférieure à h ?

Ce problème aussi appelé "problème du voyageur de commerce, est NP-complet. La réduction est obtenue à partir du problème de la recherche d'un circuit hamiltonien dans un graphe quelconque (cf. /GAR. et JOHN./ p 211).

Proposition 1 :

"Elem." est "au moins aussi difficile" que "Ham." ($\text{Ham} \ll \text{Elem}$)

Preuve:

Il s'agit d'établir une réduction polynomiale, i.e. de montrer que s'il existe un algorithme polynomial pour résoudre le problème "Elem.", alors il en existe également un pour résoudre le problème "Ham."

Considérons une donnée du problème "Ham."

$$K_n = (X, U), v, h;$$

on la transforme en un graphe

$$K' = (X', U'), v', h',$$

en choisissant x_0 de X , que l'on éclate en x'_0 et x''_0 , les arcs entrants en x_0 étant affectés à x'_0 , les arcs sortants à x''_0 .

$$v'(x, y) = v(x, y) - K \quad \text{pour } x_0 \text{ différent de } x \text{ et de } y$$

$$v'(x''_0, y) = v(x_0, y) - K$$

$$v'(y, x'_0) = v(y, x_0) - K$$

où K est un réel choisi très grand.

Appliquons un algorithme polynomial pour résoudre "Elem.", sur la donnée :

$$K', v', x''_0, x'_0, h' = h - nK$$

Si il existe un chemin élémentaire $ch = x''_0, \dots, x'_0$ de K' , tel que $v'(ch) < h'$, à ch on fait correspondre un cycle élémentaire cy de K_n

en contractant x'_0 et x''_0 .

Nous avons alors :

$$v'(ch) = v(cy) - \text{long}(ch).K < h-n.K$$

où $\text{long}(ch)$ est le nombre d'arcs du chemin ch .

$$v'(ch) < h-n.K \text{ et } K \text{ choisi très grand}$$

d'où $\text{long}(ch) = n$

et donc $v(cy) < h$

or cy est de longueur n et élémentaire, c'est donc un circuit hamiltonien. #

Remarque 1 : /GAR. et JOHN./ (p 213) affirment que le problème "Elem." reste NP-complet pour une valuation constante des arcs.

Remarque 2 :

La proposition 1 précédente permet d'obtenir le corollaire ci-dessous sur le problème de Steiner.

Problème de Steiner dans les graphes orientés :

Données : $G=(X,U)$, $v:U \rightarrow R^+$, S inclus dans X , k dans R^+

Question : existe-t-il un sous-arbre T de G , $T=(Y,W)$, avec S inclus dans Y , et tel que

$$\sum_{u \text{ dans } W} v(u) < k \quad ?$$

Corollaire : Le problème de Steiner est au moins aussi difficile que "Elem."

Preuve :

Pour résoudre "Elem." pour les données G , v et k , il suffit d'appliquer un algorithme résolvant le problème de Steiner sur le même graphe, avec une valuation $v'=-v$, pour la constante $-k$ (donc positive) et en prenant $S=\{s,p\}$. #

On démontre ainsi très facilement que le problème de Steiner même pour le cas $S=\{a,b\}$, est NP-complet. On a donc déterminé une frontière "polynomial/non-polynomial", puisque pour $S=\{a\}$, le problème est trivialement polynomial. Pour d'autres frontières cf. /GAREY et JOHNSON/.

Complexité du problème 2 sur une grille.

Question

Que peut-on dire du problème "Elem." dans le cas d'une grille? Le problème reste-t-il NP-complet?

"G-Elem."

données : $G_{n \times m} = (X, U)$, $v: U \rightarrow \mathbb{R}^-$, k réel négatif;

question : existe-t-il un chemin élémentaire de s à p et de valuation inférieure à h ?

Nous formulons la conjecture suivante :

Conjecture :

"G-Elem." est NP-complet.

Si la réponse à cette conjecture est affirmative, alors il y a fort peu de chances (*) pour qu'il existe un algorithme polynomial pour résoudre notre problème 2.

C'est pourquoi nous allons maintenant examiner différents cas particuliers sur lesquels il est possible de construire des algorithmes polynomiaux.

(*) Cet euphémisme s'explique par le fait qu'il n'a pas été démontré à ce jour, que les problèmes NP-complets forment un ensemble différent de l'ensemble des problèmes polynomiaux!

III.3 Relaxations du problème 2 : Restrictions sur les graphes considérés.

III.3.1. Grilles simples $n \times m$.

La 1° restriction que nous envisageons est aussi la plus forte. Elle consiste à ne conserver en chaque noeud que 3 arcs sortants, ceux qui font se rapprocher du puits.

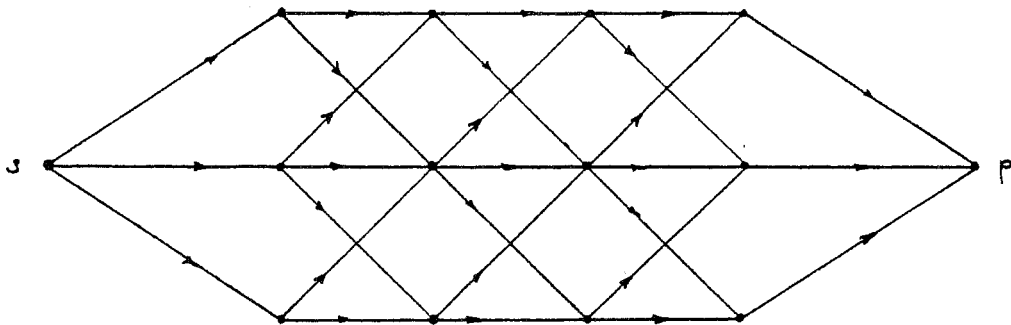
$$U = \{(s, (i, 1)) \mid 1 \leq i \leq n\} \cup \{(i, m), p\} \mid 1 \leq i \leq n\} \cup \{(i, j), (i, j+1)\} \mid i-1 \leq j \leq i+1\}$$

Ceci a pour effet,

- d'empêcher la formation de cycles,
- de faire que tous les chemins de s à p ont même nombre d'arcs.

Exemple :

GS_{3x4}



Propriété :

les problèmes 1 et 2 sur une grille simple se résolvent facilement.

Preuve :

1° Tous les chemins de s à p sont élémentaires donc les deux problèmes sont équivalents.

2° Nous montrons que le problème se ramène à un problème de minimisation de chemin dans un graphe valué positivement, et admet donc un algorithme polynomial.

Considérons la grille simple $G=(X,U)$, munie de la valuation v et soit

$$VMAX = \max_{U} (v(i,j)),$$

on construit sur la même grille simple, la valuation v'

$$v'(i,j) = VMAX - v(i,j) \text{ pour tout } (i,j) \text{ de } U$$

le plus court chemin de (G,U,v') est le plus long chemin de (G,U,v)

En effet, soient

$$m = (s=m_0, m_1, \dots, m_l=p), \text{ le plus court chemin de } (G,U,v')$$

et

$$x = (s=x_0, x_1, \dots, x_k=p), \text{ un chemin quelconque de } s \text{ à } p$$

on a

$$\sum_m v'(m_i, m_{i+1}) \leq \sum_x v'(x_i, x_{i+1})$$

et donc

$$\sum_m (VMAX - v(m_i, m_{i+1})) \leq \sum_x (VMAX - v(x_i, x_{i+1}))$$

ou encore

$$l.VMAX - \sum_m v(m_i, m_{i+1}) \leq k.VMAX - \sum_x v(x_i, x_{i+1})$$

Or sur la grille simple, tous les chemins de s à p ont même longueur $n+1$ ($=l=k$), d'où pour tout chemin x de la grille

$$\sum_m v(m_i, m_{i+1}) \geq \sum_x v(x_i, x_{i+1})$$

m est donc le plus court chemin de (G,U,v) .

#

Ainsi nous pouvons dans ce cas utiliser n'importe quel algorithme de plus court chemin (cf. /GONDRAN et MINOUX 79/), et en particulier celui de /DIJKSTRA 59/ car les valuations sont positives (cf. l'étude faite en annexe 2 et en particulier l'algorithme de Dijkstra).

Cependant nous pouvons aussi profiter du fait qu'une grille simple n'a pas de circuit, en utilisant un algorithme bien connu

de recherche d'un chemin optimal dans un graphe muni d'une "numérotation compatible" de ses sommets /CHEIN 81/ /GONDRAN et MINOUX p 41 et 42/. Cet algorithme n'impose aucune restriction sur le signe des valuations et donc les problèmes min et max sont équivalents.

Numérotation compatible :

num : $X \rightarrow N$ | si (i,j) dans U alors $\text{num}(i) < \text{num}(j)$

Cette numérotation est implicite dans le problème considéré; et de toutes façons, pour tout graphe sans circuit, un numérotation compatible peut-être obtenue linéairement en fonction du nombre d'arcs et de sommets (fonction rang d'un sommet).

(* algorithme de plus court chemin dans un graphe valué positivement et muni d'une numérotation compatible.

On suppose que l'on peut accéder aux sommets prédécesseurs d'un sommet quelconque différent de s (N.B. dans les cas des grilles, ça ne pose pas de problème). *)

début

```
N:=num(s); fermer(N); P:=num(p); g(N):=0;
pour tous les successeurs I de N, ouvrir(I);
ERREUR:=faux; FINI:=faux;
```

tant que pas FINI et pas ERREUR, faire

```
TROUVE:=faux;
```

```
tant que pas TROUVE et pas ERREUR, faire
```

```
    N:=N+1;
```

```
    si  $N \succ P$  alors ERREUR:=vrai
```

```
    sinon si ouvert(N), alors TROUVE:=vrai finsi
```

```
    finsi
```

```
finfaire
```

```
si TROUVE alors
```

```
    g(N) := max (g(I)+v(I,N));
```

```
            I prédec fermés de N
```

```
    fermer(N); ouvrir tous les successeurs de N;
```

```
    finsi
```

```
finfaire
```

```
...
```

L'analyse de la complexité de cet algorithme donne comme résultat $O(M+N)$ où M et N sont le nombre d'arcs et le nombre de sommets du graphe :

L'étape (a) ne reconsidère jamais un noeud déjà envisagé puisque la valeur de n progresse toujours. Globalement (a) donne une complexité en $O(\text{num}(p) - \text{num}(s))$ donc en $O(N)$.

L'étape (b) demande de considérer tous les arcs aboutissant à n . Mais puisqu'on ne considère jamais 2 fois un noeud, on ne considèrera non plus aucun arc 2 fois, soit globalement une complexité en $O(M)$.

D'où le résultat $O(M+N)$; et pour une grille $n \times m$, $O(m.n)$, puisque les nombres d'arcs et de sommets sont proportionnels au produit de n par m .

#

Remarque :

Il est possible de considérer une restriction légèrement moins contraignante que les grilles simples, en autorisant les arcs $((i,j), (i+1,j))$ ou (exclusif) $((i,j), (i-1,j))$.

La longueur des chemins de s à p n'est plus constante, l'équivalence max/min n'est plus assurée et l'algorithme de Dijkstra ne peut plus être utilisé. Mais puisque qu'aucun cycle n'apparaît, la méthode utilisant une numérotation compatible du graphe, reste applicable.

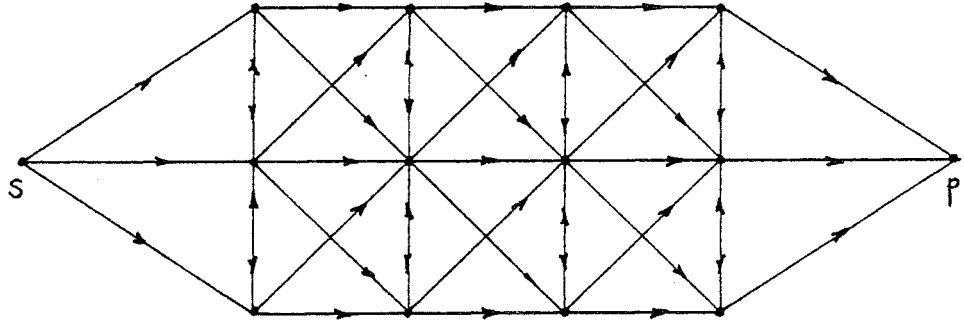
III.3.2. Grilles pseudo-symétriques.

Nous envisageons maintenant une restriction moins forte que celle des grilles simples; il s'agit de réintroduire les arcs entre sommets voisins sur une même verticale.

$$U = \left\{ (s, (i, 1)) \mid 1 \leq i \leq n \right\} \cup \left\{ ((i, m), p) \mid 1 \leq i \leq n \right\} \cup \left\{ ((i, j), (l, k)) \mid (i, j) / (l, k), i-1 \leq l \leq i+1 \text{ et } j \leq k \leq j+1 \right\}$$

Exemple :

GPS_{3x4}



La longueur d'un chemin de s à p n'est plus constante et tous les chemins ne sont plus élémentaires; cependant les boucles sont réduites à des aller-retour sur les colonnes, que l'on peut aisément tester.

Dans le cas de ces grilles nous proposons une technique apparentée à la programmation dynamique, en ce sens que l'on procède par colonnes et sans retour en arrière ("backtracking").

(* recherche chemin élémentaire maximum sur une grille pseudo-symétrique - méthode de type prog. dynamique *)

variables

IPREC : tableau (1..n,1..m) d'entiers; (*pt préc / col préc*)
INTERM: tableau (1..n,1..m) d'entiers; (*pt préc / col courante*)
MAX : tableau (1..n) de réels; (*coûts max sur col précédente*)
NMAX : tableau (1..n) de réels; (*coûts max sur col courante*)

ISOL : tableau (1..2*m) d'entiers;
JSOL : tableau (1..2*m) d'entiers;

I, J, I1, I2, II, II2 : entiers;
C, M1, M2 : réels;

.../...

```
début
pour I:=1 à n, faire (* initialisation 1° colonne *)
    MAX(I):=0; IPREC(I,1):=s; INTERM(I,1):=I; finpour

(* calcul des chaînages arrières *)
pour J:=2 à m, faire (* calcul colonne J à partir colonne J-1 *)
    pour I:=1 à n, faire
        (* calcul du plus long chemin élem de s à (I,J) *)
        M1:=0;
        pour I1:=1 à n, faire
            M2:=0;
            pour I2:=I1-1 à I1+1, faire
                C:= V((I1,J-1),(I2,J)) + V((I2,J),(I,J));
                si C > M2 alors M2:=C; II2:=I2 finsi;
            finpour;
            C:= M2+MAX(I1);
            si C > M1 alors
                M1:=C; IPREC(I,J):=I1; INTERM(I,J):=II2; finsi;
        finpour;
        NMAX(I):=M1;
    finpour;
    pour I:=1 à n, faire MAX(I):=NMAX(I) finpour;
finpour;
(* extraction du chemin élémentaire max *)
C:=0;
pour I:=1 à n, faire si C < MAX(I) alors II:=I finsi; finpour;
I:=0;
pour J:=m à 2 par pas de -1, faire
    I:=I+1; ISOL(I):=II; JSOL(I):=J;
    si INTERM(II,J) # II alors
        I:=I+1; ISOL(I):=INTERM(II,J); JSOL(I):=J; finsi;
    II:=IPREC(II);
finpour;
fin.
```

Analyse : La complexité de l'algorithme ainsi écrit se déduit aisément de l'analyse des boucles imbriquées du calcul des chaînages arrières : $O(m.n^2.3)$.

III.4. Méthodes classiques de recherche de plus court chemin.

III.4.1. ALGORITHME DE DIJKSTRA.

(* plus court chemin dans un graphe à valuations positives *)

- (a) A tout noeud du graphe est associé un des trois états : 'non encore atteint', 'ouvert' et 'fermé' (cf rem. 1 p. 101).
- (b) En tout noeud ouvert ou fermé n est défini $g : g(n)$ est le **coût du plus court chemin trouvé jusqu'à présent**, de la source à ce noeud. On montre que pour N fermé, $g(N)$ est le coût optimal de s à N (cf. plus loin).
- (c) En ces mêmes noeuds est défini un pointeur $prec(n)$ vers le noeud précédent dans ce plus court chemin jusqu'à présent.

Début

```
ouvrir(s); g(s):= 0; préc(s):= nil; FINI:= faux; ERREUR:= faux;
Tant que pas FINI et pas ERREUR, faire
    si aucun noeud ouvert alors ERREUR := vrai
    sinon
        N := x tel que x ouvert et pour tout y ouvert,
             $g(x) \leq g(y)$ ; en cas d'égalité, priorité aux puits
        fermer(N);
        Si N est un puits alors FINI := vrai
        sinon
            Pour tous les successeurs  $N_i$  de N, faire
                gchap := g(N) + c(N,  $N_i$ );
                Si  $N_i$  non encore atteint alors
                    ouvrir( $N_i$ ); g( $N_i$ ):= gchap; préc( $N_i$ ):= N;
                sinon
                    Si gchap < g( $N_i$ ) alors
                        (*  $N_i$  nécessairement ouvert *)
                        g( $N_i$ ) := gchap; préc( $N_i$ ) := N;
                    finsi;
            finpour;
        finsi;
    finsi;
fintantque;
(* on peut reconstituer le plus court chemin
à l'aide des pointeurs arrières préc(n) *)
... fin.
```

Analyse de l'algorithme de Dijkstra.

Considérons $G=(X,U)$, $v:U \rightarrow R^+$, avec $n=|X|$ et $m=|U|$;

Une 1^o analyse, nous indique que c'est la recherche du nouveau sommet à fermer qui est la partie la plus coûteuse de l'algorithme. En effet, passer en revue les successeurs de N , ne conduit jamais à reconsidérer un arc déjà pris en compte; globalement, cette dernière opération sera en $m \cdot \log(n)$.

Par contre, si à chaque fois nous testons tous les sommets ouverts pour sélectionner celui à fermer i.e. le plus petit, alors nous obtenons un algorithme de complexité de l'ordre de n^2 .

Cependant, il existe une technique meilleure obtenue en utilisant une queue de priorité ou encore un "tas" (en anglais heap), comme structure de donnée pour l'ensemble des sommets ouverts (la structure de tas est un arbre binaire quasi équilibré). Le premier à avoir remarqué cela est /JOHNSON 72/.

La structure de tas, est une structure de donnée qui permet de faire en $O(\log(k))$ si elle contient k éléments, les opérations:

- insérer un nouvel élément,
- enlever l'élément ayant la valeur numérique la plus faible.

Ainsi, en utilisant cette structure, lorsque le graphe est défini par ses listes d'adjacences, il faut

- pour chaque nouveau sommet fermé, $O(\log(n))$ opérations élémentaires,
- pour chaque voisin du sommet fermé, $O(\log(n))$ opérations;

d'où en fin de compte un algorithme en $O((n+m) \cdot \log(n))$. #

Cette technique est donc meilleure que n^2 pour m petit devant n^2 , ce qui est le cas pour les grilles.

Remarque 1 :

Dans la version classique de l'algorithme de Dijkstra, seuls 2 états, 'fermé' (dans S) et 'non fermé' (dans S), sont définis.

La scission de l'état 'non fermé' en 2 états 'ouvert' et

'non-atteint', permet d'exploiter une particularité des grilles : l'ensemble des sommets non fermés que l'on peut atteindre depuis l'un quelconque des sommets fermés, constitue un "petit" sous ensemble de S, puisqu'il correspond à sa frontière.

Remarque 2 :

En fait pour conserver la complexité $O((n+m).log n)$, sans disposer de la représentation explicite du graphe, la queue de priorité n'est pas la seule structure de donnée nécessaire.

En effet, les opérations à réaliser sur l'ensemble des sommets, sont :

- Extraction du sommet ouvert d'évaluation minimale;
- Insertion d'un nouveau sommet d'évaluation donnée, dans l'ensemble des ouverts;
- Mise à jour de l'évaluation d'un sommet déjà ouvert;
- Test de l'état d'un sommet quelconque;
- Modification de l'état d'un sommet.

Si les 2 premières opérations sont effectivement réalisées en un temps de l'ordre de $log n$, sur une queue de priorité, la 3^e opération ne peut l'être. Il nous est donc apparu nécessaire de pourvoir l'ensemble des sommets ouverts d'une 2^e structure, celle d'arbre dichotomique (arbre AVL, /.../) qui le permet.

Enfin si la structure d'arbre AVL associée à l'ensemble des sommets ouverts permet de dire en $O(log(n))$ si un sommet est ouvert ou non, il ne permet pas de dire si un sommet est 'fermé' ou 'non atteint'.

Pour cela, la solution que nous proposons, -de préférence à un arbre AVL contenant tous les sommets fermés (ou (exclusif) tous les sommets non atteints)-, est d'utiliser une structure de table à 2 adresses puisque l'on en dispose d'une sous la forme de mémoire d'image. Il suffit donc de réserver dans celle-ci 2 bits à cet effet.

III.4.2. METHODE DE PARCOURS D'UN GRAPHE DE NILSSON.

Dans son livre, /NILSSON 71/ présente une procédure qui permet de déterminer dans un graphe valué, un chemin entre 2 sommets. Cette méthode a été développée et utilisée en intelligence artificielle, par exemple en théorie des jeux lorsqu'il y a une arborescence de solutions à envisager. Elle est indifféremment citée dans la littérature comme heuristique ou algorithme; pour plus de clarté, nous parlerons de méthode ou de procédure (*).

Cette procédure semble particulièrement adaptée aux problèmes de plus courts chemins sur de grands graphes, généralement définis de façon implicite plutôt qu'explicite, comme c'est le cas en intelligence artificielle.

La méthode de Nilsson fréquemment appelée "algorithme" A*, est en quelque sorte, une généralisation de l'algorithme de plus court chemin de /MOORE 59/, lui-même constituant une extension de celui de /DIJKSTRA 59/.

En effet, l'algorithme de Moore est strictement identique à A* pour h nulle en tout sommet du graphe.

Il est une extension de celui de Dijkstra au sens où il s'applique pour une valuation quelconque; mais dans le cas d'une valuation positive, les 2 algorithmes exécutent exactement les mêmes instructions (à un test près), sur les mêmes données; pour une valuation non toujours positive, l'algorithme prévoit que certains noeud devront être réouverts.

(*) En effet à notre sens, pour un **algorithme**, on peut prouver qu'à coup sûr, la solution est obtenue (ex. l'algorithme d'Euclide pour le calcul du PGCD); une **heuristique** donne au contraire un résultat dont on ne peut garantir qu'il est la solution.

Algorithme A* (d'après Nilsson).

- (a) A tout noeud du graphe est associé un des trois états : 'non encore atteint', 'ouvert' et 'fermé'.
- (b) En tout noeud atteint n est défini $g(n)$, le coût du plus court chemin trouvé jusqu'à présent, de la source à ce noeud.
- (c) En ces mêmes noeuds est défini un pointeur $prec(n)$ vers le noeud précédent dans ce plus court chemin jusqu'à présent.
- (d) En tout noeud n est définie une estimation a priori $h(n)$ du coût du plus court chemin de ce noeud à un puits.

Début

ouvrir(s); $g(s) := 0$; $prec(s) := nil$; FINI := faux; ERREUR := faux;

Tant que pas fini et pas ERREUR, faire

 si aucun noeud ouvert alors ERREUR := vrai

 sinon

$n := x$ tel que x ouvert et pour tout y ouvert,

$g(x)+h(x) < g(y)+h(y)$;

 fermer(n);

 Si n est un puits alors FINI := vrai

 sinon

 Pour tous les successeurs n_i de n , faire

$gchap := g(n) + c(n, n_i)$;

 Si n_i non encore atteint alors

 ouvrir(n_i); $g(n_i) := gchap$; $prec(n_i) := n$;

 sinon

 Si $gchap < g(n_i)$ alors

 Si n_i fermé alors ouvrir(n_i) finsi;

$g(n_i) := gchap$; $prec(n_i) := n$;

 finsi;

 finsi;

 finpour;

 finsi;

 finsi;

fintantque;

... fin. (* retrouver le + court ch. par chaînages arrières *)

Analyse de A*.

La fonction h va avoir une importance capitale dans le fonctionnement de cet algorithme.

Nilsson a démontré les propriétés suivantes et définies des conditions sur h . La présentation identique des procédures nous permet d'en donner des preuves simples.

Définition 1 :

h est **admissible** si et seulement si
pour tout sommet x et
pour tout chemin ch de x à un puits, on a
 $0 \leq h(x) \leq \text{coût}(ch)$.

Propriété 2 : Lorsque h est admissible, A^* se termine et calcule un chemin optimal.

Preuve : Il suffit de remarquer que cette procédure est de structure identique à celle de /MOORE 59/, seule la fonction h n'apparaît pas.

Soit g' la fonction calculée par la procédure de Moore;
comme $g'(n) = g(n) + h(n) = g^*(n)$ à tout instant, la preuve de l'algorithme de Moore donne celle de la propriété 2.

#

Définition 2 :

h est **consistante** si et seulement si
pour tout chemin ch de m à n , on a $h(m) - h(n) \leq \text{coût}(ch)$

Propriété 3 :

Lorsque h est consistante, A^* se comporte exactement comme l'algorithme de Djikstra, aux notations près.

Lemme 1 : (On note S l'ensemble des sommets fermés à un instant)

Au cours de l'algorithme,

$$g(x) = \min_{ch \text{ inclus dans } S \text{ sauf } x} (\text{coût}(ch))$$

Preuve : obtenue par induction sur $|S|$.

#

Lemme 2 : Lorsque y est fermé, alors $g(y)$ = coût optimal de s à y .

Preuve :

Si y est le sommet que l'on va fermer
on a :

$$g(y)+h(y) = \min_{x \text{ ouverts}} (g(x)+h(x)) \quad (1)$$

Supposons qu'existe un chemin m de s à y plus court que $g(y)$, on a
coût(m) $<$ $g(y)$

or coût(m) = coût($m(s,x)$) + coût(x,y)

avec x , 1° sommet non fermé de m ,

on a alors

coût($m(s,x)$) \leq $g(x)$ (lemme 1)

coût(x,y) \leq $h(x)-h(y)$ (consistance)

d'où coût(m) \leq $g(x) + h(x) - h(y)$

et d'après (1)

$g(x) + h(x) - h(y) < g(y)$

Nous avons donc $g(y) <$ coût(m) $<$ $g(y)$, ce qui est exclu. #

Comportement heuristique de A^* .

La procédure A^* est d'autant meilleure que celles de Dijkstra et de Moore, que h est une bonne évaluation par défaut du coût optimal de chaque sommet à un puits et permet de faire un bon choix. Ceci justifierait une analyse du comportement moyen de cette procédure.

En dehors de la condition (2) précédente, il n'est plus assuré que A* découvre un chemin optimal en un temps proportionnel à N^2 (N nombre de sommets du graphe).

En effet, ceux-ci peuvent être explorés plusieurs fois, au maximum un nombre de fois proportionnel à N^2 ; mais ce maximum peut être atteint dans le cas particulier que l'on construit comme suit.

Dans un 1° temps, nous présentons une situation (exemple 1) où un noeud doit être réouvert, à la 4° itération dans A* :

exemple 1

exemple 2

1° itération: on ferme S pour $g = 0$
 on ouvre A pour $f = 0$ (on note $f=g+h$)
 et B pour $f = 8$

2° itération: on ferme A pour $g = 0$
 on ouvre C pour $f = 7$

3° itération: on ferme C pour $g = 7$
 on ouvre P pour $f = 9$

Sont actuellement ouverts P ($f(P)=9$) et B ($f(B)=8$)

4° itération: on ferme B pour $g = 1$
 on réouvre C pour $f = 6$

On peut sur ce principe, construire une situation (exemple 2) où un noeud est fermé $N-3$ fois (N nombre de noeuds du graphe), en mettant autant de noeuds en dérivation que nécessaire.

On peut maintenant construire un exemple où le nombre de fermetures d'un noeud est proportionnel au carré du nombre de noeuds (à une constante près). Il suffit pour cela de construire un graphe par assemblage en série, de 2 éléments identiques à l'exemple 2.

En augmentant à volonté le nombre P d'éléments en série, on montre qu'un noeud peut être exploré $\frac{N-P-2}{P}$ fois.

P

Le comportement de l'algorithme A* utilisant une information heuristique non consistante, peut donc s'avérer largement plus mauvais que l'algorithme de Dijkstra, i.e. A* n'utilisant aucune information heuristique.

III.5. Choix d'une modélisation.

L'analyse de la complexité du problème de recherche d'un chemin élémentaire sur une grille, montre que le choix d'une modélisation plutôt qu'une autre, est crucial dans le 1° sens suivant :

Pour certaines modélisations, on disposera d'algorithmes que l'on pourra raisonnablement mettre en oeuvre, alors que ce ne sera pas vrai pour d'autres.

La deuxième propriété que doit vérifier une bonne modélisation est de représenter de façon fidèle, les données du problème, le problème lui-même et donc, sa ou ses solution(s).

La modélisation du problème d'extraction de contours en terme d'optimisation dans graphe, n'est pas aussi triviale qu'il y paraît au premier abord.

L'énoncé du problème lui-même, peut donner lieu à des variantes, par exemple :

- 1) étant donnés 2 points dans une image, trouver le contour qui les relie, ou dire qu'il n'en existe pas;
- 2) étant donnée une fenêtre rectangulaire prise dans une image, trouver le contour intérieur à la fenêtre, qui relie 2 cotés opposés;
- 3) étant donné un point dans une image, trouver le meilleur contour fermé qui passe par ce point;
- 4) étant donnée une image, trouver tous les contours fermés qui y apparaissent;
- 5) étant donnée une image, trouver tous les contours fermés ou non, y figurant;

etc...

Une fois le problème clairement exprimé, il faut établir :

- le problème correspondant sur le graphe : recherche d'un chemin, d'un chemin élémentaire, d'un cycle, d'un cycle élémentaire, d'un sous-graphe;
 - le "sens" de l'optimisation (minimisation ou maximisation);
 - l'expression de la fonction de coût sur un arc;
 - l'expression de la fonction de coût sur un chemin, un cycle etc...;
 - la structure de graphe considérée;
- tout en tâchant d'aboutir à un problème de graphe résoluble par un algorithme polynomial.

/MARTELLI 72/ s'attache exactement au même problème que celui qui est au centre de tout ce chapitre : trouver le "meilleur" contour , dans une fenêtre rectangulaire, qui relie 2 cotés opposés de celle-ci. Cependant les modélisations adoptées diffèrent.

Dès le départ, Martelli exprime le problème en terme de minimisation, et il ne motive ce choix que **très** succinctement. Pourtant, il ne nous semble pas que celui-ci s'impose nécessairement ou naturellement. Au contraire, la modélisation la plus naturelle nous paraît être une **maximisation**. D'ailleurs la fonction de coût proposée par Martelli le montre bien : celle-ci (une constante moins un terme de type gradient), est visiblement construite à partir d'une fonction de coût "naturelle" qu'il faudrait maximiser; mais comme c'est un problème complexe, Martelli traite un problème de minimisation à **peu près équivalent**. Pour cela il utilise une démarche identique à celle du § II.5.3, à cela près que la longueur (en nombre d'arcs) des chemins n'étant pas fixée, l'équivalence

"maximisation" = "minimisation"

n'est pas vérifiée.

Exemple : (cf. démonstration pour les grilles simples, §III.3.1)

$$G=(X,U), v, \min_{\text{ch de } s \text{ à } p} v(\text{ch})$$

$$G=(X,U), v'=-v, \max_{\text{ch de } s \text{ à } p} (5-v(\text{ch}))$$

Les fonctions de coût sur les arcs et sur les chemins associées au problème, doivent assurément faire intervenir un terme évaluant le gradient d'intensité de gris de l'image.

La modélisation la plus immédiate, consiste probablement à évaluer -par application d'un détecteur de bords- le module du gradient en chaque pixel; mais il s'agit là d'une **valuation aux sommets** et l'on doit la ramener à une **valuation sur les arcs**, par exemple en imposant comme valeur d'arc, la valeur du sommet atteint. Enfin il est naturel d'associer comme coût à un chemin, la somme des coûts sur les arcs qui le composent.

Cependant cette valuation n'est admissible que dans le seul cas des "grilles simples". En effet, tout chemin élémentaire passant par tous les sommets d'une grille quelconque, est optimal pour cette fonction de coût. D'autre part, une telle valuation néglige à l'évidence, l'information de direction contenue dans le gradient.

Le prise en compte de la direction et du module de bord, dans une valuation réellement associée aux arcs correspondant à la relation de 8-voisinage, se fera en favorisant l'arc dont la direction est la plus proche de la perpendiculaire au gradient.

Nous avons vu en II.3.3.2, l'intérêt de faire intervenir la longueur du contour dans son évaluation.

Enfin /MARTELLI 76/ propose d'introduire un terme mesurant la forme locale du contour.

CONCLUSION.

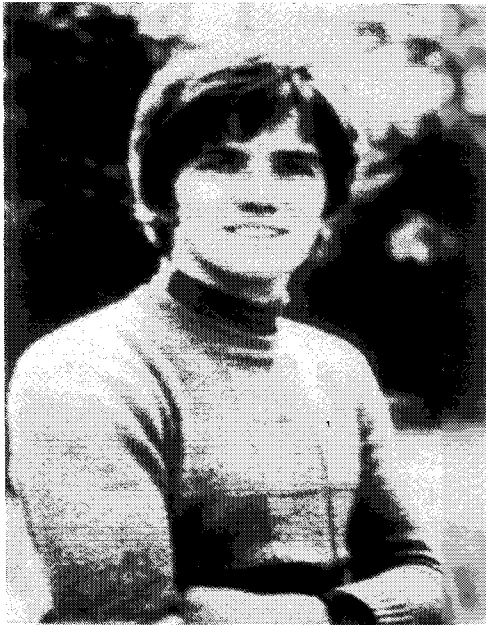
Nous récapitulons sur le tableau ci-dessous, la complexité de problèmes de recherche de chemin optimal dans une grille, que nous avons associés à un problème d'extraction de contour dans une image.

restriction de la grille $n \times m$	méthode adaptée	complexité
grille simple	+ court ch dans un graphe numéroté	$O(m.n)$
	Dijkstra	$O(n^2.m^2)$
	Dijkstra + "heap"	$O((n.m).log(n.m))$
grille pseudo- symétrique	méthode type prog dynamique	$O(m.3n^2)$
grille symétrique quelconque	méthode heuristique	NP difficile

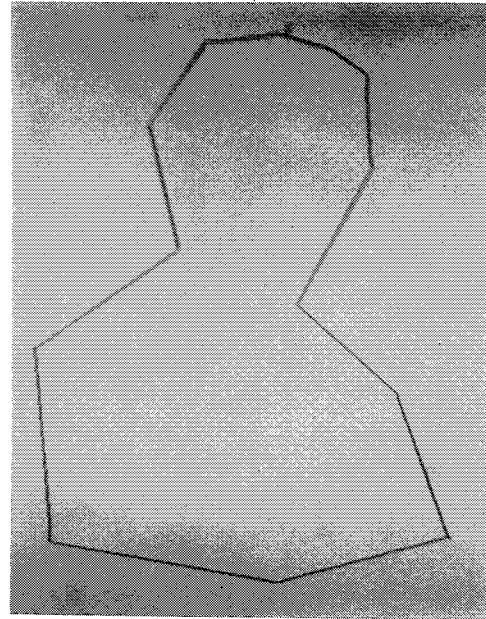
Malheureusement, les restrictions proposées pour disposer d'algorithmes efficaces apparaissent assez contraignantes lorsqu'on les ramène au problème d'extraction de contours : en effet, elles conduisent à n'admettre comme solution que les listes de pixels monotones en x , ce qui limite les circonvolutions d'un contour admissible. D'autre part, il semble difficile d'imaginer une restriction intermédiaire entre grilles symétriques et pseudo-symétriques.

Ceci justifie a posteriori, et jusqu'à l'obtention de résultats théoriques plus puissants sur les grilles symétriques, l'utilisation de procédures heuristiques, d'autant plus que le résultat présenté sur la planche de la page suivante, montre que

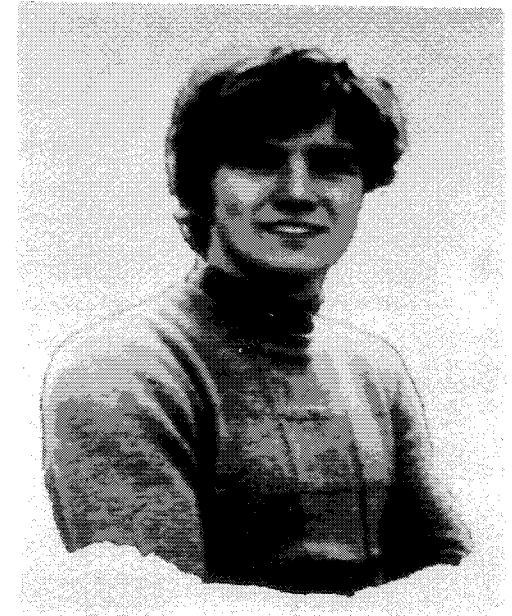
Extraction de contour comme chemin optimal - Algorithme de Dijkstra.



-a-

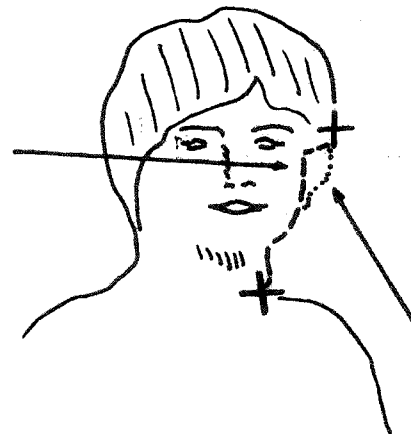


-b-



-c-

Algorithme de Dijkstra



déformations successives

Comparaison avec le résultat obtenu par Déformations Successives (d).



-d-

la qualité subjective (par opposition à l'optimalité numérique) de l'algorithme de Dijkstra est comparable à celle obtenue par déformations successives. Les temps de calcul, par contre ne sont pas du même ordre : Déformations successives = environ 3 mn CPU; algorithme de Dijkstra (sans structure de Heap) = environ 40 mn!).

Bibliographie du chapitre III.

- G.P. Ashkar et J.W. Modestino
The Contour Extraction Problem with Biomedical Applications.
CGIP 7, 1978.
- D.H. Ballard et C.M. Brown
Computer Vision
Prentice-Hall, Englewood Cliffs, 1982.
- D.H. Ballard et J. Sklansky
A ladder-structured Decision Tree for Recognition Tumors in
Chest Radiographs.
IEEE Trans. on Computers, May 1976.
- M. Chein
Optimisation dans les Graphes, 1^opartie : Chemins Optimaux,
Flot Maximum.
Institut de Programmation, Université de Paris VI, 1981.
- E.W. Dijkstra
A Note on Two Problems in Connexion with Graphs.
Numerische Mathematik 1, 1959.
- M.R. Garey et D.S. Johnson
Computers and Intractability : a Guide to the Théory of
NP-completeness.
Freeman, San Francisco, 1979.
- M. Gondran et M. Minoux
Graphes et Algorithmes.
Eyrolles, Paris, 1979.
- E. Johnson
On Shortest Paths and Sorting.
Proc. Ann. Conf. ACM, Boston, 1972.
- J.M. Lester, H.A. Williams, B.A. Weintraub et J.F. Brenner
Two Graph Searching Techniques for Boundary Finding in White
Blood Cell Images.
Computers in Biology and Medecine, Vol 8, Pergamon, 1978.
- A. Martelli
Edge Detection using Heuristic Search Methods.
CGIP 1, 1972.

A. Martelli

An Application of Heuristic Search Methods to Edge and Contour Detection.

Comm. ACM, vol 19 n°2, February 1976.

E.F. Moore

The Shortest Path through a Maze.

Proc. Int. Symp. on Theory of switching, 1957.

N.J. Nilsson

Problem-solving Methods in Artificial Intelligence.

Mc-Graw-Hill, 1971.

U. Ramer

Extraction of Line Structures from Photographs of Curved Objects.

CGIP 4, 1975.

CHAPITRE 4

=====

Chapitre IV : MODELISATION DES OBJETS A DETOURER.

IV.1. Présentation et Rôles des Modèles.

IV.1.1. Généralités sur la notion de modèle.

IV.1.2. Rôles de la modélisation dans le détournage.

IV.2. Représentation des Informations Globales et des Informations a Priori.

IV.2.1. Représentation Canonique des Silhouettes Possibles.

IV.2.2. Sources de Variation de la Forme de l'Image d'un
Objet.

IV.3. Le Modèle en Génération.

IV.3.1. Génération de la Silhouette Guide pour la
Segmentation.

IV.3.2. Mise à l'Echelle.

IV.4. Inférence : à la Charnière de l'Analyse et de la Génération.

IV.5. Analyse selon un Modèle.

IV.5.1. Intérêt d'une Analyse Automatique même Partielle.

IV.5.2. Extraction de Primitives et Reconstruction d'Eléments
de niveaux supérieurs.

IV.6. Le Rôle du Modèle dans l'Interaction.

CHAPITRE IV :

MODELISATION

Il a été indiqué dans le chapitre d'introduction (cf. § I.5) qu'une certaine **interprétation de l'image** est nécessaire dès lors que l'on désire s'assurer ou tirer parti, de la **vraisemblance globale** que doit présenter le contour de l'objet à détourer.

D'interprétation à compréhension, il n'y a pas grande différence, si ce n'est peut-être une légère nuance anthropomorphique.

La compréhension d'images est toutefois un sujet de recherche assez nouveau et dont l'ampleur et la difficulté débordent de très loin les limites de ce travail. M. Brady du M.I.T. estime que l'intérêt pour le thème de la compréhension d'images "**devrait s'affirmer dans les prochaines décennies**" /BRADY 82/ ! C'est dire l'étendue et le nombre des problèmes qui restent encore à résoudre dans l'analyse d'images quand la description à laquelle on désirerait aboutir, est d'ordre sémantique.

Nous nous limiterons ici, à essayer de démontrer que l'introduction de techniques d'interprétation d'images, dans un système informatique de mise en page, peut s'avérer utile à l'aide au détourage.

La première de ces techniques est la modélisation. Elle est l'objet de ce chapitre. Les deux courts chapitres qui suivront présenteront sur le cas particulier de deux types d'objets, l'illustration des idées et des principes généraux exposés ici.

La deuxième de ces techniques est l'élaboration et la modification du plan d'action du programme en tenant compte des résultats obtenus au fur et à mesure de l'avancement de la tâche. Ce sera l'objet du chapitre de conclusion.

IV.1. Présentation et rôles des modèles.

IV.1.1. Généralités sur la notion de modèle.

Le sens premier du terme "modèle" renvoie à l'idée d'**exemple à suivre** ("les petites filles modèles" - le "modèle" de l'artiste peintre). On doit rapprocher de ceci, la notion introduite dans les 2 chapitres précédents, de "segment-guide" et encore celles de **prototype** et de **patron** (Anglais : "Pattern", dans "Pattern Recognition", ou encore "template" dans "template matching").

Dans un autre sens, l'idée de modèle, dérivée de sens courant, est d'un usage très fréquent dans de nombreuses disciplines scientifiques (économie, physique et chimie par exemple), dès lors que l'on désire **représenter de façon condensée et raisonnablement simple une réalité complexe.**

Dans ce dernier sens, le **modèle d'un objet** (respectivement d'une classe d'objet), **pris sous un certain point de vue, est un autre objet** souvent purement abstrait (mais pas nécessairement), qui aura sous ce même point de vue, des **propriétés analogues** à l'objet modélisé (respectivement à l'un quelconque des objets de la classe), mais dont la manipulation sera beaucoup plus pratique.

- La modélisation peut être mise en oeuvre à plusieurs fins :
- pour guider la **reproduction** d'un objet : le patron utilisé en couture représente le vêtement dont on veut réaliser une copie ;
 - pour **prédire** l'état ou le comportement d'un objet par expérimentation sur le modèle, dans le cas où l'expérimentation sur l'objet lui-même est difficile ou même impossible : simulations économiques et autres, prévisions météorologiques, etc...
 - pour **reconnaître et classer** des objets : l'analyse d'un objet, en regard du **modèle d'une classe d'objets**, permet de décider de l'appartenance de l'objet à la classe (reconnaissance des formes).

IV.1.2. Rôles de la modélisation dans le détournage.

Par chacun des aspects ci-dessus, la modélisation est susceptible de jouer un rôle dans le détournage de photographies.

On introduira dans la suite, les modèles d'interprétation d'images sous la forme plus générale utilisée en intelligence artificielle de modèle du monde :

Interpréter une image, c'est faire coïncider des informations disponibles dans celle-ci avec certaines des connaissances que l'on possède a priori sur tout ce qu'elles peuvent représenter.

La représentation de cet ensemble de connaissances a priori, est classiquement appelée un **modèle du monde** /HANSON RISEMAN 79/.

Les informations représentées dans un modèle du monde, ne sont pas indépendantes et elles n'ont pas toutes la même portée. Au contraire, elles se présentent le plus souvent sous forme d'une structure hiérarchisée où les éléments d'un niveau sont décrits à l'aide d'éléments de niveaux inférieurs. Ceci est illustré sur un exemple très simplifié dans la figure IV.1.

Cette structure hiérarchique a fait l'objet de formalisations théoriques, et elle a pu être décrite comme "généralisation de la notion de langage à contexte libre" /MOHR 75/; toute une part de la terminologie en est plus ou moins directement inspirée.

Pour détourner l'image d'un objet apparaissant sur une photographie, un modèle du monde nous paraît utile dans la mesure où il apporte une aide substantielle dans l'une ou plusieurs des fonctions suivantes :

- La fabrication de la silhouette grossière de l'image d'un objet pour n'importe quelle apparition de celui-ci ; on parlera de **génération** ;
- L'analyse des informations les plus **saillantes** de l'image pour tenter de les faire coïncider avec des éléments du modèle (reconnaissance, interprétation, compréhension) ;
- L'interaction avec l'opérateur sous une forme concise et agréable pour celui-ci. C'est ce que nous appellerons "l'interaction linguistique".

Le modèle du monde utile pour détourer des photographies, sera naturellement formé d'importants sous-ensembles correspondant chacun au modèle d'une classe d'objets à segmenter ; à ceux-ci pourrait s'ajouter un modèle complémentaire, celui des fonds sur lesquels apparaissent les objets. Enfin, un modèle du monde peut présenter une structure hiérarchique entre ses sous-modèles : par exemple, le modèle d'un visage humain -utile pour détourer des portraits-, peut aussi apparaître comme un sous-ensemble du modèle du corps humain.

IV.2. Représentation des informations globales et des informations a priori.

La nécessité de disposer, lors de la segmentation finale, d'informations globales (introduction du contexte dans la segmentation), a été précédemment mise en évidence (cf. I.3.2 et II.1).

Le rôle fondamental du modèle d'un objet lors du détourage de son image, sera de représenter tout ce qui est nécessaire à la constitution de cette information dont l'aspect le plus important -on s'en souvient- est une silhouette polygonale grossière.

IV.2.1. Représentation canonique des silhouettes possibles.

Une ambiguïté reste en ce qui concerne ce qu'il y a lieu de modéliser : est-ce la forme de l'image des objets réels (modèle 2D), ou la géométrie dans l'espace des objets eux-mêmes (modèle 3D) ? Les éléments de réponse sont multiples et contradictoires : cf. II.4.3.3., V.1.3. et V.3.3.2.

La principale difficulté réside évidemment dans le fait, qu'à un objet donné, on ne peut faire correspondre un contour précisé-

Figure IV.1.

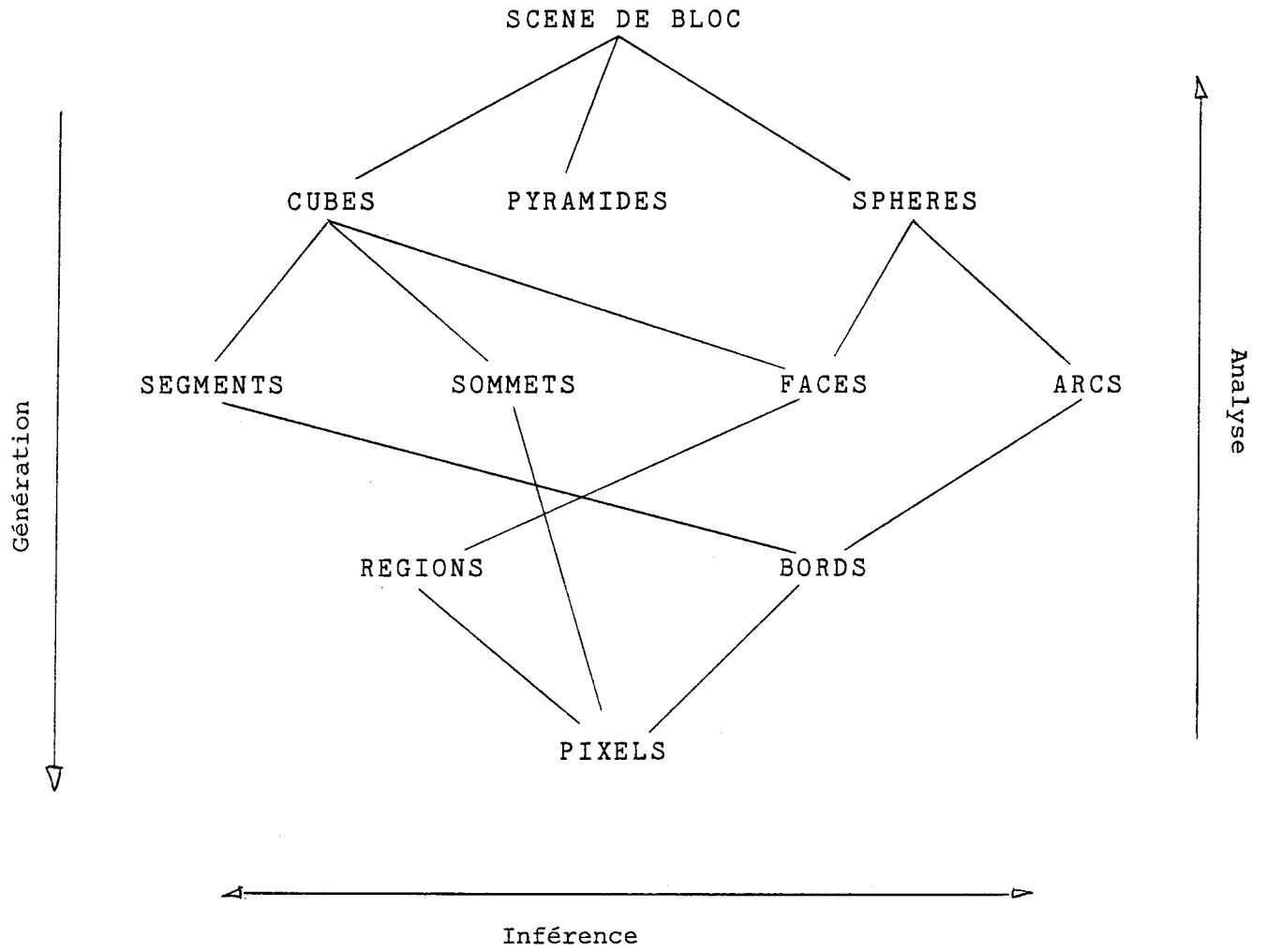
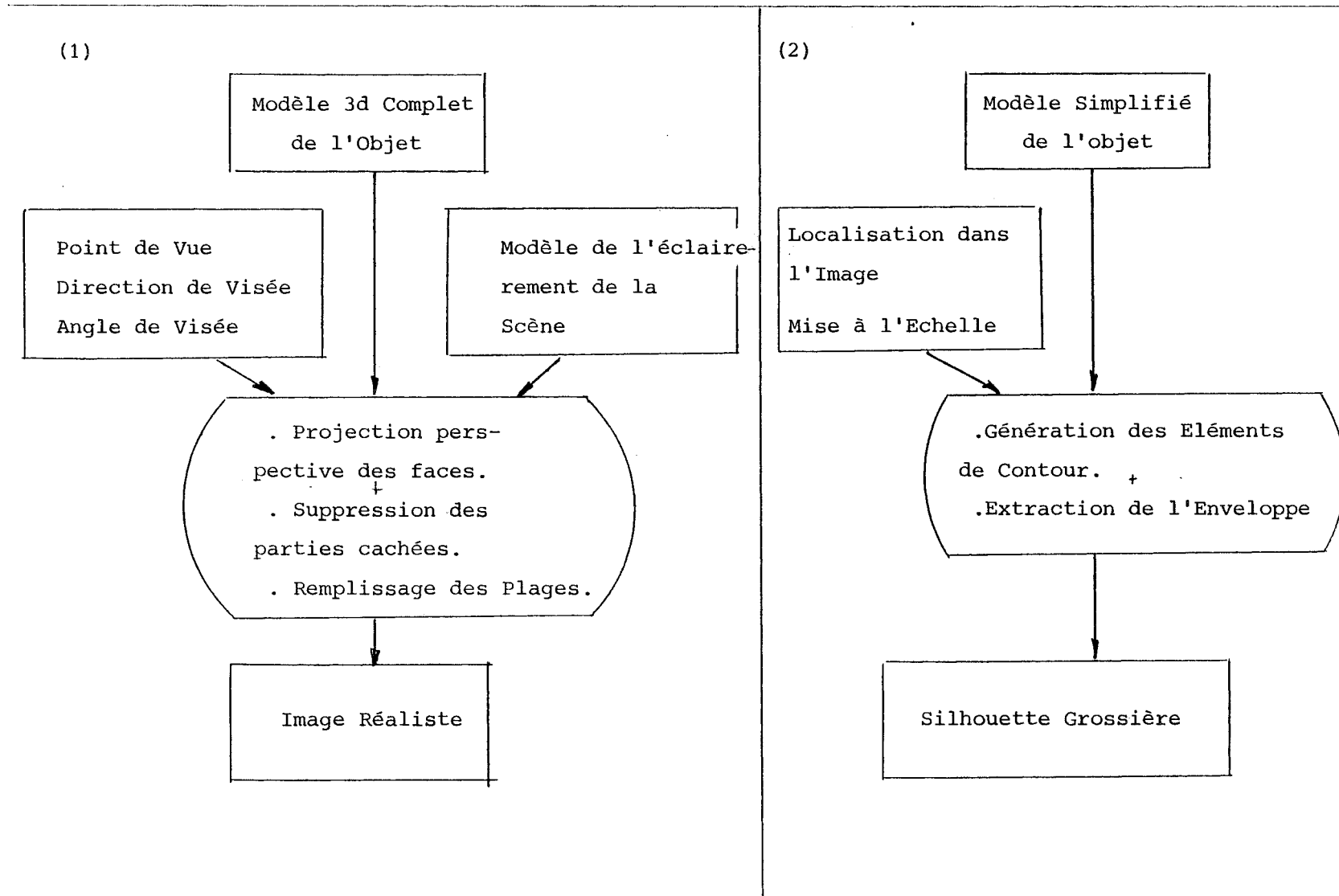


Figure Iv.2. : Synthèse d'Images Réalistes (1) et
Génération de la Silhouette Guide de Segmentation; (2)
Aspects Communs et Différences.



ment déterminé. Il est clair au contraire que le contour d'un même objet, varie de façon très importante pour des conditions de prise de vue différentes. Il semble même, au premier abord, que dans la plupart des cas, **la variabilité de la forme que peut prendre le contour de l'image d'un objet, masque** totalement toute ressemblance, **tout aspect commun** à ces multiples contours.

Il sera donc évidemment impossible de réduire la modélisation d'un objet à la simple description de sa silhouette.

L'idée naturelle qui sous-tend la modélisation est la suivante :

Le contour de l'image d'un objet dépend beaucoup du point de vue sous lequel la photographie a été prise, mais il dépend au moins aussi fondamentalement de la **structure géométrique propre de l'objet.**

C'est principalement cette information de structure, qui est indépendante d'une image particulière à traiter, que le modèle doit renfermer. Elle constituera à la fois une représentation canonique de toutes les formes que peut prendre le contour de l'objet et l'information a priori sur laquelle il sera possible de contrôler la vraisemblance ou de laquelle on pourra tirer parti.

IV.2.2. Sources de variation de la forme de l'image d'un objet.

Si l'on cherche à expliciter plus finement ce qui dans une photographie détermine la forme de l'image d'un objet, trois types d'influences apparaissent :

- Les conditions de prise de vue,
 - . axe de visée,
 - . angle du champ et/ou distance objet-caméra ;
- La géométrie propre des objets de la classe ;
- la géométrie de la scène.

On doit noter ici que l'illumination de la scène n'intervient pas directement sur la forme à détourner alors qu'elle intervient de façon majeure dans la formation de l'image.

En réalité les conditions d'illumination peuvent quand même affecter de façon capitale la netteté des contours. Dans le meilleur des cas, il s'agit d'une influence positive, mais au pire, il y a là une source d'ambiguïtés sur les contours (cf § II.4.3.3.). De plus on cite au §V.1.3. des travaux qui montrent que l'influence de l'illumination peut permettre de récupérer une information de profondeur a priori absente sur une photo.

On remarque que de nombreuses expériences menées à l'heure actuelle dans le domaine de la robotique, tirent largement profit de conditions particulières qui réduisent beaucoup la variabilité de la forme du contour d'un objet :

- distance objet-caméra, angle et axe de prise de vue fixés ;
- objets rigides ;
- objet à une seule (ou quelques) position(s) d'équilibre ;
- pas de recouvrement entre objets ou à la rigueur, "semi-vmc".

Pour cette possibilité qu'offre le domaine de la robotique d'imposer de telles hypothèses restrictives sur les scènes, c'est un champ d'expérimentation privilégié pour des modélisations relativement simples. Cf par exemple /PERKINS 78/ et /LEVINE 81/.

Pour les photographies d'intérêt général, au contraire, il est impossible d'introduire de telles contraintes, le corpus existe préalablement et la modélisation doit s'y adapter.

Les conditions de prise de vue, pour commencer, peuvent être à peu près quelconques et surtout elles sont inconnues à l'avance. (Toutefois, certains angles de visée et certains cadrages sont plus courants et l'on pourrait en tenir compte).

L'influence de la géométrie propre de l'objet semble relativement simple à caractériser. Elle répond aux lois de la projection perspective du volume de l'objet sur le plan image. Mais cette simplicité n'est qu'apparente. En effet, elle correspond à un

cas bien particulier : celui d'objets ayant une forme intrinsèque **invariable**. Ainsi apparaît pour le cas général, une hiérarchie entre objets rigides, objets articulés et objets déformables. Le cas des objets articulés sera abordé plus en détail dans le prochain chapitre à propos de l'exemple du modèle du pantin.

Dès lors que l'on est parvenu à établir le modèle d'un objet à détourer - i.e. l'information de structure nécessaire pour générer **toutes** les formes que peut prendre son contour -, il est absolument nécessaire, même au prix d'une augmentation minime de la complexité, d'étendre le modèle d'un objet à tous ceux d'objets de structure analogue : si par exemple on modélise pour le détournage un modèle d'automobile, il devra s'appliquer tout aussi bien à toutes les berlines.

En fait nous parlerons désormais sans distinction des objets d'une classe aussi bien à propos des objets différents acceptant la même représentation canonique qu'à propos des divers avatars d'un objet unique.

Si la modélisation de la forme de l'image des objets apparaît très complexe, deux remarques permettent néanmoins de l'envisager :

- en cas d'incapacité pour le programme d'aboutir au résultat souhaité, et à condition que celui-ci s'en rende compte lui-même, il existe une possibilité de recours ultime à l'opérateur; ce dernier devra alors débloquer la situation en donnant au programme un élément manquant;

- d'autre part, il ne faut pas oublier que le détournage n'est pas initialement une tâche de reconnaissance. La reconnaissance est simplement un des moyens disponibles pour parvenir au détournage.

Finalement les deux remarques se résument en notant que le détournage interactif se place dans une optique différente de la reconnaissance automatique (comme cela peut-être le cas en robotique, par exemple).

IV.3. Le modèle en génération.

Le passage du modèle d'une classe d'objets à des portions de son contour, est une tâche que l'on peut qualifier de générative dans la mesure où, d'une description normalisée de la classe et de paramètres fixant une occurrence, on détermine la description précise de l'âge de l'objet dans ce cas particulier.

IV.3.1. Génération de la silhouette guide pour la segmentation.

La génération de la forme guide à partir d'un modèle est une entreprise que l'on peut rapprocher de la synthèse d'images réalistes bien qu'ici, le problème soit d'une complexité bien moindre.

En synthèse d'images, les contours générés sont d'abord beaucoup plus complexes - tous les détails fins doivent être construits, mais surtout le remplissage des plages fait intervenir la structure de l'objet, l'éclaircissement qu'il reçoit, la texture de sa surface... Enfin, les modélisations mises en jeu sont très sophistiquées et en particulier ne sont pas nécessairement déterministes (modèle probabiliste de synthèse de l'image de montagnes /CARPENTER /; synthèse de textures imitant des textures naturelles /GAGALOWITCZ /; synthèse de l'image d'un danseur /BADLER, O'ROURKE et TOLTZIS 79/...).

La bien plus grande simplicité du contour grossier d'un objet comparé à son image réaliste, permet de se contenter ici d'un modèle de génération beaucoup plus sommaire et à proprement parler, schématique. Il n'est en particulier pas indispensable de disposer de la représentation 3-D complète du volume d'un objet pour en générer un guide de détournage suffisamment précis.

En effet, le cas fréquent où, pour des conditions courantes de prise de vue, l'objet ne s'étend que sur une profondeur faible comparée à la distance objet-caméra, les erreurs introduites par une modélisation strictement 2D sont négligeables par rapport à la précision grossière de la silhouette guide recherchée, et une telle modélisation sera donc tout à fait suffisante.

Dans l'étape de génération à partir d'un tel modèle, l'aspect de projection perspective disparaît complètement, et le travail à réaliser consiste uniquement à **extraire l'enveloppe** de tous les éléments de contour obtenus après que l'on ait **fixé les paramètres** du modèle et effectué la **localisation** et la **mise à l'échelle** (cf. fig. IV.2), données irréductibles :

même dans le cas le plus trivial que l'on peut envisager, celui d'un objet sphérique dont la forme du contour est donc indépendante du point de vue, la localisation et la mise à l'échelle restent nécessaires.

L'exemple d'une telle procédure pour le modèle de l'image de personnages apparaissant sur une photographie est l'objet du chapitre V : le modèle du pantin.

IV.3.2. Mise à l'échelle.

Echelle intrinsèque

La taille de l'image d'un objet rigide dépend évidemment des conditions de prise de vue de la photographie. Un bon modèle pour le détournage d'un objet doit permettre de construire sa silhouette grossière pour des distances et angles de prise de vue différents. Le modèle doit donc être paramétré par l'échelle à laquelle apparaît l'objet sur la photo, c'est-à-dire par la correspondance qui existe entre la taille d'un élément repéré sur l'image et la taille de l'élément correspondant de la scène réelle. On a appelé ce paramètre, échelle intrinsèque de l'apparition d'un objet sur une photographie, car nous allons voir qu'une autre notion d'échelle peut être introduite.

On remarque que l'échelle intrinsèque n'est pas unique pour une photo; un exemple le montre : les images de 2 personnages à des distances différentes de l'objectif apparaissent évidemment à des échelles intrinsèques différentes.

Pour l'observateur humain, l'échelle intrinsèque de l'image est extraite, très aisément en général, du contexte. Cette remarque est suggérée par la facilité avec laquelle un objet hors

d'échelle devient ambigu. Voilà donc 2 contributions simples susceptibles d'être apportées par l'opérateur.

L'échelle intrinsèque de l'image d'un objet étant connue la mise à cette échelle du modèle de l'objet est aisée ; il s'agit simplement d'appliquer une homothétie aux longueurs des éléments du modèle.

Echelle extrinsèque :

Cette deuxième notion d'échelle est apparue lorsque l'on s'est interrogé sur la subjectivité des contours : si l'on donne une photographie à détourer à plusieurs observateurs, le tracé du contour d'une zone précisée oralement, n'est pas rigoureusement identique. Cela renvoie en partie aux différences des modèles des observateurs mais surtout à une petite marge d'incertitude. Il existe cependant une échelle à partir de laquelle tous les contours se confondent. Il est inutile de travailler à une échelle plus fine ; les résultats obtenus à cette échelle devront simplement être interpolés.

IV.4. Inférence : à la charnière d'analyse et de génération.

Toutes les informations que doit nécessairement renfermer le modèle pour permettre de générer la silhouette d'un objet vu **depuis n'importe quel point de l'espace**, n'est pas indispensable pour construire la silhouette de l'objet vu **depuis un point particulier**.

Autrement dit, la génération du contour d'un objet ne nécessite pas de connaître précisément la valeur de tous les paramètres de la description d'une occurrence de son image ; Certains paramètres n'interviennent directement que dans la construction d'éléments de bords n'appartenant pas à l'enveloppe finale.

Dans l'exemple d'un parallélépipède opaque (fig. IV.3), la localisation dans l'espace (modèle 3D) ou dans l'image (modèle 2D) des sommets A et B, est sans importance pour décrire la silhouette guide.

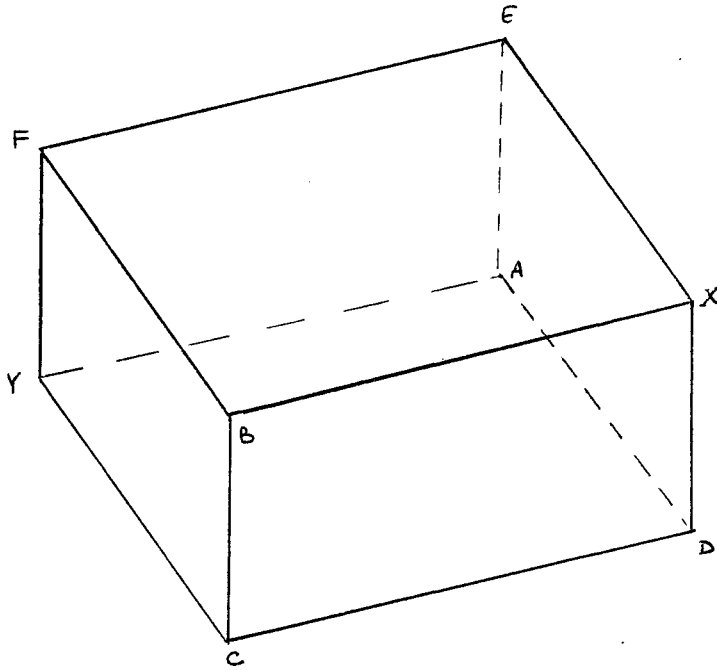


Figure IV.3.

Mais inversement, il arrivera que tous les éléments nécessaires pour établir la silhouette grossière ne peuvent être déterminés ; on devra alors tenter de déduire des connaissances disponibles (contexte) qu'elles concernent ou non le contour, la valeur des paramètres manquants.

Dans l'exemple du parallélépipède, la localisation du point B - qui n'apparaîtra plus dans la description du contour -, et des points C et F, permet de déduire l'emplacement des points X et Y s'ils manquaient.

Dans l'exemple précédent, le contexte et la règle d'inférence sont de nature géométrique ; ils permettent de déduire à partir d'éléments d'assez bas niveau, d'autres éléments de même niveau.

Le même type de mécanisme doit pouvoir être appliqué à d'autres niveaux et en particulier aux niveaux les plus élevés du modèle ; ceci devrait permettre de faire intervenir le contexte au niveau sémantique et les règles de son influence (vraisemblance/incompatibilités) tels qu'ils peuvent être énoncés verbalement par un expérimentateur.

Les techniques à mettre en oeuvre pour réaliser ces déductions dans un univers des connaissances incomplet, rappellent celles des systèmes dits experts /LAURIERE 82/.

IV.5. Analyse selon un modèle

IV.5.1. Intérêt d'une analyse automatique même très partielle.

Que ce soit pour générer la silhouette guide grossière finale, ou pour actionner le mécanisme d'inférence, il est nécessaire de s'appuyer sur des prémisses ; c'est-à-dire ici sur l'identification à partir de l'image, de certains éléments du modèle, ou sur l'évaluation de certains de ses paramètres pour le cas à traiter.

Il est donc nécessaire qu'une étape antérieure tente d'extraire de l'image une description de celle-ci dans les termes du modèle. C'est l'étape d'analyse.

L'automatisation intégrale d'une telle tâche, pose des problèmes non encore résolus dans le cas général de photographies complexes comme celles envisagées ici ; par exemple la représentation et l'utilisation de grandes quantités de connaissances.

Devant l'impossibilité actuelle de parvenir à une analyse automatique complète, et vues l'aisance et la rapidité avec lesquelles, au contraire, l'observateur humain s'acquitte de la même tâche, on est tenté de la laisser en totalité à sa charge.

Il est cependant intéressant, comme pour n'importe quel système interactif, de tenter d'alléger au maximum les interventions de l'opérateur. Deux choses peuvent y contribuer :

- l'établissement de l'interaction à un niveau sémantique ;
- l'analyse automatique partielle.

Le premier point fera l'objet du paragraphe suivant (IV.6).

Les résultats partiels obtenus par analyse automatique sont importants à deux titres : d'abord parce qu'ils réduisent d'autant l'intervention de l'opérateur ; mais aussi parce qu'ils permettront de diriger les interventions qui resteront nécessaires, avec une meilleure efficacité.

IV.5.2. Extraction de primitives et reconstruction d'éléments de niveaux supérieurs.

Quel que soit le modèle hiérarchique mis en oeuvre, l'analyse présente deux phases distinctes :

Dans un premier temps, on doit identifier dans l'image des éléments de base du modèle. Par analogie avec la théorie des langages, on parlera de symboles terminaux ou de primitives. La méthode d'extraction dépend évidemment du choix qui a été fait quand à la nature des primitives du modèle ; ou plutôt on va voir que l'on détermine leur nature de manière à ce que l'on sache les extraire.

La deuxième phase d'analyse consiste à réaliser la combinaison des informations dont on dispose à un niveau donné et à tous les niveaux inférieurs, pour reconstituer des éléments de niveaux supérieurs dans la hiérarchie. Un tel problème est malencontreusement d'une grande complexité algorithmique (exponentielle), puisqu'il consiste à vérifier parmi tous les arrangements d'éléments identifiés, ceux qui vérifient les conditions imposées par le modèle.

Le choix de la nature des primitives est guidé par les deux contraintes contradictoires ci-après : elles doivent être suffisamment simples pour pouvoir être extraites de façon peu énumérative, et à la fois suffisamment complexes pour ne pas apparaître en trop grand nombre au départ de la phase d'énumération.

Ce compromis peut se poser en terme de "portée de l'information" (cf. I.5.2) /BULLOCK 76/ présente la problématique du choix entre primitives ("features") extraites par mesure :

- 1°) au niveau du pixel,
- 2°) locale,
- 3°) globale.

Il constate qu'en général la complexité d'extraction est faible dans le premier cas, moyenne dans le deuxième cas, insurmontable dans le troisième. Enfin, il évalue l'**invariance** des diverses primitives, aux transformations suivantes : modification de l'intensité lumineuse, de l'échelle, rotation.

Cette idée d'invariance permet de définir à nouveau la fonction du modèle : représentation de l'image d'un objet d'une classe donnée, invariante pour la position de l'objet,...

IV.6. Le rôle du modèle dans l'interaction

on a déjà noté qu'il ne serait pas réaliste de chercher à obtenir à coup sûr une identification complète de tous les paramètres d'un modèle par la seule analyse automatique de l'image à traiter.

C'est pourquoi, un système de détournage devra être interactif : c'est-à-dire qu'en cas de besoin le système devra provoquer une intervention de l'opérateur.

En intervenant, l'opérateur - pour qui l'analyse de l'image est instantanée mis à part quelques cas énigmatiques auxquels nous ne nous intéresserons pas - doit apporter des éléments manquants pour que l'analyse puisse progresser jusqu'à permettre de générer le contour ou activer le mécanisme d'inférence.

L'intervention de l'opérateur peut par exemple consister à introduire des "symboles terminaux" qui n'ont pu être extraits automatiquement de l'image.

Mais pour des raisons de confort et d'efficacité d'interaction, il est probablement plus judicieux d'essayer d'établir le dialogue à un autre niveau. L'usage d'un modèle hiérarchique le permet. En effet, le mécanisme d'inférence au niveau de la vraisemblance, nécessite d'introduire dans le modèle des éléments lourdement chargés de sens pour l'opérateur ; ce sont ceux-là même qu'il utilisent naturellement pour décrire le résultat de sa propre analyse de l'image. On parlera de haut niveau sémantique ou de niveau linguistique.

Un exemple de vocabulaire lié à des éléments de haut niveau dans le modèle, sera donné au cours du chapitre suivant.

Enfin, d'autres informations, que la forme grossière, peuvent être mémorisées lors de la modélisation d'une classe d'objets ; c'est par exemple la manière dont le contour devra, dans chaque région, suivre le segment-guide : portions de contours "tourmentées" dans la zone des cheveux, "rigides" dans d'autres régions. On note leur expression naturelle en termes linguistiques et non numériques.

Bibliographie du chapitre IV.

Badler, O'Rourke et Toltzis

A Spherical Representation of a Human Body for Visualizing Movement.

Proc. IEEE, vol 67 n°10, 1979.

M. Brady

Computer Vision.

North Holland, 1981.

Bullock

Finding Structure in Outdoor Scenes.

Pattern R. & Artificial I., Chen Ed, Academic Press, 1976.

Carpenter

Gagalowicz

Hanson et Riseman

Computer Vision Systems.

IEEE Trans. Comp., C 27-2, 1978.

J.L. Laurière

Représentation et Utilisation des Connaissances.

TSI n° 1 et 2, Dunod 1982.

Levine et Ting

Intermediate Level Picture Interprétation using Complete Two-dimensional Models.

CGIP 16, 1981.

Roger Mohr

Généralisation de la Notion de Langage à Contexte Libre.

Application à l'Analyse Syntaxique de Figures.

Perkins

A Model Based Vision System.

IEEE Transaction on Computers, C 27-2, 1982.

CHAPITRE 5

=====

Chapitre V : DETOURAGE DE PERSONNAGES : LE MODELE DU PANTIN.

V.1. Le Personnage, un Objet Articulé.

V.1.1. Les Objets Articulés.

V.1.2. Les Modèles du Corps Humain.

V.1.3. Modèle 2D vs Modèle 3D.

V.1.4. Le Modèle du Pantin.

V.2. Génération de la Silhouette du Pantin.

V.2.1. Les Points-clé de la Silhouette.

V.2.2. Evaluation de l'Echelle à partir des Points-clé.

V.2.3. Extraction du Contour Polygonal.

V.3. Manipulation Linguistique du Modèle du Pantin.

V.3.1. Alternative à l'Introduction des 13 Points-clé.

V.3.2. Les Termes de l'Interaction.

V.3.3. Problèmes de Mise en Oeuvre.

V.3.3.1. Occlusions Partielles.

V.3.3.2. Nécessité d'un Modèle 3D.

V.3.3.3. Problème de Quantification.

CHAPITRE V :

DETOURAGE DE PERSONNAGES : LE MODELE DU PANTIN.

Le cas des personnages dont l'image doit être extraite de la photographie où elle figure, est intéressant à plus d'un titre.

C'est tout d'abord un cas très courant; et il est donc particulièrement bien adapté pour juger du gain que la modélisation peut procurer. Modéliser une classe d'objets peu fréquents serait au contraire, un investissement que l'allègement des interventions d'un opérateur ne parviendrait jamais à rentabiliser.

Le cas du personnage est aussi très important du point de vue théorique parce qu'il est typique du problème plus général posé par toute une famille de classes d'objets : les objets articulés.

Enfin, pour ce qui concerne l'aspect linguistique de l'interaction, le cas du personnage renvoie à un vocabulaire aussi élémentaire que courant.

V.1. Le personnage, un "objet" articulé.

V.1.1. Les objets articulés.

La modélisation des objets déformables, en vue d'en prévoir la silhouette grossière présente bien évidemment la difficulté suivante :

Outre les conditions de prises de vue, la forme intrinsèque de l'objet au moment de la saisie influe d'une façon majeure sur la forme-guide à fabriquer.

Si le cas général des objets déformables semble trop complexe à traiter, le cas particulier des objets articulés présente le double avantage de mieux se prêter à une modélisation simple d'une part, et d'autre part de permettre une approximation suffisante d'un bon nombre d'objets déformables.

Par définition, les objets articulés sont constitués de parties rigides parmi lesquelles sont établies des liaisons mobiles.

Le nombre de parties rigides et la description de chacune d'entre elles, fixés une fois pour toutes, doivent être inclus dans le modèle lors de son élaboration. Il en est de même de la **structure de l'objet articulé**, c'est à dire le nombre de liaisons mobiles et, pour chacune d'elles, l'identification des deux éléments qu'elle relie.

L'ensemble constitue les informations essentielles connues a priori sur la géométrie de l'objet.

A un moment donné, la forme d'un objet articulé est déterminée par la **valeur instantanée** de chacune de ses articulations. On l'appellera **position propre ou intrinsèque de l'objet**, pour la différentier de la position de l'objet dans l'image.

La valeur d'une articulation identifiera la position relative des deux parties rigides liées, parmi toutes les positions relatives qu'elles peuvent présenter. Dans le cas d'une modélisation 3D, la valeur prise par une articulation, est donnée par la transformation qui fait passer du repère fixe lié à l'une des parties rigides à celui lié à l'autre (voir fig. V.1.). Pour une modélisation 2D, la valeur de l'articulation est celle de l'angle formé par la projection des axes des deux parties sur le plan de l'image.

Le modèle d'une classe d'objets articulés, nécessaire pour décrire complètement sa géométrie et son image, dans tous les cas,

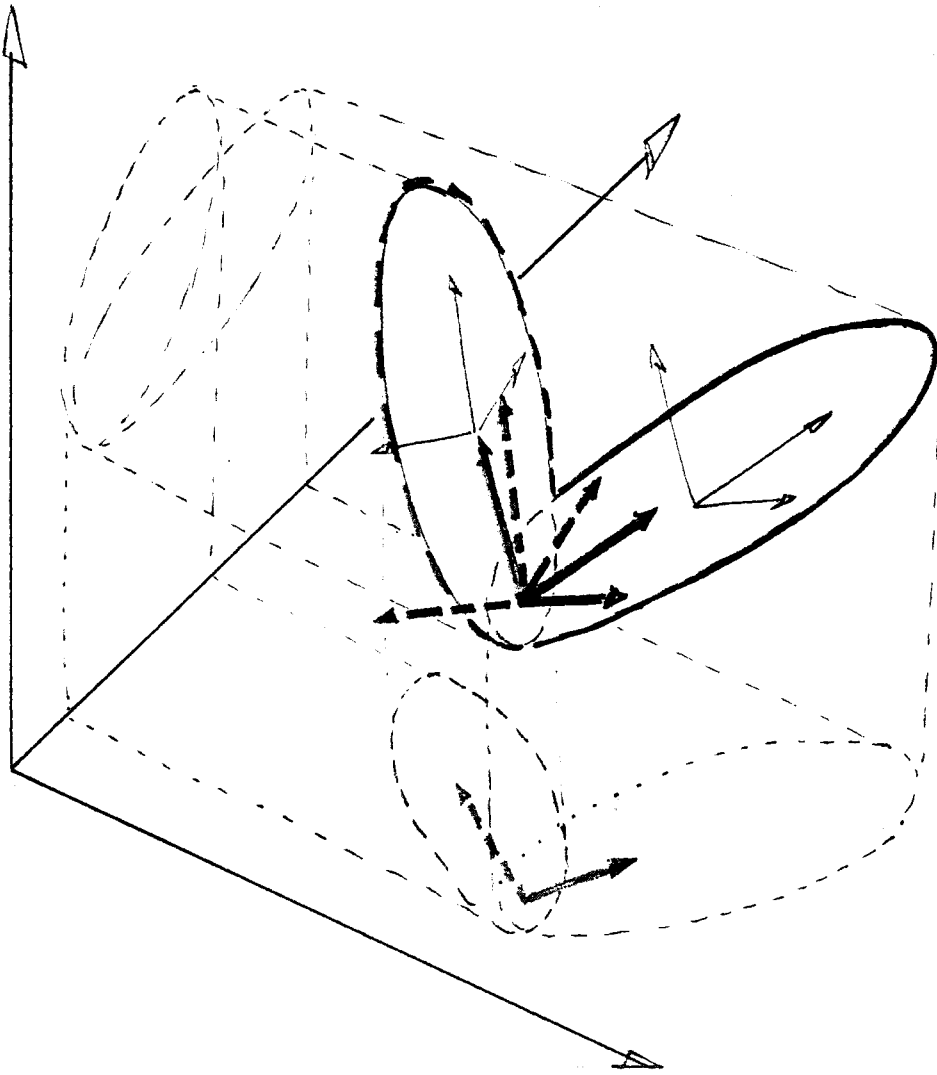


Figure V.1. - Objet articulé : "valeur d'une articulation".

regroupe des informations des trois types (voir fig. V.2.) :

- les informations communes à tous les objets de la classe;
- les paramètres déterminant chaque position intrinsèque;
- et ceux fixant les conditions de prise de vue.

V.1.2. Les modèles du corps humain.

La modélisation du corps humain comme objet articulé se déduit assez naturellement des grands traits de son anatomie :

La structure est fixée par le squelette; les demi-membres, la tête, le tronc et l'abdomen forment des sous-ensembles qu'il est raisonnable de considérer comme rigides.

Sur ce principe général, diverses variantes de modélisation ont été proposées :

- /HERMAN 78/ analyse la position d'un bonhomme fil-de-fer ("stick-figure").

- /BADLER et al. 79/ reconstituent des images de bonne qualité du corps humain (celui d'un danseur). Pour cela ils font appel à une modélisation assez fine des parties rigides : elles sont représentées par le recouvrement de multiples sphères.

- /NEVATIA et BINFORD 77/ analysent sur des bases proches, "l'image 3D" d'une scène; mais leur appareil de saisie associe à chaque point de l'image 2D habituelle, la profondeur correspondante. Cette information supplémentaire facilite évidemment la segmentation.

Pour l'usage que nous voulons en faire, une description précise des parties rigides serait d'un intérêt tout à fait illusoire. En effet, les variations entre images de parties rigides homologues, pour des personnages différents, cacheraient largement les détails introduits.

C'est pour cette raison que nous avons choisi de représenter les éléments du pantin par les formes les plus élémentaires possibles.

V. 1.3. Modèle 2D vs modèle 3D.

Avant d'entrer dans le détail de la génération, il reste encore un choix fondamental à effectuer : doit-on mettre en oeuvre un modèle 2D (modélisation des images des objets) ou un modèle 3D (modélisation de la géométrie des objets dans l'espace)?

a) Un modèle 3D ne s'impose pas d'emblée dans la mesure où l'information brute en entrée (l'image) d'une part, et la description finale à laquelle il faut aboutir d'autre part, sont toutes les deux intrinsèquement planes. Ainsi, le problème du détournage diffère-il nettement du problème de segmentation abordé par Nevatia et Binford et cité plus haut, qui disposent en entrée d'une information 3D.

b) Par ailleurs, le modèle 3D entraîne une plus grande complexité des processus associés. Si le côté génératif n'est guère plus complexe en 3D qu'en 2D (on rajoute une étape de projection perspective), l'aspect analytique est beaucoup plus délicat à traiter en 3D qu'en 2D.

Toutefois des travaux récents montrent qu'il est possible d'extraire d'une simple image 2D et sur la base d'un modèle de l'illumination et de la réflexivité de la surface des objets, des informations 3D :

/IKEUCHI et HORN 81/ décrivent une méthode itérative qui, appliquée à des images synthétiques simples (objet de couleur unie et à une seule valeur de réflectance), permet d'évaluer en chaque point, l'orientation de la surface.

/WITKIN 81/ aboutit au même type de résultats sur des images réelles plus complexes par analyse des textures. Il se fonde sur le fait que bien que les textures soient a priori irrégulières, la projection perspective leur imprime une distorsion systématique.

c) Il reste que ces méthodes sont complexes; de plus l'orientation des surfaces en chaque point de l'image n'est qu'une

information 3D partielle de laquelle il faudrait encore dériver la profondeur.

V.1.4. "Le Modèle du Pantin".

Puisque le modèle du pantin est avant tout une illustration de l'aspect génératif de la modélisation, nous avons choisi un modèle 2D, sans nier pour autant l'intérêt des modèles 3D.

L'image d'un personnage est représentée par un ensemble de 14 cercles et de 13 quadrilatères comme cela apparaît sur la fig. V.3. Le recouvrement de leurs surfaces formera la silhouette-guide.

On constate, sur cette figure que la description des quadrilatères se déduit de la donnée des cercles : leurs sommets sont, soit le centre d'un cercle, soit un point de tangence. La procédure permettant de déterminer ainsi les quadrilatères traduit l'information de structure du modèle.

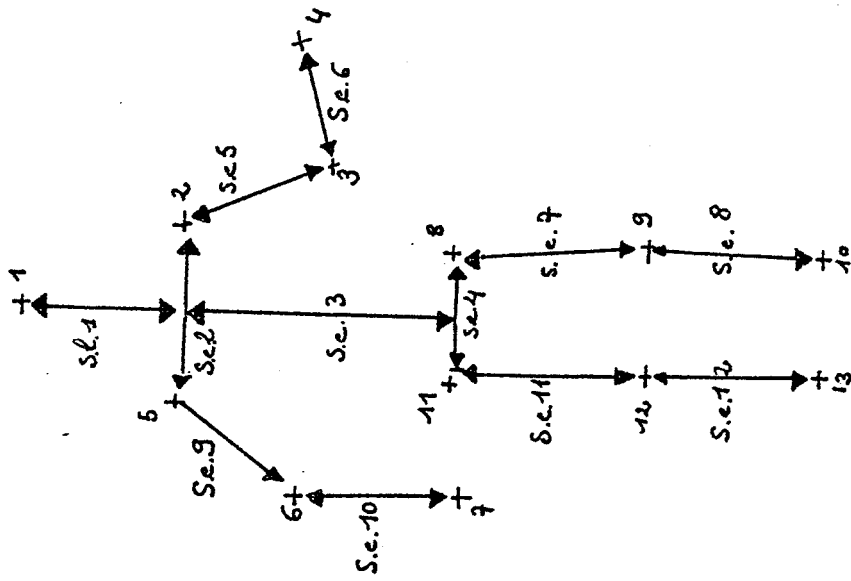
La position propre, la situation du personnage dans l'image, et l'échelle intrinsèque au sens du chapitre IV, déterminent une occurrence du modèle. Celle-ci étant fixée, on peut générer la forme guide de segmentation.

Deux représentations de celle-ci nous ont paru utiles :

1) la représentation sous forme d'une tâche binaire de l'intérieur de la silhouette,

2) et la représentation - dite graphique - consistant en la liste des segments et des arcs de cercle du contour.

Les deux représentations sont bien sûr, redondantes mais d'une part il est pratique de disposer des deux pour extraire les segments et les arcs qui suffisent à décrire le contour (voir § V.2.3) et d'autre part la deuxième représentation n'introduit aucun calcul important supplémentaire puisqu'elle est implicitement utilisée à la fabrication de l'image binaire de la silhouette. On se contente de mémoriser certains résultats intermédiaires.



-b-

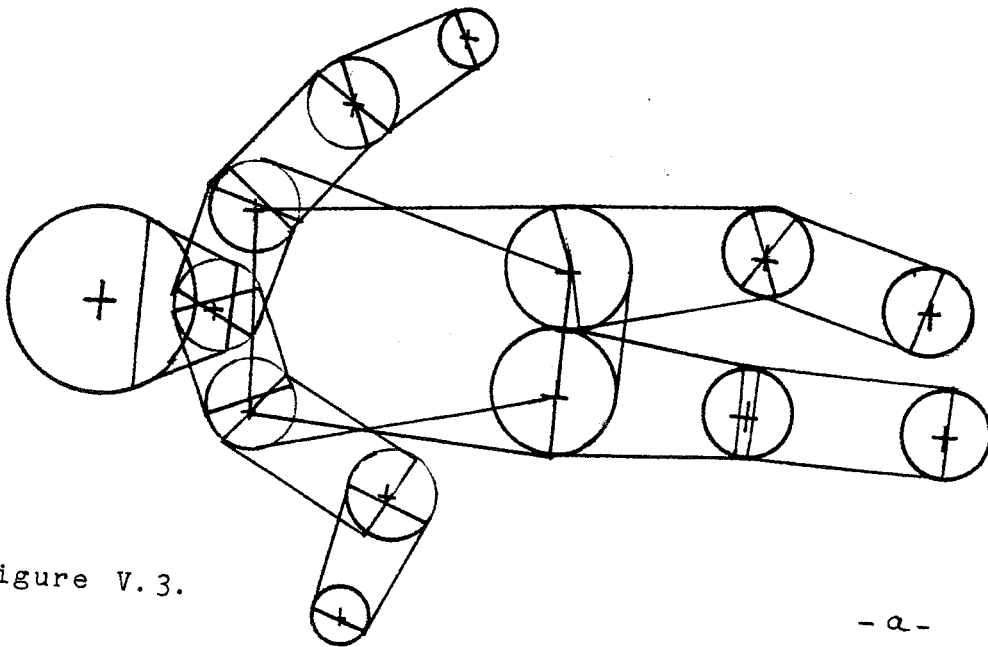


Figure V.3.

-a-

V.2. Génération de la silhouette du pantin.

V.2.1. Les points clé de la silhouette.

Pour décrire une occurrence du modèle, la donnée du centre et du rayon de tous les cercles n'est pas utile. En fait, le modèle est complètement paramétré dès que sont fixés les 13 points-clés suivants (figure V.3.b) :

- . Centre de la tête
- . Epaules (2 points)
- . Coudes "
- . Mains "
- . Bassin "
- . Genoux "
- . Chevilles "

La situation du personnage dans l'image et sa position propre sont données par ces points clé puisque par construction le centre du 14ème cercle est déduit du centre de la tête et des épaules.

Le rayon des cercles est fixé par rapport à l'échelle intrinsèque. Or celle-ci peut être évaluée sur les points clé, comme on le montrera dans le prochain paragraphe.

On justifie la possibilité de fixer le rayon de tous les cercles par rapport à un paramètre unique par la remarque qui consiste à dire que les diverses parties de "l'objet" modélisé s'étendent dans l'espace de la scène sur une distance faible par rapport à la distance objet-caméra. A cette condition, on peut admettre que l'échelle de représentation est unique pour l'objet entier (cf. discussion sur l'échelle intrinsèque au chapitre précédent).

V.2.2. Evaluation de l'échelle à partir de la donnée des points clé.

Le principe de l'évaluation est le suivant :

On a mesuré une fois pour toutes, sur un modèle de corps humain, la longueur de plusieurs parties approximativement rigides, et que nous appellerons segments étalon (s-e). Dans le cas de notre expérimentation, ces mesures ont été prises sur le pantin de bois de la figure V.4.

Lorsque la localisation des points clé a été précisée pour une image, on doit en extraire la longueur de l'image des s-e. On calcule alors le rapport de chaque mesure sur l'image à la mesure homologue sur le prototype.

Dans la liste des rapports ainsi obtenue, la valeur maximale permet d'identifier le s-e du personnage sur lequel l'effet de la perspective est le moins accentué : c'est-à-dire celui qui forme le plus petit angle avec le plan de l'image.

On fait alors la supposition qui consiste à affirmer que ce segment là, est **exactement** dans un plan parallèle à l'image, et donc que l'effet de la perspective sur la longueur est nul. Ainsi, le rapport associé à ce segment constitue finalement une évaluation par défaut, de l'échelle.

On se doute que la fiabilité de cette évaluation dépend du nombre de s-e et de leur choix. Deux remarques permettent d'établir une liste de s-e assurant une bonne évaluation de l'échelle :

1°) Si la liste est formée d'un nombre suffisant de segments "logiquement indépendants", on a statistiquement de bonnes chances que l'un d'eux au moins soit approximativement orthogonal à l'axe de visée ;

2°) Il est particulièrement intéressant que la liste comporte des couples de segments habituellement orthogonaux entre eux.

La figure V.4 présente :

- (a) le jeu des s-e que nous avons utilisé
- (b) et pour des positions-propre variées et pour plusieurs angles de prise de vue, le s-e choisi à chaque fois pour le calcul de l'échelle.

Un tel mécanisme d'évaluation de l'échelle s'avère donc tout à fait satisfaisant.

Le modèle du pantin n'est donc pas strictement 2D dans la mesure où ce procédé incorpore indirectement des connaissances 3D.

V.2.3. Extraction du contour polygonal.

La donnée des centres des cercles et de l'échelle -déduite de ceux-ci-, détermine tous les paramètres de la silhouette (fig V.5.).

Dans un premier temps, on en déduit la détermination des quadrilatères, d'après la structure reliant les parties rigides.

On peut alors construire les 2 représentations de la silhouette déjà décrites plus haut (tâche binaire, et "description graphique brute" -voir fig. V.5.b et c).

Un grand nombre d'arcs et de cercles de la description graphique brute n'interviennent pas dans le contour. L'extraction des arcs et des segments effectivement sur le contour (voir fig. V.5d) est réalisée à l'aide des 2 représentations précédentes (b) et (c).

L'algorithme d'extraction consiste à parcourir la représentation graphique brute et à en éliminer les parties qui apparaissent intérieures à la silhouette par test sur son image binaire. Par ce moyen, certains arcs et segments sont complètement supprimés alors que d'autres sont simplement écourtés à l'une ou aux deux extrémités.

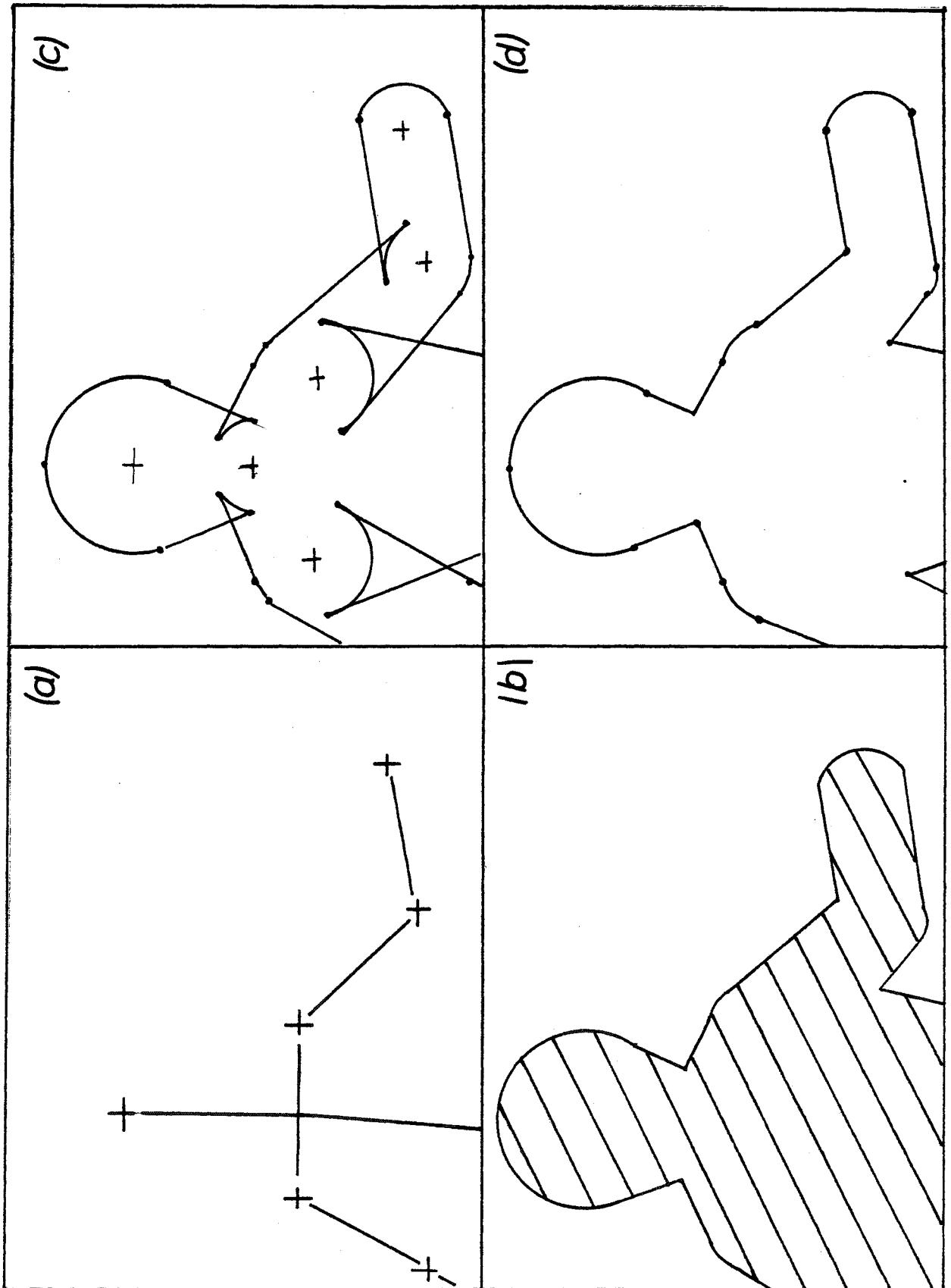


Figure V.5.

V.3. Manipulation linguistique du modèle du pantin.

V.3.1. Alternative à l'introduction des 13 points clés.

On ne s'intéressera pas dans le cas du pantin, à une tentative de positionnement automatique du modèle pour une image donnée ; on laissera cette tâche à l'opérateur. On doit donc préciser la forme sous laquelle celui-ci indiquera au système le résultat de sa propre analyse, c'est-à-dire la valeur du paramètre décrivant une occurrence du modèle.

On vient de voir que ces paramètres sont liés entre eux et que la localisation de 13 points clé suffit à fixer entièrement le modèle.

Le premier mode d'interaction consiste donc à demander à l'opérateur d'introduire les coordonnées de ces 13 points clé dans l'image. Dans la mesure où l'on dispose d'une table à numériser ou d'un appareil de saisie équivalent ("souris", boule roulante ou "manche à balai"), le volume de l'interaction est acceptable.

On doit toutefois prendre aussi en compte pour juger de l'interaction, la précision dans son geste, demandée à l'opérateur : s'il est impératif que l'opérateur fixe à un pixel près les points qu'il introduit, l'interaction prendra un tour pénible.

Ceci doit d'ailleurs être également évité dans les procédures où l'opérateur introduit directement le contour polygonal grossier (chap. II et III), qu'à fortiori ici où l'opérateur indique des points qu'il est de toutes façons incapable de localiser exactement car le "centre de la tête" d'un personnage est une notion floue.

Ceci n'enlève nullement l'intérêt d'une interaction de nature linguistique qui réduirait encore l'ampleur de l'interaction, et qui consisterait surtout une interface adéquate avec un système intelligent de compréhension d'images. En effet, le but ultime de tels systèmes est d'élaborer une description en des termes du langage courant, de l'image analysée.

On a donc évalué le vocabulaire nécessaire pour une telle interaction. On a situé son volume autour des quelques dizaines de mots, que l'on regroupe aisément selon le type d'informations du modèle et la fonction.

On peut noter ici une analogie avec d'autres domaines où il est nécessaire de décrire des positions du corps : il existe en radiologie un vocabulaire de positionnement du squelette (antéro/postérieur, etc ...); un vocabulaire remplissant la même fonction est utilisé par les chorégraphes.

V.3.2. Les termes de l'interaction.

Désignation des éléments du modèle.

Parties Rigides	Articulations	Autres points-clé
COU	NUQUE	TETE
TRONC-ABDOMEN	EPAULE (D/G)	MAIN (D/G)
BRAS (D/G)	COUDE (D/G)	PIED (D/G)
AVANT-BRAS (D/G)	HANCHE (D/G)	
CUISSE (D/G)	GENOU (D/G)	
JAMBE (D/G)		

Attributs des éléments du modèle

DROIT / REPLIE
OUVERT / FERME
LEVE / BAISSÉ
DEVANT / DERRIERE

Grâce à ces premières listes de mots, la **position propre** peut être décrite en fixant des attributs aux éléments ; mais on peut également proposer des mots identifiant toute une liste d'associations Elément-Attribut pour décrire complètement et une fois pour toute des positions courantes :

"Macros" de positionnement propre.

DEBOUT (couché)	ASSIS	MARCHE
ACCROUPI	AGENOUE	COURSE

On proposera dans le même ordre d'idée, des "macros" de positionnement partiel :

- BRAS-ECARTES - MAINS-SUR-LES-HANCHES
- ENJAMBEE

Les conditions de prise de vue .

Elles font appel à un autre vocabulaire qui sert à déterminer 3 informations :

- 1°) la rotation entre l'axe de visée et un repère fixé au personnage;
- 2°) la position de l'image du personnage dans translation de l'image;
- 3°) l'échelle intrinsèque.

. <u>Rotation</u>	FACE	3/4
	PROFIL	PLONGEE
	DOS	CONTRE-PLONGEE
. <u>Translation dans l'image</u>		
- AU CENTRE	A DROITE	A GAUCHE
- EN HAUT	EN BAS	
. <u>Echelle</u>		
- PLAN-D'ENSEMBLE (LOIN)		- EN-PIED (PRES)
- "PLAN-AMERICAIN" (*)		- GROS-PLAN

V.3.3. Problèmes de mise en oeuvre de l'interaction linguistique sur le modèle du pantin.

Plusieurs difficultés se posent pour introduire l'interaction linguistique dans l'expérience du modèle du pantin.

V.3.3.1. Occlusions partielles.

Les valeurs "plan-américain" et "gros-plan" proposées pour l'échelle font apparaître le problème d'une silhouette qui déborde

* On appelle "plan américain" en photographie et en cinéma, un cadrage présentant les 3/4 environ d'un personnage debout.

du cadre de l'image. Dans un pareil cas, toute une partie du modèle devra être inhibée, et plus particulièrement dans le dernier cas, où le modèle du pantin perdrait très probablement toute efficacité. Le terme "GROS-PLAN" constituerait donc en définitive l'activation d'une autre modélisation : le modèle de visage humain.

Le cas ci-dessus est un cas particulier du problème général des occlusions d'une partie de l'image de l'objet. La cause de l'occlusion peut être :

- (a) les limites du cadre de l'image,
- (b) un autre objet de la scène.
- (c) une autre partie de l'objet ("Devant/Derrière"), ou encore

Le 2ème cas renvoie à une interprétation globale de la scène qui impliquerait la modélisation de multiples objets.

Ceci peut indiquer à un autre titre la nécessité de changer d'échelle : l'individu n'est pas analysé de la même façon qu'un membre d'une foule; ou simplement l'utilité de décrire des obstacles. Notons toutefois qu'en général, le nombre des obstacles est réduit dans une photo bien cadrée.

Le dernier cas se pose en des termes différents : l'extérieur du modèle n'est pas en cause mais sa formule interne.

V.3.3.2. Nécessité d'un modèle 3D :

Le modèle du pantin tel qu'il a été présenté en début de chapitre est strictement un modèle "en deux dimensions" :

- 1°) Les points-clé sont déterminés dans le plan de l'image ;
- 2°) Ils constituent simplement une représentation concise d'une tâche bicolore qui répond à des règles particulières de construction.

Pour cette raison, cette modélisation est insuffisante pour permettre l'introduction immédiate de l'interaction linguistique dans les termes indiqués plus haut. En effet, celle-ci ne décrit

pas une tâche bicolore, mais le volume d'un objet situé dans l'espace ; en particulier l'interaction linguistique ébauchée ci-dessus cherche à fixer **indépendamment** la position propre du personnage dans l'espace et la condition de prise de vue de la scène, contrairement à la donnée du points-clés qui intègrent les deux types d'information.

Il est donc nécessaire de passer à une phase de projection perspective dans la génération de la silhouette guide.

V.3.3.3. Problème de quantification.

Les termes précisant les attributs à associer aux éléments du modèle, aussi bien que les termes déterminant les conditions de prise de vue, permettent de fixer **les valeurs de paramètres** qui mesurent des angles, des longueurs, l'échelle... Ces données sont de nature continue ; or les différentes valeurs que l'on propose pour les mesures en question, sont très peu nombreuses. Elles introduisent en fait une quantification extrêmement grossière (2, 3 ou quelques niveaux seulement). Ceci risque de s'avérer bien insuffisant.

La nécessité d'introduire un **"degré"** dans une indication est donc évidente. Sous forme purement linguistique, cela ne semble possible qu'au prix d'une multiplication inacceptable des termes de l'échange.

L'interaction linguistique s'avérant parfaitement inadaptée ici, il nous semble indispensable de disposer dans un tel mécanisme de communication, d'une échelle de valeur à usage général.

Bibliographie du chapitre V.

Badler, O'Rourke et Toltzis

A Spherical Representation of a Human Body for Visualizing Movement.

Proc IEEE vol 67-10, 1979.

Herman

Ikeuchi et Horn

Numerical Shape from Shading and Occluding Boundaries.

in Computer Vision, M. Brady éditeur, North Holland, 1981.

Nevatia et Binford

Witkin

Recovering Surface Shape and Orientation from Texture.

in Computer Vision, M. Brady éditeur, North Holland, 1981.

CHAPITRE 6

=====

Chapitre VI : ANALYSE D'IMAGES DE BATIMENTS.

VI.1. Les Images de Batiments Dans un Environnement Réel.

VI.2. Détection de Bords.

VI.2.1. Généralités.

VI.2.2. Détection de Bords par Calcul du Gradient.

VI.2.3. Réponse en Amplitude et en Direction d'un Détecteur de bords 3x3.

VI.3. Détection de Segments par Transformation de Hough.

VI.3.1. Transformation Paramétrique d'une Droite du Plan.

VI.3.2. Détection de Lignes Droites dans une Figure.

VI.3.3. Transformation de Hough d'une Image de Contours.

VI.4. Recherche des Maxima sur l'Image Transformée de Hough.

VI.5. Un Modèle Hiérarchique d'Images de Batiments.

CHAPITRE VI :

ANALYSE D'IMAGES DE BATIMENTS.

VI.1. LES IMAGES DE BATIMENTS DANS UN ENVIRONNEMENT REEL.

Nous présentons ici plusieurs difficultés apparues lorsqu'il s'est agi d'illustrer la fonction du modèle d'un objet à détourer, dans l'**analyse** de l'information fournie par l'image.

Cette opération d'analyse, correspond au parcours de bas en haut de la hiérarchie du modèle (cf. plus haut fig. IV.1.).

Le choix de la classe des images de bâtiments pour cette illustration peut être justifiée par la remarque suivante :

les bâtiments sont dans une certaine mesure, représentatifs d'une gamme plus large d'objets, **les objets manufacturés**. A ce titre, ils présentent le plus souvent des arêtes rectilignes vives ; ces dernières se traduisent en général sur l'image, par des **segments rectilignes de bord**. Ce n'est cependant pas toujours vrai ; en effet, sous certains éclairages, certaines arêtes n'apparaissent pas et inversement des bords tranchés peuvent apparaître, qui ne correspondent à aucune arête de l'objet, mais par exemple à une ombre portée.

Ces problèmes ont été clairement mis en évidence dans des expériences de vision par ordinateur, dans un environnement simplifié, celui des "scènes de blocs". /GUZMAN 68/ montre que l'interprétation en trois dimensions de l'image de la scène permet de régler les problèmes d'occlusions de certaines arêtes par d'autres blocs. On pourra voir aussi /KANADE 81/.

Les vues réelles de bâtiments apparaissant dans des photographies d'extérieur peuvent dans une certaine mesure être assimilées à des scènes de blocs, car les bâtiments sont grossièrement constitués de volumes prismatiques. Cependant des difficultés majeures supplémentaires apparaissent pour ce type d'images.

Ces difficultés sont liées d'une part à la grande complexité du "bloc" que peut constituer un bâtiment : juxtaposition et emboîtement de prismes multiples ; présences d'éléments secondaires par rapport à la forme prismatique mais importants par leur impact dans la formation d'une image complexe (ouvertures, cheminées).

Les difficultés viennent aussi de l'environnement plus complexe encore dans lequel le bloc apparaît : occlusions partielles par des objets d'une autre nature ; fond de luminosité et de texture non homogène ; présence d'objets non prismatiques.

Il reste malgré tout naturel de baser la modélisation sur la structure qui relie entre elles les arêtes d'un bâtiment ; et l'analyse selon le modèle, doit donc commencer par l'**extraction des segments rectilignes de bords** de l'image.

Une fois extraits les segments les plus apparents de la scène, on devra mettre en oeuvre l'analyse de la disposition de ceux-ci selon un modèle largement inspiré des modèles de scènes blocs.

Le principe de base consiste à retrouver les connexions entre segments puis à regrouper les connexions de manière à retrouver ceux qui forment des figures fermées, ou des points de rencontre de plus de 2 segments, et ainsi de suite jusqu'à reconstituer la structure de l'image du bâtiment.

VI.2. DETECTION DE BORDS.

VI.2.1. GENERALITES.

On appellera segment rectiligne de bord, dans une image, un segment de droite qui sépare localement l'image en 2 régions homogènes distinctes. L'homogénéité d'une région est en fait une notion complexe comme nous l'avons indiqué dans le chapitre II. Mais l'on peut dans un premier temps parler d'homogénéité en niveau de gris.

Plusieurs méthodes sont présentées dans la littérature de l'analyse d'image, qui permettent d'extraire d'une image numérisée, les longs segments rectilignes de bord. "Long" signifie ici que les segments s'étendent sur un grand nombre de pixels (plusieurs dizaines).

Toutes les méthodes ont en commun de procéder en deux étapes qui peuvent, suivant les cas, apparaître plus ou moins distinctement. Ce sont les étapes de détection de bord d'une part et d'extraction de segments ensuite.

L'étape de détection de bord a pour fonction de détecter en quels points de l'image, la luminosité varie localement de façon importante, suggérant par là la présence en ce point d'un élément de contour.

Sur ce point précis et fondamental existe une littérature très abondante à laquelle on accèdera à travers /ROSENFELD et KAK 76/ /BALLARD et BROWN 82/ et /DAVIS 75/. Deux classes de techniques sont souvent employées. L'une consiste à **évaluer le gradient du potentiel d'intensité lumineuse** (cf. infra VI.2.2); et l'autre consiste à chercher la "marche de luminosité" qui se rapproche au mieux de l'image au voisinage de chaque point.

Cette dernière technique proposée par /HUECKEL 71 et 73/ donne des résultats d'une qualité généralement reconnue comme très bonne mais elle a l'inconvénient d'être relativement couteuse en temps de calcul. En effet la méthode repose sur le calcul du produit terme à terme d'un voisinage d'image de l'ordre de 50 à 100 pixels avec 8 matrices prédéfinies différentes.

Or l'un des objectifs de Hueckel était l'insensibilité au bruit haute fréquence ; et, nous l'avons déjà noté, le corpus considéré ici présente peu de bruit H.F. Nous avons donc adopté une méthode plus simple et plus rapide, du type évaluation du gradient de luminosité.

Quelle que soit la méthode de détection employée, de nombreux auteurs ont jugé nécessaire d'appliquer un post-traitement. Le but ainsi visé est de réduire le volume de l'information de bord préalablement extraite, en conservant parmi celle-ci, la plus fiable.

Le post-traitement le plus immédiat est un simple **seuillage** : élimination des éléments de bords de valeur inférieure à une valeur fixée.

Une technique beaucoup plus fine consiste à **renforcer les éléments de bord présentant une bonne cohérence avec les éléments de bords voisins**, et inversement d'atténuer les autres. (Application de techniques de **relaxation** à l'extraction de bords /ZUCKER 77/ /SCHACHTER 76/).

Enfin, une fois extraits et éventuellement sélectionnés les éléments de bords de l'image, la deuxième étape consiste alors à former des regroupements parmi ceux-ci en fonction de leurs localisations et de leurs directions.

Ici encore réapparaissent à nouveau les deux démarches opposées de suivi et d'extraction en "parallèle".

Certains travaux utilisent des techniques de suivi : /NEVATIA & BABU 80/ /RAMER 75/, mais la technique la plus fréquemment utilisée est du deuxième type. C'est la **technique basée sur la transformation de HOUGH**.

Plusieurs articles présentent des applications de cette technique à l'extraction de segments dans des photographies de scènes naturelles : /DUDA & HART 72/, /O'GORMAN & CLOWES 76/, /DUDANI & LUK 78/. Nous avons utilisé cette technique (cf plus loin § VI.3.).

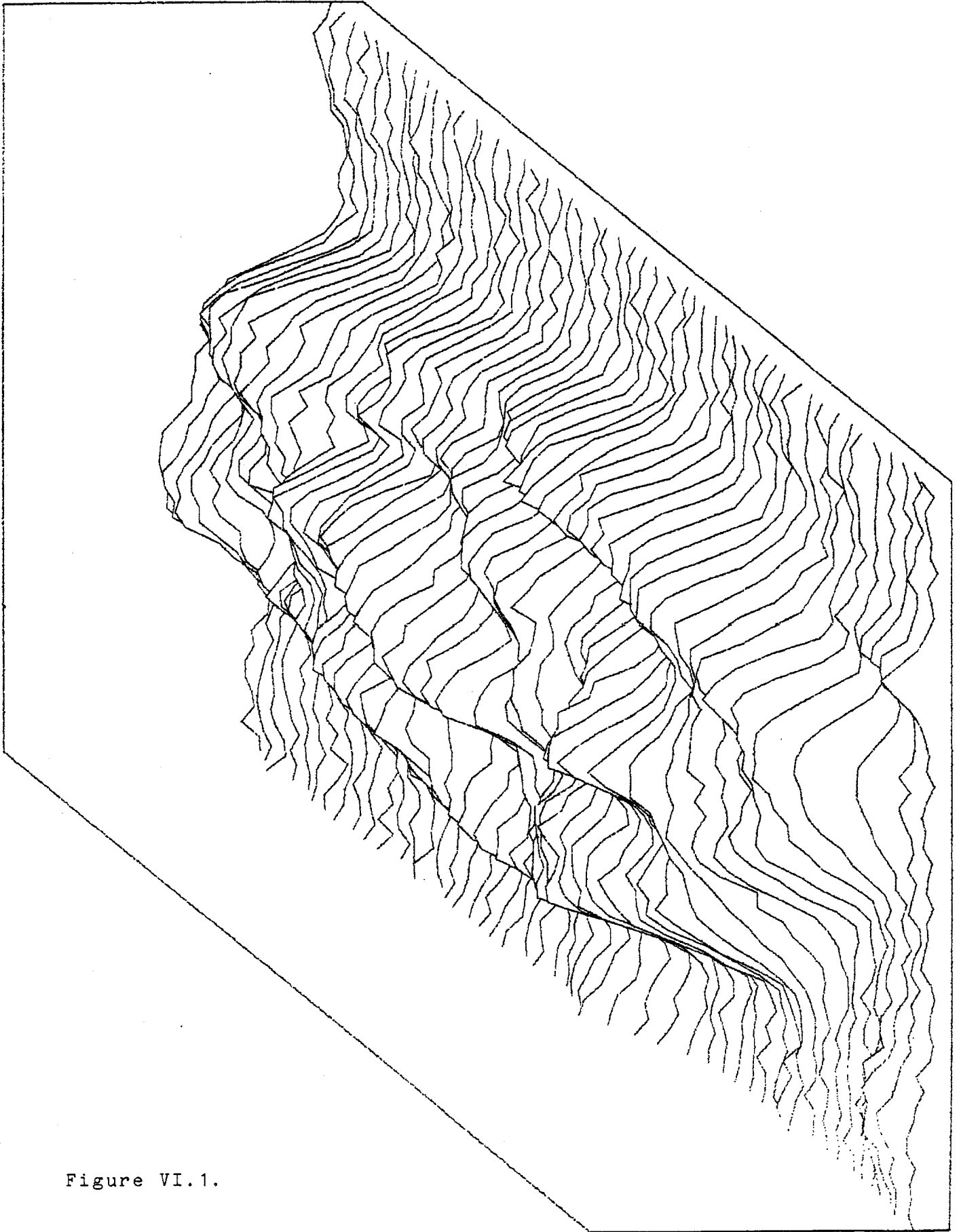


Figure VI.1.

VI.2.2. DETECTION DE BORDS PAR CALCUL DU GRADIENT.

Une image N et B est une fonction de R^2 dans R : à chaque point pris dans le domaine de définition (le cadre de l'image), elle associe la luminosité de celui-ci. On peut commodément se représenter l'image comme une surface gauche où l'altitude de chaque point représente cette luminosité (Cf. figure VI.1.). Le vecteur gradient représente alors en chaque point, la direction et l'inclinaison de la plus forte pente. Ce qui en terme d'image correspond à la direction et à la valeur locale de la plus forte transition Noir/Blanc.

Dans la mesure où l'image n'est disponible que sous forme échantillonnée, l'évaluation du gradient doit être réalisée par calcul de différences finies de valeurs de gris dans le voisinage de chaque point.

Le moyen le plus élémentaire pour évaluer le gradient de l'image en un point (i,j) est de prendre comme valeurs des composantes du vecteur gradient, les expressions suivantes :

$$f(i+1,j) - f(i,j) \quad \text{et} \quad f(i,j+1) - f(i,j)$$

Sur la base de ce principe, de nombreuses variantes ont été proposées. On les présente généralement sous la forme de la matrice des coefficients qu'il faut multiplier terme à terme au voisinage de chaque pixels pour obtenir en sommant ces produits, une composante du gradient. On parle de "convolution" de l'image par un "masque".

0	-1	1	0	-1	1	0	-1	3	3	-5	1	1	-1
		1	0	-1	2	0	-2	3	0	-5	1	-2	-1
1	0	1	0	-1	1	0	-1	3	3	-5	1	1	-1
ROBERTS		PREWITT		SOBEL		KIRSH		COMPASS					

fig. VI.2. les opérateurs de bords les plus simples.

Un autre point sur lequel des variantes ont été proposées est le choix :

1- de convoluer successivement à l'image deux masques -le deuxième étant obtenu par transposition du premier- et de disposer ainsi d'évaluations pour les deux composantes orthogonales du vecteur gradient ;

2- de convoluer successivement à l'image autant de masques que l'on désire différencier de directions; chaque masque étant construit par rotation du premier, donne une réponse maximale pour une portion d'image présentant un bord dans une direction particulière.

Le choix à ce niveau est lié à l'erreur de quantification que l'on désire ne pas dépasser sur les directions. On verra plus loin (§ VI.3.4.) que la direction précise des éléments de bords en entrée, est une donnée fondamentale de la technique de la transformation de Hough. Les variantes de type 2 ne sont donc pas adaptées ici.

VI.2.3. REPONSE EN AMPLITUDE ET EN DIRECTION D'UN DETECTEUR DE BORDS 3 x 3

Les performances des divers détecteurs de bords ont été comparées aussi bien empiriquement et pour des classes d'images particulières /ROSENF. et KAK 76/ que plus théoriquement /ABDOU et PRATT 79/.

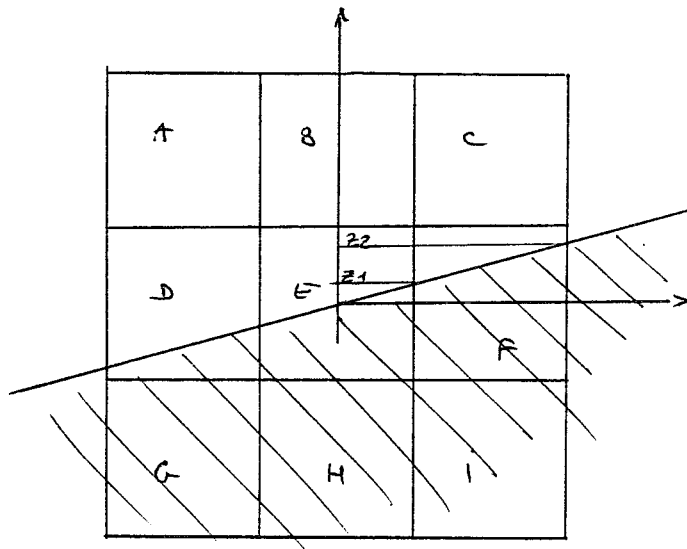
Les performances de ces détecteurs ne diffèrent sensiblement qu'en présence de bruit haute fréquence dans l'original. Ceci s'explique aisément par le fait qu'à l'effet différentiateur introduit par les poids de signes opposés, la convolution avec ces matrices ajoute un effet intégrateur plus ou moins important entre les pixels correspondant à des poids de même signe.

La réponse en amplitude et en direction d'un détecteur de bord à masque 3 x 3, peut être prévue par le calcul qui suit.

Pour cela on modélise un bord idéal de direction θ , d'amplitude unitaire et passant par le centre du masque, par une matrice 3 x 3 dont chaque élément a une valeur proportionnelle à sa surface éclairée.

Le calcul de la valeur des 9 pixels du "bord idéal" d'angle θ donne lieu à deux calculs selon la valeur de θ entre 0 et $\pi/4$. Il est clair que le calcul pour θ entre 0 et $\pi/4$, s'étend par symétrie aux autres octants.

1° cas : $0 \leq \theta \leq \arctg(1/3)$



$$A, B, C = 1$$

$$G, H, I = 0$$

$$E = 0,5$$

$$D = 0,5 + X$$

$$F = 0,5 - X$$

$$\text{où } X = (z_1 + z_2) * 0,5 = \text{Tang } \theta$$

Fig. VI.3.

D'où le calcul des composantes en X et en Y du vecteur gradient par application des deux masques de Sobel transposés.

$$m_x \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

$$B_x = 2F - 2D = -4 \text{ Tang } \theta$$

$$m_y \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

$$B_y = A + 2B + C = 4$$

D'où enfin les valeurs d'amplitude et de direction mesurées sur un "bord parfait" dans ce 1° cas :

$$AMP_m = 4^2 + 4^2 \cdot \text{tang}^2 \theta = 4 \cdot (1 + \text{tang}^2 \theta)$$

$$DIR_m = \text{arctg}(-B_x/B_y) = \text{arctg}(\text{tang} \theta) = \theta$$

2° cas : $\text{arctg}(1/3) \leq \theta \leq \pi/4$

$$A, B = 1$$

$$H, I = 0$$

$$E = 0.5$$

$$D = 1 - Y$$

$$F = Y$$

$$G = Z$$

$$C = 1 - Z$$

où Y et Z sont les aires des triangles ci-contre.

$$\text{tang} \theta = 0.5/x_1 = y_1/0.5 = y_2/1.5$$

$$x_1 = 1 / 2 \cdot \text{tang} \theta$$

$$y_1 = \text{tang} \theta / 2$$

$$y_2 = 3 \cdot \text{tang} \theta / 2$$

$$Y = (x_1 - 0.5) \cdot (0.5 - y_1) / 2 = (1 - \text{tang} \theta)^2 / 8 \cdot \text{tang} \theta$$

$$X = (1.5 - x_1) \cdot (y_2 - 0.5) / 2 = (3 \cdot \text{tang} \theta - 1)^2 / 8 \cdot \text{tang} \theta$$

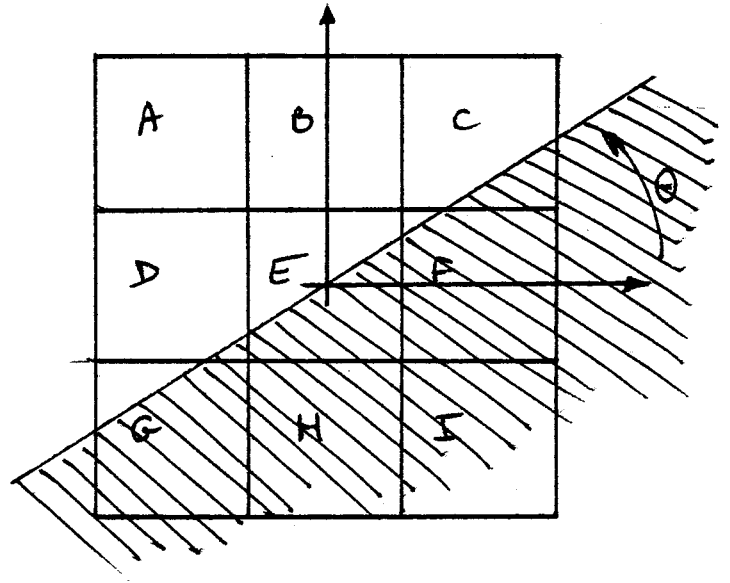


Fig. VI.4.

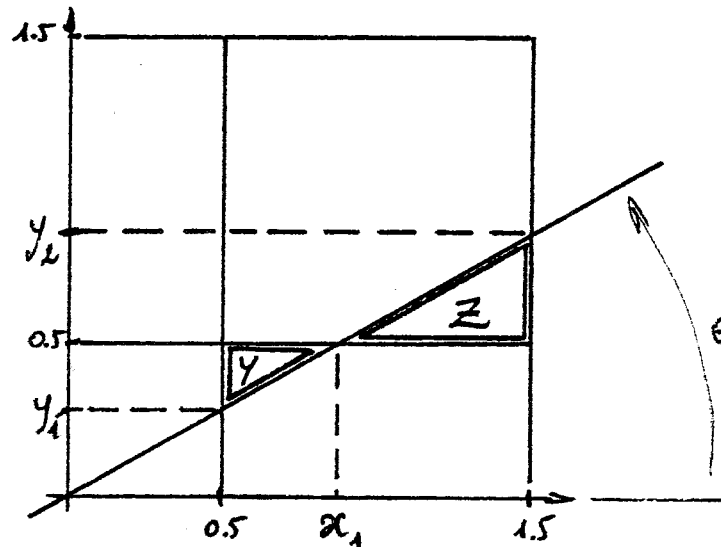


Fig. VI.5.

$$m_x \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

$$B_x = -A - 2D - G + C + 2F$$

$$B_x = 2 \cdot (2Y - Z - 1)$$

soit, en posant $t = \text{tang } \theta$, $B_x = (-7t^2 - 6t + 1)/4t$

$$m_y \begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

$$B_y = A + 2B + C - G - 2H - I$$

$$B_y = 4 - (3 \cdot t - 1)^2 / 4 \cdot t$$

d'où l'expression de l'amplitude mesurée

$$\text{AMP}_m = ((-7t^2 - 6t + 1)^2 + (16t - (3t - 1)^2)^2)^{1/2}$$

soit

$$\text{AMP}_m = (130t^4 - 312t^3 + 524t^2 - 56t + 2)^{1/2} / 4t$$

et

$$\text{tang}(\text{DIR}_m) = -B_x / B_y = \frac{7t^2 + 6t - 1}{-9t^2 + 22t - 1}$$

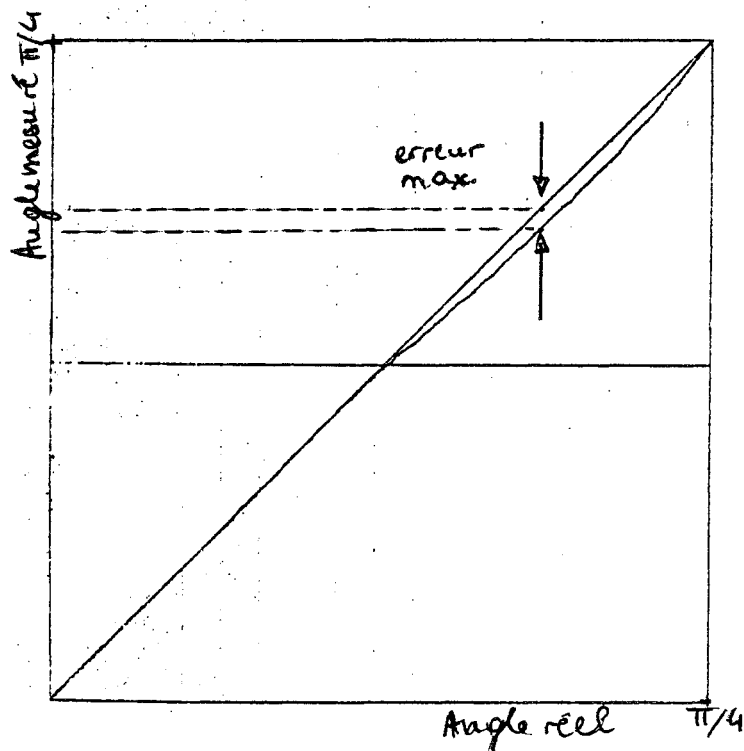


fig. VI.6.

Des résultats analytiques on tire surtout que l'erreur sur la direction est nulle dans le 1° cas; le tracé de la courbe (cf. fig. VI.2.) donnant l'angle mesuré par application de l'opérateur de Sobel permettent de juger de l'anisotropie de l'opérateur de Sobel. L'erreur maximale entre angle du bord idéal et angle mesuré est de l'ordre de 2°.

Nous retrouvons donc ici une partie des résultats publiés par /ABDOU et PRATT 79/ qui ont réalisé ce calcul pour les autres masques 2x2 et 3x3 couramment utilisés; ils ont également calculé la sensibilité de ces détecteurs aux décalages -non pris en compte ici- du bord idéal par rapport au centre du masque.

Si nous avons développé le calcul ici, c'est que le résultat théorique que nous avons obtenu ainsi, nous semblait en contradiction avec l'expérience que nous avions du calcul de bord sur des photos de bâtiments.

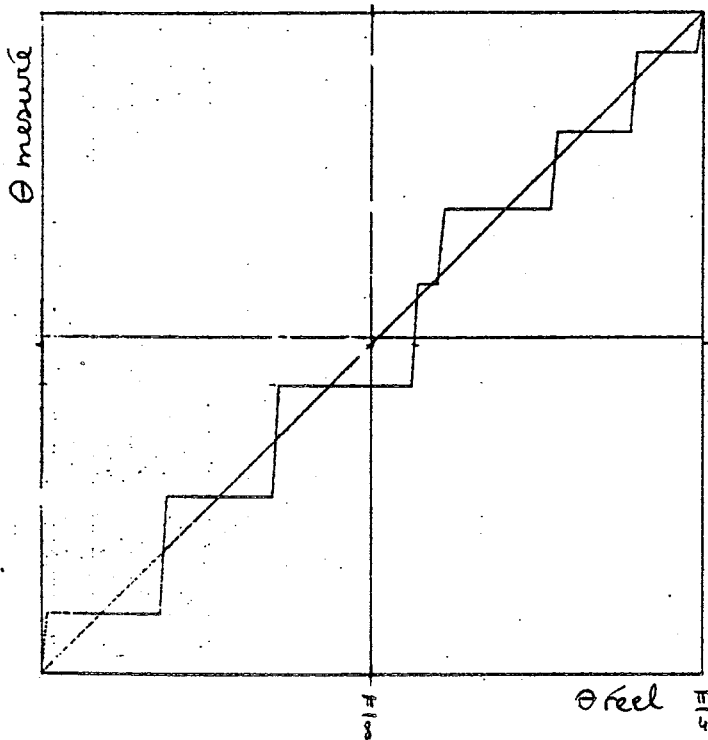
En effet nous allons voir que dans la technique de détection de segments selon la transformation de Hough que nous avons adoptée, apparaît un calcul de l'histogramme de la distribution de la direction des bords calculés sur toute l'image. Or l'histogramme que nous obtenions, présentait un bruit inexplicable par le calcul précédent ou par la nature des photos traitées, bruit qui s'avérait gênant pour la suite de la procédure de détection de segments.

Nous nous sommes donc demandé pourquoi l'étude théorique et les expériences réalisées ne concordaient pas; c'est à dire en quoi, la modélisation de bord idéal ne convenait pas.

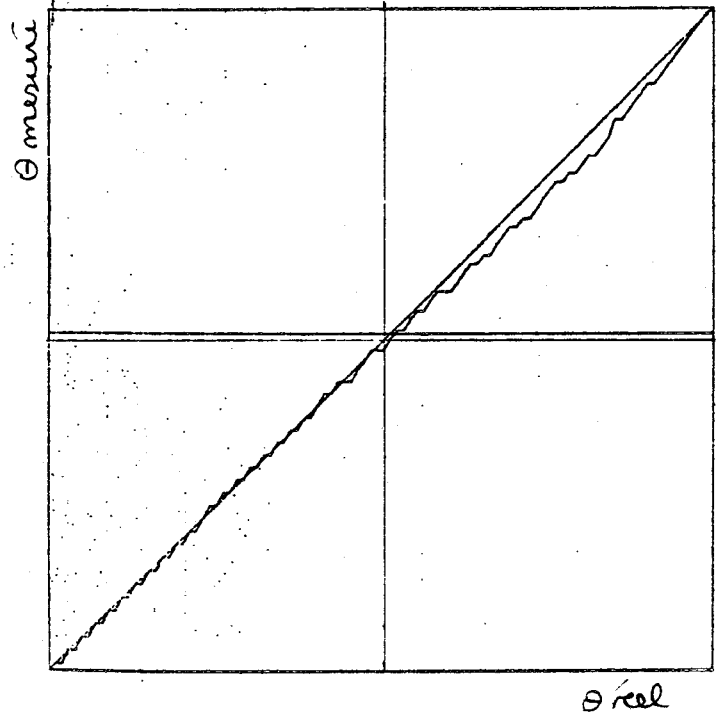
L'explication que nous avons trouvée est que le calcul théorique précédent est réalisé **algébriquement** alors que les coefficients des matrices effectivement traitées sont **quantifiés sur n niveaux**.

Restait à voir si l'erreur de quantification était effectivement la source du bruit sur la détection des directions. Nous l'avons établi expérimentalement en calculant la direction détectée par l'opérateur de Sobel, non pas sur un bord idéal représenté par 3x3 réels, mais par la matrice obtenue par arrondi des coefficients réels aux NG niveaux de quantification disponibles sur l'échelle des valeurs de gris.

Le tracé des courbes ainsi obtenues pour diverses valeurs de NG, démontre clairement l'influence de la quantification sur la distribution des directions.



(a) original sur 8 niveaux.



(b) original sur 64 niveaux.

fig. VI.7. : influence du nombre de niveaux de quantification sur le biais de détection des directions par application de l'opérateur de Sobel.

Le résultat pratique est que pour une image N et B numérisée sur 64 niveaux (6 bits) -ce qui est le cas pour la mémoire d'images utilisée ici-, et bien que visuellement la quantification n'apparaisse pas et que le rendu soit très satisfaisant, la détection de bords par masque de Sobel favorise certaines directions au détriment des directions voisines.

Il était donc illusoire d'accorder une grande confiance aux pics apparaissant dans l'histogramme, et nous avons donc dû effectuer le calcul de gradient à partir de l'image mémorisée sur disque sur 256 niveaux.

VI.3. Détection de segments par transformation de Hough.

VI.3.1. Transformation paramétrique d'une droite du plan.

La méthode d'extraction de segments dans une image que nous décrivons ici, repose sur une transformation qui applique l'ensemble des droites du plan sur un domaine de variation de paramètres de ces droites (transformation de Hough /ROSENFELD 69/).

/DUDA et HART 72/ rejettent la paramétrisation tout d'abord proposée (pente de la droite et intersection avec l'axe des ordonnées), et introduisent la paramétrisation généralement adoptée depuis :

H :

$$(D): \text{d'équation } (x \cdot \cos \theta_d + y \cdot \sin \theta_d = \rho_{od}) \longrightarrow H(D) = (\theta_d, \rho_{od})$$

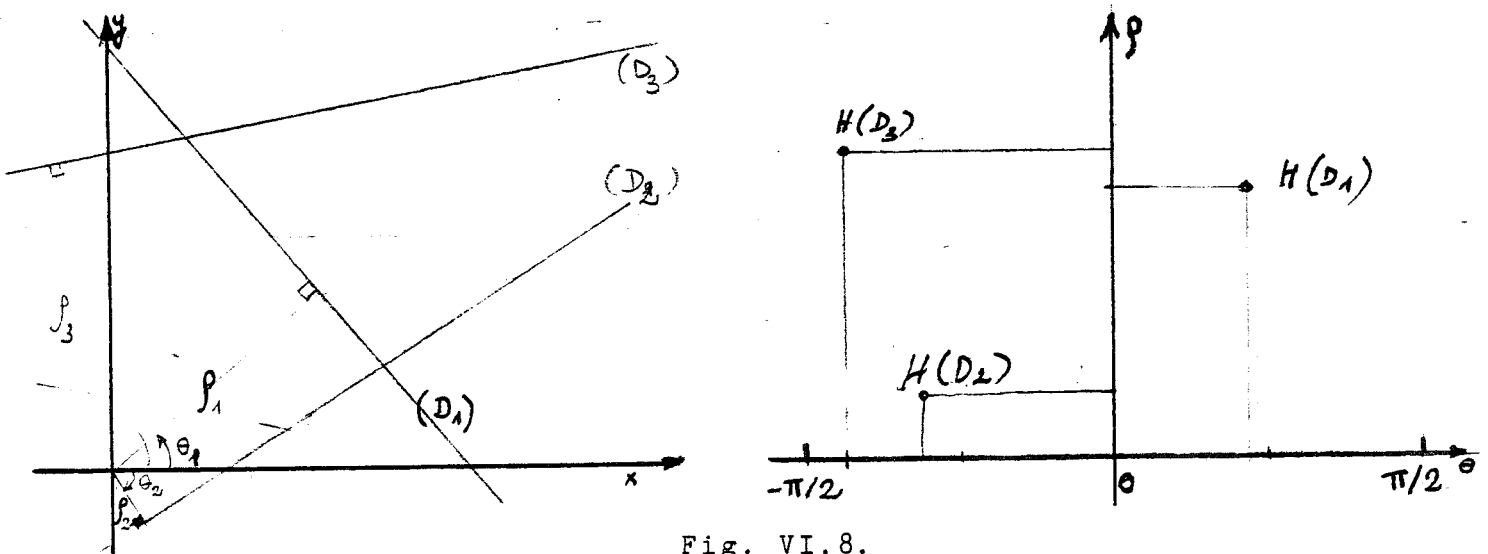


Fig. VI.8.

Les auteurs étendent cette transformation en faisant correspondre à un point du plan xOy, l'ensemble L des points du domaine Rho, Téta qui sont l'image d'une droite passant par P.

$$H : P:(x_p, y_p) \longrightarrow H(P) = \bigcup_{-\pi/2 \leq \theta \leq \pi/2} (\theta, \rho(\theta))$$

$$H(P) = L \text{ où } L \text{ a pour équation, } \rho(\theta) = x_p \cdot \cos \theta + y_p \cdot \sin \theta$$

VI.3.2. Détection de lignes droites dans une figure.

La transformation paramétrique permet de ramener la recherche d'une droite dans le plan, à la recherche d'un point dans le domaine transformé.

Le principe de la méthode consiste à quantifier le domaine transformé et à l'utiliser comme un tableau accumulateur initialisé partout à 0. Pour chaque point noir P de la figure, on incrémente alors les points de l'accumulateur correspondant à la ligne transformée $L=H(P)$.

Les points qui sont les transformés des droites support de segments de la figure ayant été incrémentés le plus souvent, ils peuvent être extraits comme maxima locaux dans l'accumulateur après seuillage de celui-ci.

VI.3.3. Transformée de Hough d'une image de contours.

Du procédé précédent, /DUDA et HART 72/ déduisent directement une méthode de détection des portions de contours d'une image. Elle consiste à agir comme indiqué ci-dessus sur la "figure" obtenue par détection de bords sur l'original, et seuillage de l'amplitude de bord mesurée en chaque point.

Cette méthode a l'inconvénient d'utiliser incomplètement l'information extraite par détection de bord et entraîne pour chaque point du contour, autant d'accès au tableau accumulateur que l'on désire différentier de directions.

/O'GORMAN et CLOWES 76/ proposent au contraire d'utiliser la direction détectée lors du calcul de bord, pour limiter le nombre des accès à l'accumulateur à un par point de l'image où le module du gradient est supérieur au seuil.

Ces auteurs proposent également de remplacer l'incrémentation unitaire par l'incrémentation par le module de l'élément de bord. Mais l'on peut craindre que cette mesure ne facilite la seule

détection des segments les plus prononcés, au détriment de longs segments peu prononcés que seul le nombre des éléments de bord qui les forme peut permettre d'identifier.

Enfin, la direction d'un élément de bord étant comprise entre 0 et 2π , alors que la direction d'une droite est entre 0 et π , ils doublent l'étendue du domaine transformé D.

$$D =]-\pi, +\pi] \times [-\text{Rhomax}, +\text{Rhomax}]$$

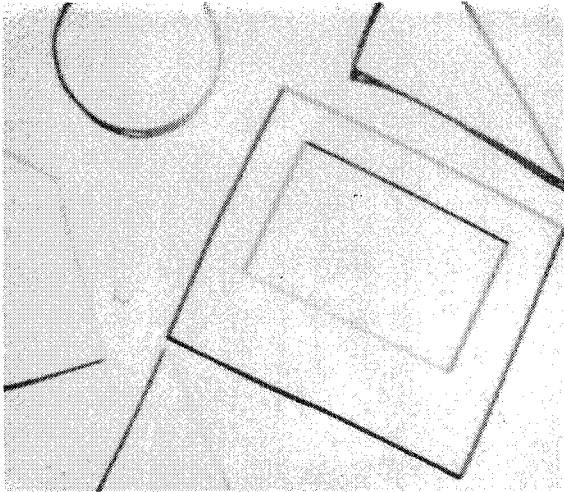
où Rhomax est la longueur de la demi-diagonale de la fenêtre d'image.

Nous présentons sur la figure VI.9., le résultat obtenu par transformation de Hough appliquée à une image test (non représentée) constituée de figures découpées dans du papier de diverses couleurs et ramenées sur un fond clair.

Sur la fig. VI.9.a. apparaissent en blanc, les pixels où la détection de bords donne un résultat de module inférieur à un seuil qu'il est très facile de fixer pour cette image test. Aux points où une transition d'amplitude supérieure au seuil a été détectée, le niveau de gris code la direction du gradient évalué en ce point.

Sur la fig. VI.9.b. est représentée la partie du plan des paramètres (θ, ρ) sur laquelle se transforment les points de la fig. a. Le tracé ainsi obtenu, s'interprète comme suit :

- les regroupements de points en nuages bien individualisés correspondent à l'image par H des éléments de bord colinéaires;
- l'alignement vertical de plusieurs nuages correspond à la détection de segments parallèles;
- enfin, les points qui semblent disposés sur un arc s'étendant sur une période d'une sinusoïde, sont l'image du cercle en haut à gauche de l'image (le cercle est tronqué à la visualisation, mais complet en mémoire d'image). Nous obtiendrions une sinusoïde exacte pour un cercle de rayon très petit. Plus le diamètre du cercle augmente, plus sa courbe transformée est dissymétrique.



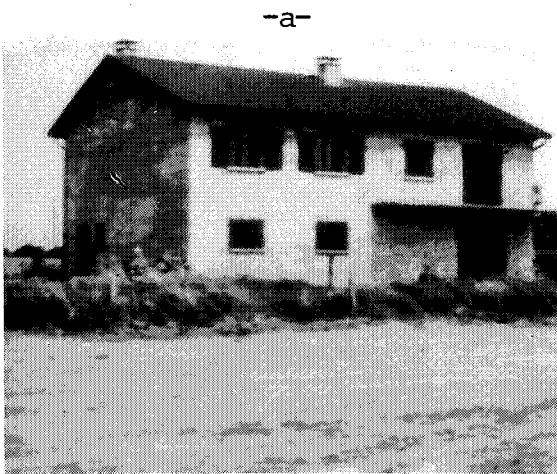
-a-



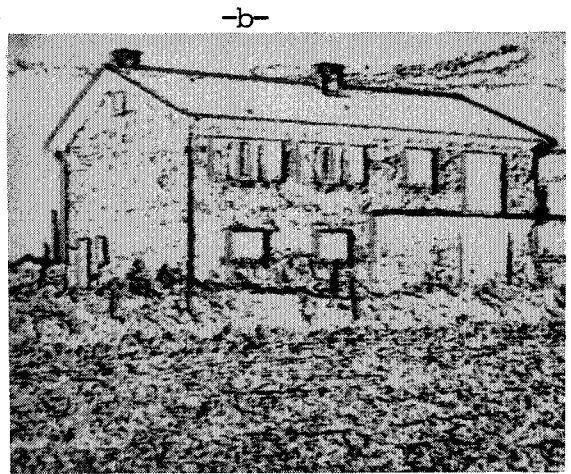
-b-

Figure VI.9.

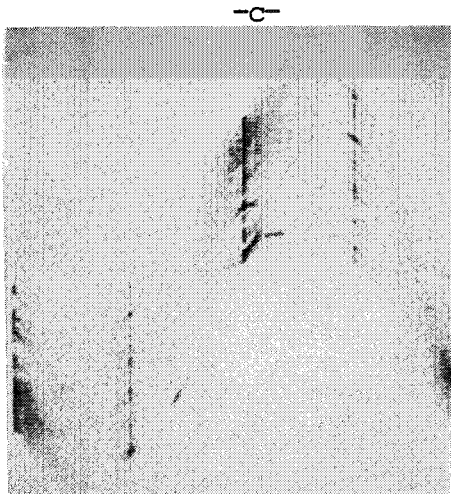
Figure VI.10.



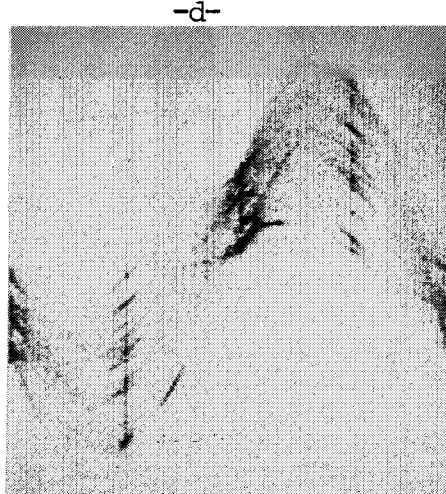
-a-



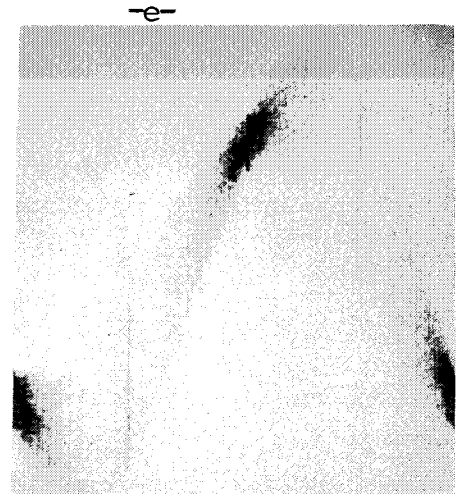
-b-



-c-



-d-



-e-

La figure suivante (VI.10.) présente l'application de la même procédure à une photographie réelle de bâtiments (a).

Le résultat du calcul de gradient est présenté en (b) en utilisant le même codage des directions que ci-dessus.

Le choix du seuil à fixer sur le module du gradient est beaucoup plus délicat que précédemment; en effet, la texture de l'image de l'herbe, fait apparaître des bords désordonnés de modules très souvent supérieurs aux modules des éléments de bord qui correspondent à la limite inférieure du toit. Cependant, la transformation de Hough est justement supposée profiter de la colinéarité de ces derniers, pour les isoler.

Nous donnons ensuite "l'image transformée de Hough", de l'image entière (c), de la moitié supérieure de l'image (d) et enfin de la moitié inférieure (e).

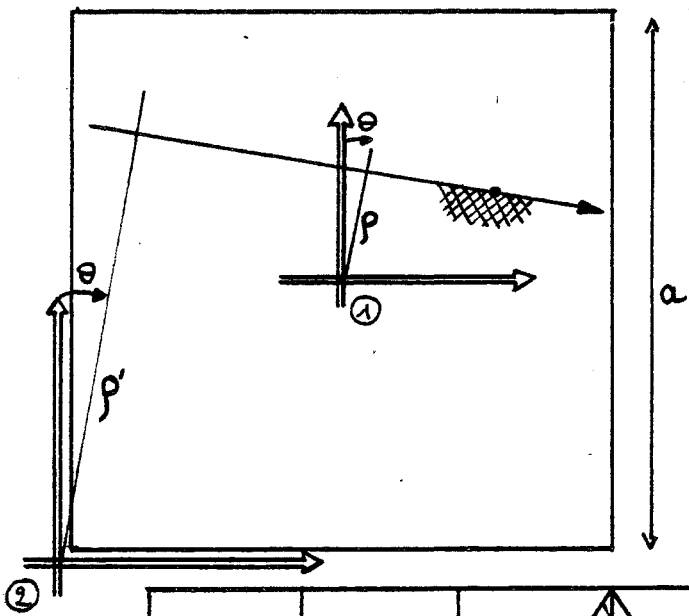
Ainsi apparaissent deux phénomènes préjudiciables à l'extraction de segments :

1) présence sur les fig. c, d et e, de raies verticales marquées qui traduisent l'anisotropie du détecteur de direction (cf plus haut); la gravité de cet effet apparaîtra dès le paragraphe suivant;

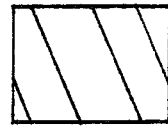
2) formation sur la fig. e (transformée de la texture d'herbe), d'importantes accumulations, alors que l'on peut supposer que grossièrement toutes les directions sont autant représentées. La formation de ces agrégats n'est pas tant due à des directions plus fréquentes, mais à la géométrie de la fenêtre d'image qui a été transformée : en effet celle-ci étant nettement plus large que haute, les éléments de bords verticaux sont dispersés sur un plus grand intervalle du paramètre ρ (amplitude de l'intervalle proportionnel à la largeur du rectangle), que les bords horizontaux.

Quand à l'allure générale de la figure, c'est à dire à la forme du domaine transformé, on montre sur la figure VI.11. l'influence qu'a sur elle, une simple translation du repère de l'image.

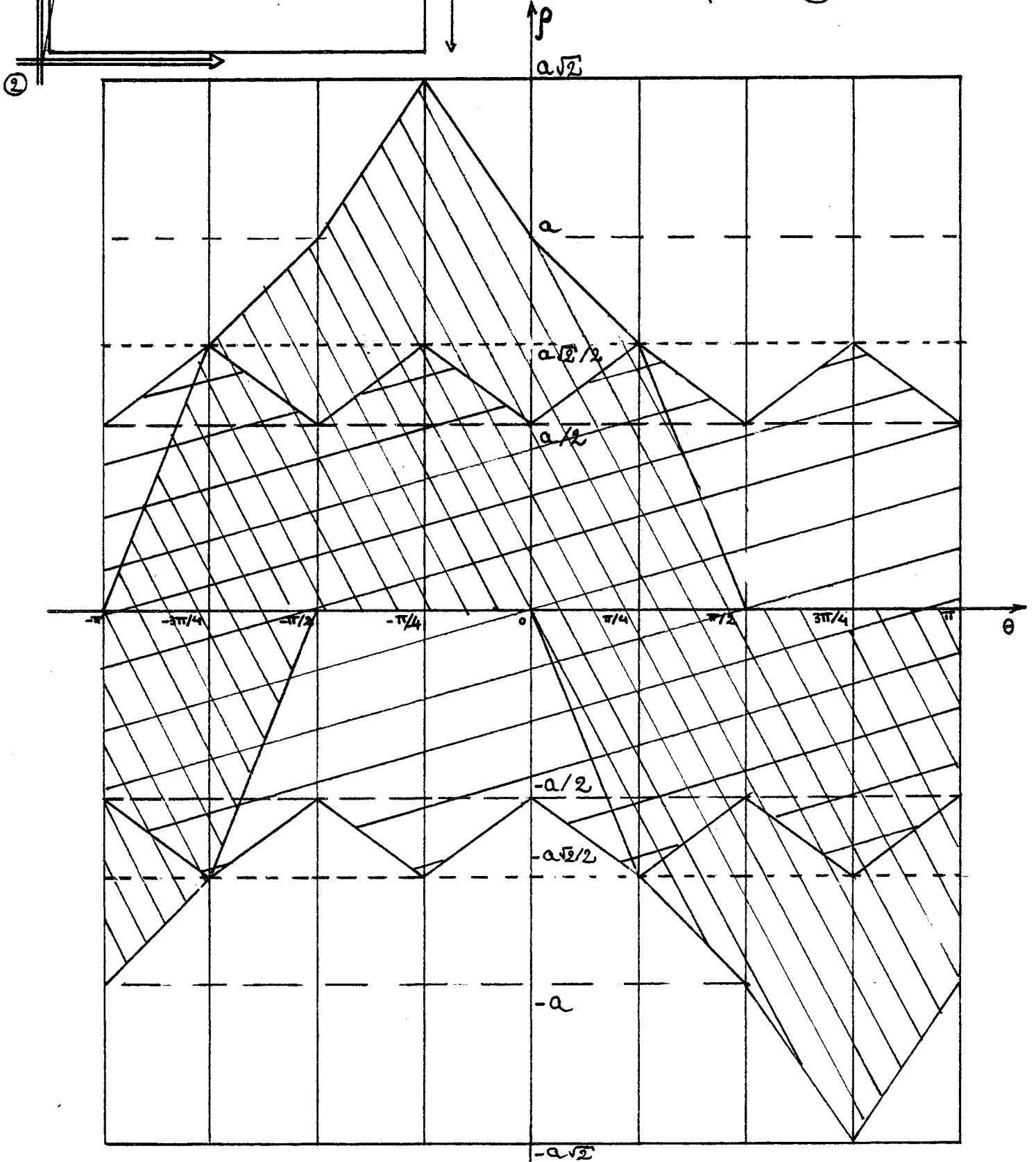
- Figure VI. 11 -



Domaine transformé de l'image relativement au repère (1).



Domaine transformé de l'image relativement au repère (2).



VI.3.4. Recherche des Maxima sur "l'image Transformée de Hough".

Les points (O, Rho) auxquels l'accumulateur prend des valeurs maximales localement, permettent d'identifier les axes qui traversent l'image et le long desquels sont disposés les longs segments de bord.

La recherche de ces maxima peut être réalisée de 2 manières :

- balayage du plan (O, Rho) et comparaison en chaque point de la valeur de l'accumulateur aux valeurs voisines;

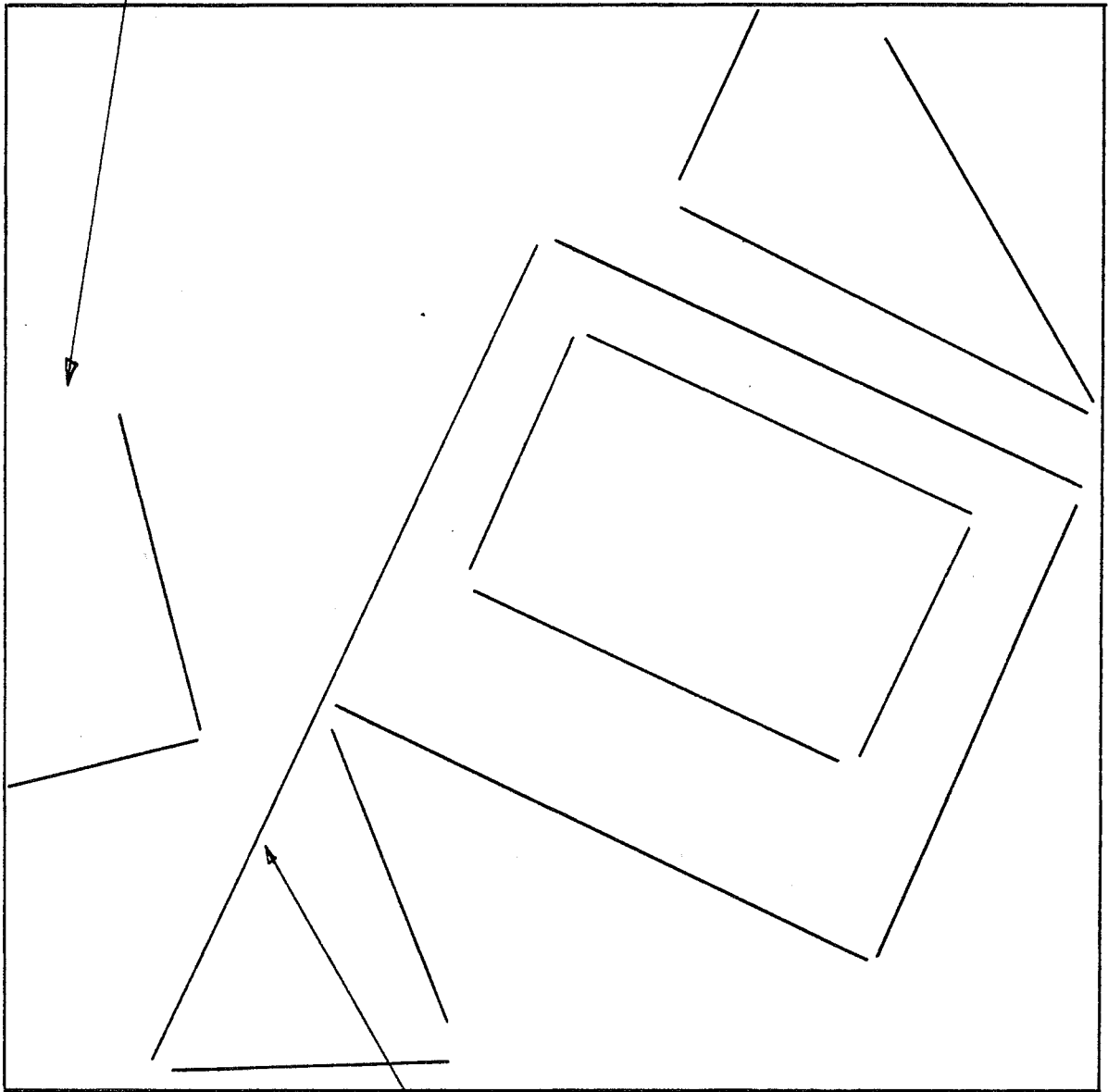
- constitution dans un 1^o temps d'un histogramme des valeurs de 0 pour tous les points de l'image; extraction des pics de cet histogramme et analyse indépendante de chacun des ensemble de pixels contribuant à un pic; cette 2^e étape consiste à nouveau à calculer un histogramme, celui de la distribution des valeurs du 2^o paramètre, Rho .

L'application de ce 2^o procédé à l'image test donne le résultat convenable, comme on le voit sur la figure 12.

Par contre, nous avons en vain tenté de mettre en oeuvre cette technique pourtant présentée par /DUDANI et LUK 78/ comme efficace. La raison principale de cet échec est évidemment l'anisotropie du détecteur de bord qui provoque justement des pics nets et indus dans l'histogramme des directions.

On peut se demander toutefois si l'idée de séparer les directions préalablement à l'analyse du paramètre Rho , est aussi judicieuse qu'il nous est paru au premier abord : en effet un pic en 0 peut être produit par plusieurs axes de directions proches mais non rigoureusement identiques; l'évaluation de la direction est alors trop grossière car l'imprécision est propagée et amplifiée dans la 2^o étape puisque le calcul de Rho fait intervenir 0.

un court segment n'a pas été détecté.



1 seul segment a été extrait
au lieu de 2 segments alignés.

Figure VI.12.

VI.4. Un modèle hiérarchique d'images de bâtiments.

Quelle soit la méthode d'extraction de segments utilisée, on ne doit pas s'attendre à ce que les segments soient extraits avec une fiabilité telle qu'il suffirait alors d'extraire l'enveloppe pour avoir le contour grossier du bâtiment.

Pourquoi ne peut-on s'attendre à cela?

1) En raison de l'imperfection des méthodes utilisées d'une part : lors de la détection de bords, la technique de Hueckel serait meilleure que celle du détecteur de Sobel, mais ne suffirait pas, il faudrait disposer d'un détecteur de frontières de textures /DAVIS 75/; pour l'extraction de segments, nous avons insisté sur les problèmes posés par la transformation de Hough (pouvoir séparateur en θ et ρ , variable selon les régions d'image).

2) Même si l'on disposait d'un extracteur de segments parfait, il y aurait des imperfections dans la description de la scène en segments; imperfections produites -par l'occlusion de certains segments par d'autres objets de la scène; -par l'apparition de segments rectilignes de bords qui ne sont pas l'image d'une arête de la maison.

La question à laquelle nous aurions voulu répondre, mais que les difficultés surgies lors de l'extraction de segments nous ont empêché d'aborder, est la suivante :

Un modèle du type de ceux utilisés sur des images de scènes de blocs, permet-il ou non, à partir des segments extraits de l'image, de déduire la position des segments qui manquent et de reconstituer ainsi le contour grossier du bâtiment à détourer?

Bibliographie du chapitre VI.

Abdou et Pratt

Quantitative Design and Evaluation of Enhancement/Thresholding
Edge Detectors.

Proc IEEE, Vol 67-5, 1979.

D. Ballard et Brown

Computer Vision

Prentice-Hall, 1982.

L. Davis

A Survey of Edge Detection Techniques.

CGIP 4, 1975.

Duda et Hart

Use of the Hough Transformation to Detect Lines and Curves in
Pictures.

Comm. ACM, vol 15-1, 1972.

Dudani et Luk

Locating Straight-Line Edge Segments on Outdoor Scenes.

Pattern Recognition, vol 10, 1978.

A. Guzman

Decomposition of Usual Scenes into 3-d Bodies.

Fall Joint Conference, 1968.

Hueckel

a) An Operator which Locates Edges in Digitalised Pictures.
JACM vol 18-1, 1971.

b) A Local Visual Operator which Recognizes Edges and Lines.
JACM vol 20-4, 1973. (Errata in JACM 74).

T. Kanade

in Computer Vision, M. Brady éditeur, North Holland, 1981.

Nevatia et Babu

Linear Feature Extraction and Description.

CGIP 13, 1980.

O'Gorman et Clowes

Finding Picture Edges Through Collinearity of Feature Points.

IEEE Trans. Comp. vol C 25-4, 1976.

U. Ramer

Extraction of Line Structures from Photographs of Curved
Objects.

CGIP 4, 1975.

A. Rosenfeld et Kak

Digital Image Processing.

Academic Press, 1976.

Schachter *et al*

An Application of Relaxation Methods to Edge Reinforcement.

IEEE Trans. on Systems, Man and Cybernetics, SMC 7-11, 1976.

Zucker, Hummel and Rosenfeld

An Application of Relaxation Labeling to Line and Curve
Enhancement.

IEEE trans. on Computers, vol C 26-4, 1977.

CHAPITRE 7

=====

Chapitre VII :

Vers un Système Intégré de Détourage.

VII.1. Les Entités du Système.

VII.2. Interdépendance et Complémentarité des Acteurs.

VII.3. Les Actions du Programme et leur Enchaînement.

VII.4. Conclusion.

CHAPITRE VII :

**CONCLUSION : VERS UNE SYSTEME INTEGRE
DE DETOURAGE.**

L'insuffisance des résultats expérimentaux qui devrait illustrer l'intérêt de la modélisation dans l'assistance au détournage, ne doit pas nous empêcher d'esquisser brièvement, en guise de conclusion, les grands traits d'un système de détournage interactif tel qu'il nous semble devoir s'organiser.

Les principales entités mises en jeu dans le système peuvent être classées comme suit :

a. les acteurs :

- . le programme
- . l'opérateur

b. les données :

- . l'image originale
- . le modèle du monde des objets à détourer
- . le contour polygonal
- . les éléments du modèle identifiés
- . les hypothèses formulées par le programme
- . les indications de l'opérateur

c. les actions du programme :

- . l'extraction automatique des primitives ("feature extraction")
- . le détournage ultime d'après le contour polygonal grossier

- . la "reconnaissance" des éléments du modèle
- . les recours à l'opérateur
- . la "dérivation"

d. les actions de l'opérateur :

- . analyse et interprétation de l'image
- . introduction de la consigne
- . Intervention,
 - pour introduire des éléments non identifiés
 - pour corriger des hypothèses erronées du programme

L'interdépendance de ces entités est illustré sur la figure ci-dessous :

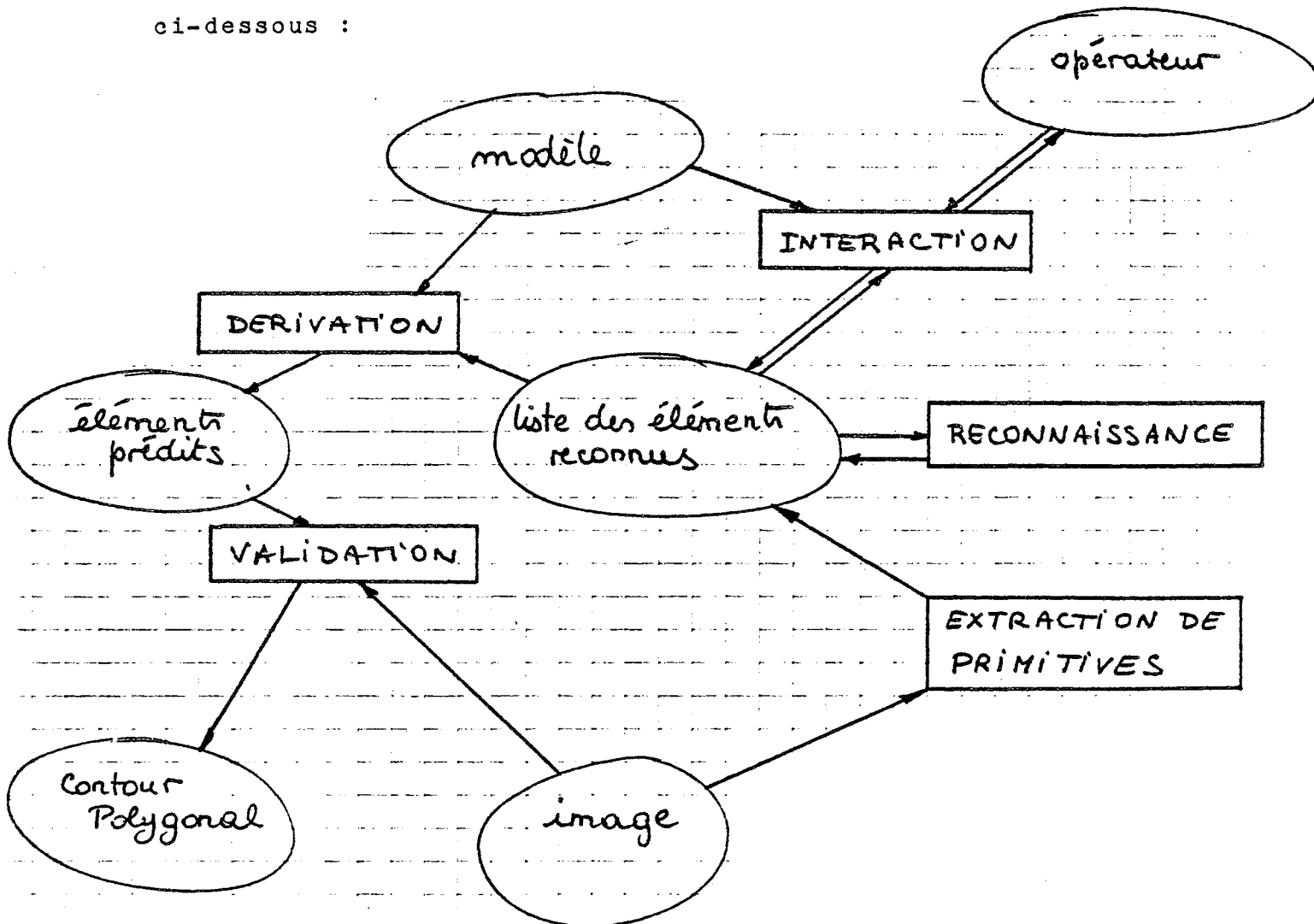


Figure VII.1.

Les deux acteurs du système ont des performances très dissemblables :

- la capacité de l'opérateur d'interpréter instantanément la très grande majorité des images, s'oppose à la très faible efficacité prévisible du programme sur ce point.

- à l'inverse, nous avons démontré dans les chapitres II et III que l'on peut envisager la mise au point d'un programme de découpe finale à l'échelle du pixel, plus rapide que l'opérateur, et plus "adroit" que celui-ci dans les contours nets mais très tourmentés.

Sur l'exploitation de cette dissymétrie repose le gain apporté par le détournage assisté.

Ceci a une 1^o conséquence pratique très importante dans la programmation de l'interaction : celle-ci doit être largement asynchrone; c'est-à-dire que le programme ne doit pas être le seul à pouvoir prendre l'initiative des échanges, et c'est pourquoi nous avons distingué 2 cas d'introduction d'informations par l'opérateur selon l'initiateur de l'interaction :

- recours à l'opérateur :

le programme ayant des difficultés à avancer dans son analyse, il interroge l'opérateur sur un point crucial; c'est à dire un point sur lequel une information sera la plus profitable;

- intervention de l'opérateur :

celui-ci doit pouvoir à tout instant, c'est-à-dire même si le programme n'a pas jugé utile de le questionner, apporter une information dont lui, l'opérateur, se rend compte qu'elle simplifiera la tâche de la machine; par exemple :

. indication qu'une hypothèse non encore apparue fausse au programme, doit être immédiatement abandonnée;

. introduction d'une primitive non extraite automatiquement et dont la connaissance sera a priori précieuse au programme.

L'écriture de modules asynchrones n'est pas directement possible lorsque l'on utilise les langages de programmation de haut niveau les plus courants aujourd'hui (Fortran IV, Pascal...); mais d'une part de nouveaux langages (Concurrent Pascal, Modula II, Ada...) sont actuellement en cours de développement, qui devront permettre la programmation aisée de tâches à déroulements parallèles (i.e. dont les déroulements sont simultanés ou au moins entrelacés).

D'autre part, il est d'ores et déjà possible sur certains systèmes d'exploitation, d'émettre depuis un programme écrit en langage de haut niveau, des requêtes au système, pour réaliser la synchronisation entre modules dont l'ordre d'exécution est indifférent.

Les actions du programme.

Le travail à réaliser par le programme se décompose en deux étapes bien distinctes :

- la 1° consiste à déterminer le contour polygonal grossier de l'objet à détourner;
- la 2° étape est celle de la segmentation ultime guidée par le contour grossier.

Ces 2 étapes se succèdent nécessairement dans le temps : la 2° ne peut démarrer avant que la 1° ne se soit terminée, et n'agit sur cette dernière par aucune sorte de rétro-action.

Le résultat de la 1° étape, par contre est réalisé par l'exécution de nombreuses tâches plus élémentaires.

Les tâches qui concourent à l'élaboration du contour guide, sont les suivantes :

a. l'extraction des primitives. Les caractéristiques de cette tâche dépendent évidemment dans une très large mesure, des primitives choisies (zones homogènes, segments de bords...).

b. reconnaissance d'éléments non-terminaux du modèle. Il s'agit là de tenter la recombinaison d'éléments extraits ou reconnus préalablement, pour identifier dans leur arrangement des éléments de niveau supérieur. La nature combinatoire de cette procédure de reconnaissance est évidente.

c. ce que nous avons appelé plus haut "dérivation", est la procédure inverse de la précédente, qui consiste à déduire d'un élément non terminal reconnu, les éléments qui le constituent. Nous avons expérimenté une procédure de ce type sur le modèle du pantin (cf. IV. .); d'une façon plus générale, et pour un modèle donné, la constitution de ces procédures ne devrait pas poser de difficultés majeures; elles sont la conséquence directe de la modélisation adoptée.

d. le recours à l'opérateur, pour lui demander d'introduire un élément, se présente pour le système de la même façon que l'extraction d'une primitive ou la reconnaissance d'un élément du modèle.

l'enchaînement des actions du programme.

L'enchaînement de ces multiples actions peut-être établi de manière statique, lors de la programmation. Plusieurs choix sont alors possibles :

a. progression par niveaux :

. Commencer par exécuter l'extraction de primitives jusqu'à son terme.

. Tenter toutes les combinaisons de primitives pour identifier tous les éléments de niveau immédiatement supérieur.

. passer à la reconnaissance des éléments de niveau encore supérieur, et ainsi de suite...

. ... jusqu'à la reconnaissance globale ou à son échec et dans ce dernier cas, recourir à l'opérateur.

b. de façon duale, on peut :

. entamer l'extraction de primitives et la suspendre dès que les résultats permettent de ...

. ... commencer la remontée dans la structure hiérarchique.

. ... et tenter la reconnaissance d'élément d'un niveau dès que quelques éléments du niveau précédent sont disponibles.

Le choix entre ces solutions extrêmes et une solution de compromis, toujours possible, dépend de manière générale du cas précis que l'on est en train de traiter. Aussi est-il souhaitable que l'enchaînement des tâches ne soit pas statique, mais qu'il varie d'un cas en fonction des résultats obtenus tout au long du déroulement de l'analyse.

La solution que nous proposons à cette nécessité, et que nous avons expérimentée sur une simulation partielle de l'analyse de bâtiments, consiste à réaliser un programme qui gère lui-même l'enchaînement des actions qu'il est capable d'exécuter.

Pour cela, il est nécessaire que le programme dispose d'une structure de données lui décrivant pour chacune de ses actions, les éléments actuellement reconnus que cette action pourrait analyser (éléments "en entrée pour cette action"), le type des éléments que cette action pourrait produire et leur importance au point actuellement atteint par l'analyse.

Grâce à ces informations, le programme peut calculer à chaque fois que l'exécution d'une de ses actions se termine, la priorité à accorder à toutes celles-ci, et provoquer ainsi l'action de la plus efficace a priori.

En l'absence d'une expérimentation approfondie de ce mécanisme d'analyse, nous n'entrerons pas plus avant dans les détails d'une description qui ne saurait être qu'incomplète.

C'est ainsi que nous concluerons l'étude que nous avons consacré au détournage interactif de photos d'intérêt général. De cette étude nous retiendrons surtout ici, le très large éventail de techniques auxquelles elle nous a renvoyé, qu'il s'agisse des techniques d'analyse d'images et en particulier de segmentation; d'analyse de la complexité des algorithmes ; de modélisation ...

Pratiquement, la principale leçon que nous pensons pouvoir tirer de ce travail, est qu'il est d'ores et déjà possible de réaliser une application de taille et de complexité suffisamment réduites pour permettre de construire avec un système de traitement de texte et une palette électronique, un poste de travail de composition de maquettes pour l'imprimerie.

Quand aux techniques plus complexes que nous avons vues, et à leur introduction dans une telle station de fabrication de documents, la question de leur efficacité reste ouverte.

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 avril 1974,

VU les rapports de présentation de M. COUEIGNOUX et M. HABIB

M. Marc BERTIER

est autorisé à présenter une thèse en soutenance pour l'obtention
du diplôme de DOCTEUR-INGENIEUR, spécialité Informatique

Fait à Saint-Etienne, le 24 juin 1983

Le Directeur de l'EMSE,

A handwritten signature in black ink, appearing to be 'M. MERMET', written over a horizontal line.

M. MERMET.

