Extraction et analyse d'objets-clés pour la structuration d'images et de vidéos

Jérémy Huart

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

THÈSE

pour obtenir le grade de

DOCTEUR DE L'INP Grenoble

Spécialité : « Signal, Image, Parole et Télécoms »

préparée au laboratoire GIPSA-lab DIS

dans le cadre de **l'École Doctorale**
« Électronique, Électrotechnique, Automatique, Télécommunications et Signal »

présentée et soutenue publiquement

par

Jérémy HUART

le 14 février 2007

EXTRACTION ET ANALYSE D'OBJETS-CLÉS POUR LA STRUCTURATION D'IMAGES ET DE VIDÉOS

Sous la Direction de

Pascal BERTOLINO

JURY

M. James L. CROWLEY, Professeur de l'INPG	Président
M. Dominique BARBA, Professeur émérite de Polytech' Nantes	Rapporteur
M. Henri NICOLAS, Professeur de l'Université Bordeaux 1	Rapporteur
M. Pascal BERTOLINO, MCF à l'UPMF	Directeur
Mme. Anne GUERIN-DUGUE, Professeur de l'UJF	Examinatrice

De celui qui dans la bataille a vaincu mille milliers d'hommes et de celui qui s'est vaincu lui-même, c'est ce dernier qui est le plus grand vainqueur.

Bouddha Shākyamuni

Extraction et analyse d'objets-clés pour la structuration d'images et de vidéos

Résumé

La description synthétique du contenu d'une image ou d'une vidéo est à l'heure actuelle une problématique majeure. Nous nous intéressons aux objets qui les composent pour leur pouvoir de représentativité. Après un état de l'art, ce document présente une méthode de segmentation locale par pyramide de graphes irrégulière permettant d'extraire, à partir de critères bas-niveaux, des régions d'intérêt assimilables à des objets sémantiques. Cette méthode est utilisée pour détourer avec précision des objets dans des images fixes, dans un environnement interactif puis totalement automatique. Une estimation de mouvement permet d'étendre le procédé aux vidéos en extrayant dans chaque image les entités mobiles. Un filtrage et une classification de ces entités permet de ne retenir que les plus représentatives de chaque objet réel du plan. Ces représentants sont appelés objet-clé et vues-clés. La qualité des résultats expérimentaux permet de proposer de nombreuses applications en aval.

Mots clés : Objets clés, attributs inter/intra images, structuration des vidéos, indexation, segmentation locale, pyramide irrégulière, classification d'objets.

Extraction and Analysis of Key-Objects for Image and Video Structuring

Abstract

The compact description of image and video content is currently a difficult task. We are interested in the objects that make up this content because of the representative power of these objects. After a review of the state of the art, this thesis presents a local segmentation method based on the irregular graph pyramid algorithm, which allows us to extract, using low-level features, regions of interest comparable to semantic objects. This method is used to precisely excise objects from still images, first in an interactive environment and then in an entirely automatic one. A motion estimation allows us to extend the process to videos by extracting the foreground entities from every frame. A filtering and a clustering of these entities allow us to retain only the most representative of each real object in the shot. These representations are called key-objects and key-views. The quality of the experimental results allows us to propose some future applications of our methods.

Key words : Key-objects, intra/inter frame features, video structuring, indexing, local segmentation.

GIPSA-Lab (CNRS UMR 5216)
Département Images et Signal (DIS)
961 rue de la Houille Blanche
ENSIEG, Domaine Universitaire, BP 46,
38402 St-Martin-d'Hères Cedex, France

Remerciements

Ce travail a été réalisé au Département Images et Signal de GIPSA Lab, laboratoire de l'image, de la parole, du signal et de l'automatique de Grenoble. En tout premier lieu, je remercie tous les membres du laboratoire qui font de ce lieu un environnement de travail serein et accueillant.

Je souhaite remercier Monsieur Jean-Marc Chassery, Directeur de Recherche CNRS et Directeur de GIPSA Lab de m'avoir accueilli dans le laboratoire.

Je voudrais exprimer mes sincères remerciements à Pascal Bertolino, Maître de Conférences à l'Université Pierre Mendès France, pour ces trois années que j'ai passées sous sa direction. Après avoir été un directeur remarquable, il est devenu un collaborateur avec qui travailler et relever les défis technologiques relatifs à l'édition vidéo s'avère être un réel plaisir. Je lui suis reconnaissant de m'avoir transmis sont goût pour la recherche mais également son intérêt pour le transfert technologique.

Je remercie aussi l'ensemble des membres du jury pour ses remarques constructives et ses qualités humaines qui ont fait de la soutenance un évènement enrichissant et qui restera un très bon souvenir. Je leur suis également très reconnaissant du temps qu'ils ont consacré à l'étude de ce manuscrit.

James L. Crowley, Professeur de l'INPG m'a fait l'honneur de présider mon jury de soutenance. Je le remercie grandement pour l'étude détaillée de mon manuscrit et pour l'habilité dont il a fait preuve pour présider la soutenance.

Dominique Barba, Professeur émérite de l'École Polytechnique de l'Université de Nantes a accepté d'être rapporteur de ma thèse. J'en ai été très honoré et je tiens à lui témoigner mes sincères remerciements pour l'intérêt qu'il a porté à mon travail et les remarques enrichissantes qu'il a émises.

Henri Nicolas, Professeur de Université Bordeaux 1 a également accepté d'être rapporteur de ce manuscrit. Ses domaines de recherche étant proches de ceux évoqués dans la thèse, ce fut très gratifiant pour moi d'obtenir son expertise.

Anne Guerin, Professeur de l'Université Joseph Fourier de Grenoble a accepté d'examiner mon travail. Je la remercie pour ses critiques pertinentes qui me serviront à enrichir mon travail futur.

Je tenais aussi à saluer les ingénieurs du laboratoire, Hervé Colasuonno et Jean-Marc Sache, pour leur travail qui fait de l'environnement informatique du DIS un exemple du genre.

J'aimerais terminer par des remerciements plus personnels. Tout d'abord, je remercie ma famille pour son soutien moral. Ensuite je tiens à dire un grand merci à la « p'tite famille »

qui m'est si chère. Je ne nommerai pas tous ses membres mais ils se reconnaîtront. Et enfin, je tenais à remercier Claire, pour le soutien essentiel qu'elle m'a apporté et pour ses longues relectures qui ont permis de limiter grandement les fautes de français dans ce manuscrit. De plus, elle a su rendre plus agréable cette période de rédaction où la sérénité n'était pas toujours de mise. Elle a su me supporter pendant cette période et j'espère qu'elle saura me supporter encore longtemps.

Table des matières

Table des figures

Introduction

Contexte général

A l'heure actuelle, nous vivons une révolution à tous niveaux dans le domaine de plus en plus vaste de l'image numérique. Cette révolution nous apporte quotidiennement à nous, consommateurs, des services et des produits qui se trouvaient encore dans les laboratoires quelques années ou parfois même quelque mois plus tôt. En voici quelques exemples représentatifs :

- En acquisition, avec des appareils photos et des camescopes numériques capturant des images avec une définition et une qualité toujours plus élevée et ceci dans des boîtiers toujours plus compacts. De plus, les constructeurs se sont lancés dans une véritable course à l'innovation en proposant des fonctionnalités toujours plus originales : construction de panoramiques, détection de visage pour restituer des couleurs les plus naturelles, détection et correction anti-yeux rouges temps réel...

- Dans le domaine de l'affichage : remplacement des écrans à tubes cathodiques par des écrans plats, tout d'abord pour les ordinateurs personnels puis pour les écrans de télévision. Cette évolution n'est pas sans contrainte puisqu'il s'avère que ces "nouvelles" technologies (LCD, plasma, DLP) demandent en entrée un signal vidéo de bonne qualité afin de bénéficier entièrement de leur capacité. Ils privilégient donc les signaux numériques avec un affichage progressif des images, du fait que les signaux analogiques entrelacés posent encore des problèmes.

- Dans le domaine de l'impression : pour une imprimante au coût d'achat souvent faible et des consommables prohibitifs, il est maintenant facile d'imprimer en silence et rapidement ses photos et étiquettes de CD-ROM/DVD, avec une qualité comparable à ce qui était jusque-là réservé aux professionnels dans les laboratoires photo.

- En édition : PhotoShop n'est plus réservé aux professionnels, et si son coût est trop élevé, il existe un nombre non négligeable de concurrents, dont le célèbre Gimp qui est distribué gratuitement sous licence GNU. L'utilisateur trouve également à sa disposition tout un ensemble de logiciels pour réaliser ses propres montages vidéo des séquences qu'il a tournées lui-même.

- Dans le domaine du stockage : le CD-ROM a rapidement laissé la part belle au DVD qui lui-même est en train de céder la place aux deux supports concurrents proposant la haute définition (HD) : le Blue-Ray et le HD-DVD. De nouveaux usages et un goût prononcé pour la mobilité ont entraîné la démocratisation des "clés" USB, des mémoires flash, des disques miniatures (1 pouce). De nouvelles fonctionnalités voient régulièrement le jour dans les objets que nous côtoyons : baladeurs MP3 permettant de visionner ses vidéos

et ses photos, téléviseurs permettant le contrôle du direct (*Time shifting*)...

- Dans le domaine des formats et des normes de compression : le JPEG est le format que tout le monde côtoie et connaît. Le DVD a démocratisé le MPEG2. Les formats dérivés (H264, DivX, MPEG3 layer 2) permettent sans cesse une plus forte compression accompagnée d'une meilleure qualité.

- Dans le domaine de la transmission et des réseaux : ce grand brassage de l'image numérique a débuté pour le consommateur moyen il y a une dizaine d'années, sans doute en partie par le biais d'Internet et des réseaux locaux. Maintenant, on dispose d'Internet haut débit à domicile, téléphonie et télévision sur IP, télévision numérique terrestre, technologie sans fil embarquée dans les appareils photo numériques pour partager ses clichés ... L'évolution des réseaux a également permis l'émergence des plate-formes d'hébergement et de partage d'images ou de vidéos personnelles sur l'Internet : chaque jour, le service *YouTube* diffuse près de 100 millions de vidéos et enrichit sa base de 65000 nouvelles vidéos. Quant à la plate-forme *Flickr*, elle héberge désormais plus de 130 millions de photos postées par 3 millions d'utilisateurs amateurs ou professionnels.

- Enfin, dans le domaine du traitement et de l'analyse : quoi que l'on veuille faire avec des images numériques, on sait maintenant que le traitement sera d'autant plus efficace qu'il prendra en compte le contenu de l'image. Bien souvent, on aimerait avoir un traitement d'aussi bonne qualité que celui que nous offre notre système visuel et notre sens artistique. D'autres fois, on voudrait que le traitement offre à nos yeux la meilleure qualité subjective. Pour ces deux raisons, il est devenu inévitable de manipuler une image non plus comme un signal discret, mais comme un ensemble d'objets sémantiquement représentatifs pour l'Homme.

Objectif

L'objectif de la thèse est de traiter des images et des séquences vidéo pour en fournir une description compacte et représentative de leur contenu. Nous entendons par *contenu d'une image*, les entités appelées *objets* qu'elle peut contenir. Ainsi, à titre d'illustration, l'image de la figure 1.a[1] peut être décomposée ou segmentée en deux parties : l'objet[2] (cf. figure 1.b) et le fond (cf. figure 1.c). Le résultat de la segmentation peut se représenter sous la forme d'un masque binaire définissant d'une part, les pixels de l'image appartenant à l'objet et d'autre part, ceux qui appartiennent au reste de l'image considéré alors, par opposition, comme le fond. Cette segmentation fondée sur les objets s'appelle également *extraction d'objet*. La notion d'objet est détaillée dans le chapitre 1. En effet, c'est une notion complexe nécessitant une définition précise et qui est parfois utilisée de manière approximative dans la littérature.

Rendre une machine capable de « comprendre » le contenu d'une image ou d'une vidéo comme nous le permet notre système visuel n'est pas chose aisée. Il vous est déjà arrivé de reconnaître une personne que vous connaissez au milieu d'une foule alors que vous ne distinguez qu'une silhouette, une démarche ou une allure. Aucune méthode de reconnaissance automatique n'est encore arrivée à ce stade de compréhension. Prenons trois exemples illustrés par des illusions bien connues qui sont certes des cas particuliers mais qui permettent d'avoir un bon aperçu du problème. (1) Certaines caractéristiques d'une région d'une image comme

[1]Photo de *Born Sleepy* disponible sur le site *http* : *//www.flickr.com*. Distribution sous licence *Creative Commons* : Paternité - Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique 2.0

[2]Extraction réalisée de manière semi-automatique avec notre application ExtraK'obs

(a) L'image originale (b) L'objet (c) Le fond

FIG. 1: Étude du contenu d'une image

la taille ou le niveau de gris sont simples à quantifier par une machine mais peuvent pourtant être interprétés d'une manière assez surprenante (mais explicable) par notre système visuel. Ce phénomène est illustré dans les figures 2.a et 2.b. (2) Notre système visuel est adaptatif et utilise le mouvement. De ce fait, une image peut être interprétée d'une manière dynamique comme le montrent les figures 2.c et 2.d. (3) Les innombrables traitements cognitifs modelés par notre culture et notre apprentissage s'ajoutant à notre système visuel, permettent d'interpréter ce que nous voyons et ainsi de donner du sens à cet ensemble de pixels qu'est l'image. Les figures 2.e et 2.f nous montrent des illusions que notre réflexion et notre imagination permettent de bien discerner. Une reconnaissance d'objet automatisée est facilement piégée par ce type d'illusions.

Ces quelques exemples ne sont qu'un infime aperçu de l'extrême complexité que constitue la compréhension des images par la machine : ce qui est simple pour nous l'est beaucoup moins, voire pas du tout pour une machine. L'extraction d'objet n'est bien sûr qu'une étape menant à la compréhension des images et des vidéos par la machine mais constitue déjà un véritable défi technologique.

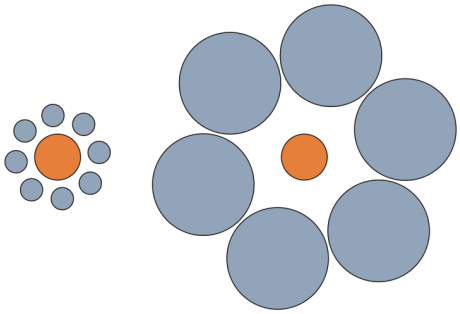
Ce manuscrit se doit donc de rester humble devant le système visuel dont nous a doté la nature. Son objectif est de traiter des procédés d'extraction d'objets dans les images et les séquences vidéo qui tentent de reproduire avec toute l'humilité qu'il se doit, une infime partie de ce que notre système visuel nous permet de faire de manière presque parfaite et ce sans pratiquement aucun effort. En plus de proposer des méthodes d'extraction, nos travaux portent sur une utilisation originale des régions extraites automatiquement au cours d'une vidéo en considérant les limites inhérentes à de tels traitements pour finalement, proposer une représentation fondée sur les objets de la séquence.

Structure du document

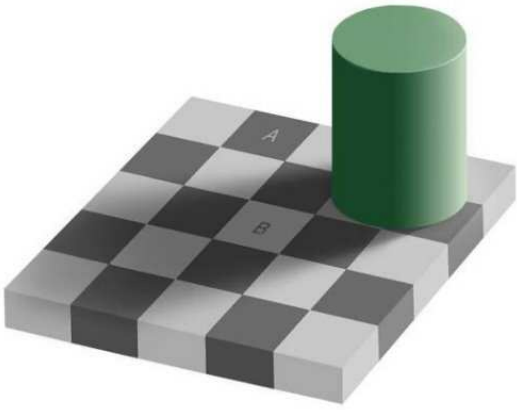
Après un premier chapitre présentant des définitions délimitant le cadre de l'étude et les applications liées au sujet, ce document aborde deux problématiques principales qui sont les suivantes :

L'extraction d'objets dans les images et les vidéos

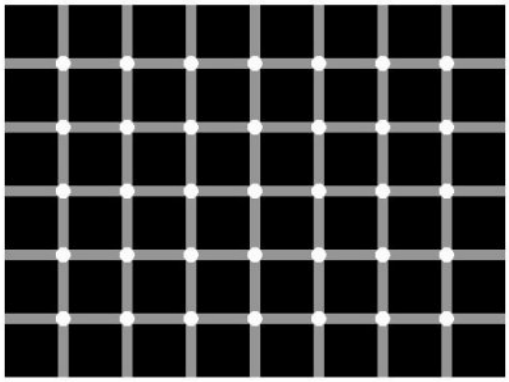
De nombreuses applications ont comme besoin générique d'extraire les objets dans les images et les vidéos. Par exemple, en ce qui concerne les images, le fait de pouvoir les décomposer en objets permet d'améliorer les moteurs de recherche dans les grandes bases de données en focalisant la recherche sur le contenu de l'image et non pas sur ses caractéristiques globales. Les méthodes actuelles orientées sur le contenu nécessitent encore une forte inter-



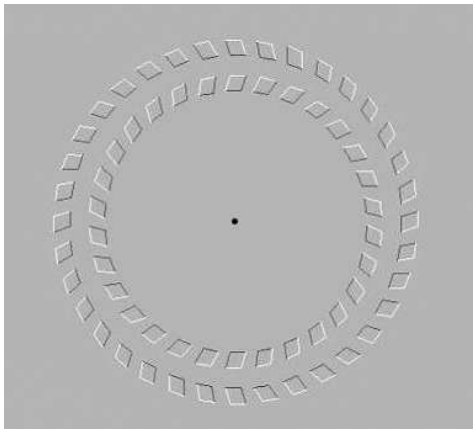
(a) Illusion de Titchener : Les deux cercles intérieurs sont de la même taille



(b) Echiquier d'Adelson : Le niveau de gris du carré A est le même que celui du carré B



(c) Comptez les points noirs...



(d) Approchez-vous...

(e) Le plus plat des portables

(f) Tout le monde a un sosie

FIG. 2: Quelques illusions

vention manuelle. Dans le domaine de la vidéo, l'extraction d'objets permet non seulement un suivi des objets mais aussi et surtout une localisation précise de leurs limites ou contours tout au long de la séquence. Illustrons ce principe avec deux exemples diamétralement opposés : la vidéo-surveillance avec ses contraintes de temps réel et d'autonomie, et la publicité interactive pour laquelle des traitements lourds, en différé et supervisés sont envisageables. C'est ce type d'applications qui va davantage nous intéresser. Cette forme de publicité n'en est qu'à ses balbutiements mais devrait bientôt générer un marché énorme, essentiellement basé sur tous les produits que le spectateur peut voir pendant la diffusion d'un film (habillement, ameublement, tourisme, services, ...). Actuellement, le travail de détourage et d'indexation des objets est bien souvent réalisé manuellement ou semi-automatiquement à l'aide de produits du marché : Movideo de Arts Video Interactive, Smox Editor de Manalee, PatchMaker de Pixmart. Le passage à une grande échelle de ce nouveau marché ne peut être réalisé qu'avec des outils de production efficaces, permettant de réduire l'intervention manuelle.

En conséquence, la première problématique abordée dans les chapitres 2 et 3 concerne la représentation fondée sur les objets des images et des vidéos. Le premier chapitre présente un état de l'art tandis que le deuxième chapitre présente nos méthodes. L'étude se porte tout d'abord sur des procédés interactifs puis s'étend aux méthodes automatiques spatiales et spatio-temporelles. Les traitements interactifs permettent une bonne décomposition des images en objets. Nous verrons que dans le cas de méthodes automatiques, il est nécessaire d'être plus prudent sur l'utilisation du terme "objet" et on lui préfèrera le terme de *région d'intérêt*.

La méthode de segmentation utilisée est *la pyramide irrégulière* [90]. Il s'agit d'une technique de segmentation par croissance en parallèle de régions. Les travaux antérieurs ont permis dans un premier temps d'effectuer de la segmentation multirésolution d'images fixes [6] puis d'étendre la technique au suivi d'objet initialisé manuellement développé par G. Foret [40]. L'objectif de nos travaux a été d'étendre l'utilisation de la pyramide irrégulière à l'extraction d'objet dans les images puis dans les vidéos. Nos travaux se sont donc orientés aussi bien sur la localisation de l'objet que sur son extraction proprement dite. Celle-ci est effectuée par une méthode originale appelée *pyramide locale* qui est une modification de la pyramide irrégulière permettant de focaliser la segmentation sur des zones bien précises de l'image.

> En ce qui concerne l'extraction d'objets dans les images et les vidéos, je propose deux outils interactifs suivis d'une méthode spatiale et une méthode spatio-temporelle de segmentation automatique fondée sur les objets (chapitre 3)

La représentation de vidéos fondée sur les objets

La deuxième problématique correspondant aux deux derniers chapitres s'intéresse à la représentation d'un plan vidéo[3] orientée sur l'étude et la classification des objets vidéo qu'il contient : le chapitre 4 introduit la problématique et présente un état de l'art des méthodes existantes. Le chapitre 5 expose notre approche.

Ce travail est motivé par le fait qu'à l'heure actuelle, il est possible de filmer ou de visualiser des vidéos dans n'importe quelle circonstance et à n'importe quel endroit car les techniques associées sont communément intégrées dans les systèmes portables qui inondent le marché. Les vidéos (films, clips, bandes-annonces, publicités, journal télévisé, fêtes de familles, souvenirs de vacances, visiophone 3G), font partie intégrante de notre quotidien. Le consommateur et

[3] Un *plan* est une portion d'une vidéo représentant une séquence d'images prises en continu par la caméra

l'utilisateur se retrouvent face à une masse importante de données difficile à gérer et à manipuler. Il est commun d'entendre "trop d'informations tue l'information". Il est donc nécessaire et urgent de trouver des techniques efficaces pour structurer, synthétiser, indexer, archiver, cataloguer, représenter, interroger et parcourir ces vidéos toujours plus nombreuses. Les chercheurs ont tout d'abord représenté le contenu des vidéos avec des images caractéristiques. Désormais, je crois qu'il est indispensable de représenter le contenu d'une vidéo grâce à des objets représentatifs. Par la suite, l'étude du comportement de ces objets pourrait permettre une représentation de haut niveau du contenu.

En conséquence, l'objectif de notre méthode exposée dans le dernier chapitre est de fournir une représentation synthétique d'une vidéo en sélectionnant des régions extraites automatiquement qui constituent des occurrences représentatives des objets contenus dans la séquence : les *objets-clés*. Une première étape de notre travail réside dans la classification intra-plan des régions extraites à partir d'un objet réel. Cette étape permet de construire une classe représentative d'un objet d'intérêt comportant un certain nombre de régions issues de celui-ci. Une deuxième étape consiste en la sélection intra-classe de la région la plus représentative de l'objet d'intérêt. L'originalité de notre approche, réside dans le fait que nous considérons que la qualité de l'extraction d'un même objet est inévitablement variable et ce, à cause du modèle utilisé, de la complexité de l'objet ou du contexte du plan vidéo, *etc.* Ainsi, nous proposons et discutons une méthode de sélection de la région extraite la plus représentative pour chaque objet filmé détecté. Pour cela, nous étudions l'évolution de l'extraction d'un même objet au cours d'un plan vidéo. Puis, par une sélection *ad hoc*, on conserve un représentant de l'objet sous la forme d'une région comportant les meilleures caractéristiques.

Pour répondre à cette problématique, je propose une approche originale de classification des régions extraites et de sélection des régions représentatives du contenu d'un plan vidéo (chapitre 5)

Chapitre 1

Introduction aux objets vidéo

Ce qui peut être montré, ne peut être dit.
Ludwig Wittgenstein, *Tractus.*

Sommaire

L'objectif de ce chapitre est de définir quelques notions importantes qui seront utilisées tout au long de ce manuscrit ainsi que de présenter le cadre dans lequel se situe l'étude. Pour cela, nous verrons tout d'abord le type de signaux à partir desquels la recherche d'objets est effectuée. Ensuite nous nous attacherons à définir clairement la notion délicate d'*objet* en soulevant les ambiguïtés qui pourraient exister entre l'objet réel $3D$, sa projection dans un espace $2D$ et le résultat final de son extraction. Enfin, nous terminerons ce chapitre par une présentation des domaines d'applications dans lesquels nos travaux peuvent s'inclure et où l'extraction d'objet peut être mise à profit.

1.1 Signaux analysés : une acquisition non maîtrisée

1.1.1 Le signal image

Nous allons aborder dans ce manuscrit le problème complexe de l'extraction d'objets dans les images numériques. Le signal analysé pourra donc être considéré comme une matrice $2D$ de points appelés pixels, où chaque dimension représente une dimension spatiale : la hauteur et la largeur. On considère donc les images comme une fonction de $\mathbb{R}^2$ dans $\mathbb{R}$. Les conditions de l'acquisition de l'image numérique ne sont pas supposées être connues.

L'objectif des méthodes dites *spatiales*, présentées par la suite, est d'engendrer à partir de cette seule matrice $2D$, une partition de l'image regroupant entre eux les pixels constituant chaque objet contenu dans l'image. Les méthodes spatiales nécessiteront des traitements dits *intra-image*. Dans le cas des méthodes automatiques aucune information *a priori* sur le contenu ne sera utilisée. Nous aurons le loisir par la suite de comparer ces méthodes à des traitements semi-automatiques permettant l'apport d'informations par un utilisateur. Nous verrons que l'élaboration d'une fonction totalement automatique constitue un fort enjeu mais également un véritable défi technologique de part les nombreux obstacles que cela comporte et le manque important d'information dans le signal étudié.

1.1.2 Le signal vidéo

Formalisation

Nous étudierons également des traitements dits *spatio-temporels*. Le signal alors utilisé sera issu d'une séquence vidéo. Il sera considéré comme une matrice de dimension $3 : 2D + t$, où là encore, les deux premières dimensions représentent les dimensions spatiales - largeur et hauteur - et la troisième représente le temps. Les points de la matrice sont également appelés *pixels* mais sont à associer à une image particulière correspondant à sa position temporelle. Mathématiquement, il s'agit d'une fonction de $\mathbb{R}^3$ dans $\mathbb{R}$.

L'objectif des méthodes spatio-temporelles est de remplir les mêmes fonctions que les méthodes spatiales et ceci pour chaque image de la séquence. Cependant, le principe est d'utiliser au mieux l'information temporelle inter-images pour améliorer le traitement en palliant le manque d'information de la matrice $2D$.

Acquisition

Il existe 2 grandes familles de prises de vue déterminant le type d'algorithmes spatio-temporels utilisables. La première concerne les séquences issues de caméras fixes. La deuxième regroupe les séquences acquises à l'aide d'une caméra mobile. Dans le domaine de la vidéo-surveillance par exemple, c'est le premier type de séquences qui est le plus souvent utilisé, bien que des traitements s'étendent maintenant aux caméras mobiles. Cependant, lorsque les domaines d'application concernent le multimédia, il est obligatoire de pouvoir prendre en compte le mouvement de la caméra puisque la prise de vue à l'aide de caméras mobiles est

la plus répandue. De plus, il n'y a aucun contrôle possible sur l'acquisition de la séquence car le traitement intervient après la production qui a été effectuée uniquement à des fins de diffusion de la vidéo. Certains procédés proposent désormais des traitements d'indexation lors du tournage intégrant des informations sur le sujet filmé, mais ils sont encore rares et compliquent considérablement la prise de vue. Dans la plupart des cas, l'analyse du contenu de la séquence ne dispose donc d'aucune information sur l'acquisition telles que les caractéristiques de la caméra, la balance des blancs ou le mouvement de la caméra, *etc.*

C'est dans ce cadre que cette étude se place. Nos méthodes spatio-temporelles traitent donc le cas des prises de vues issues de caméras mobiles et n'utilisent aucune information *a priori* sur le protocole d'acquisition.

1.1.3 L'échelle

Une propriété inhérente aux objets réels est qu'ils n'existent en tant qu'entités sémantiques seulement à une certaine échelle. Prenons l'exemple des branches d'un arbre : elles n'ont de sens que si on les observe à quelques mètres. Le concept d'arbre change complètement suivant la distance d'observation. Si on s'éloigne à des kilomètres, on parle de forêt. En revanche, si on l'observe au microscope, on parle alors de molécules. Le type d'informations extraites d'une image est donc largement déterminé par la relation entre la taille de la structure recherchée dans les données - l'échelle d'observation - et la taille de l'opérateur utilisé pour la recherche. Un des problèmes fondamentaux dans le traitement d'image est la détermination de la taille de l'opérateur et où l'utiliser. Dans nos méthodes présentées dans ce manuscrit, bien que des approches *multi-échelles* soient prises en compte, nous considérons tout de même que les objets contenus dans les images sont de tailles raisonnables. Ainsi, nous nous attacherons à extraire l'objet sémantique qui correspond à l'échelle de la prise de vue elle-même. C'est-à-dire qu'il n'est pas question d'extraire les branches d'un arbre dans le cas d'une prise de vue d'une forêt par exemple. Le principe reste d'extraire les objets d'intérêt qui motivent la prise de vue ; la taille de ces objets est donc liée à la taille même de l'image et non pas à sa résolution.

1.1.4 La couleur

Nos méthodes d'extraction et de classification d'objets sont orientées couleur. Ainsi, tout au long de ce manuscrit, nous évoquerons différents espaces colorimétriques permettant de rendre compte de la couleur associée à chaque pixel ou voxel. Chaque espace, détaillé dans l'annexe A, possède ses propres caractéristiques et donc ses propres avantages et inconvénients. L'objectif est d'utiliser l'espace couleur le plus adapté au traitement désiré. Ainsi chaque pixel est associé à plusieurs composantes le localisant dans l'espace couleur choisi. Cette étude se place dans le cadre des images représentant uniquement le domaine visible. Cependant, il est tout à fait envisageable d'étendre certaines techniques présentées aux domaines de l'infra-rouge voire à des images multispectrales.

1.2 Notion d'objet

La notion d'objet est une des notions les plus importantes dans ce manuscrit. Ainsi le but de ce paragraphe est de définir clairement les différentes terminologies se rapportant aux objets. On distinguera donc les objets réels et leurs représentations $2D$ dans les séquences vidéo : les objets vidéo.

1.2.1 Régions d'intérêt

Avant de rentrer davantage dans le détail en ce qui concerne les objets, il est nécessaire de définir la notion de *régions* et *a fortiori* de *régions d'intérêt*.

Le terme de *région d'intérêt*, ou ROI (*Region Of Interest*), est utilisé en opposition au terme *objet* pour désigner une zone de l'image qui ne se rattache pas forcément à une entité réelle sémantique mais qui répond à certains critères. Ce sont ces derniers qui définissent le degré d'intérêt de cette région. Une région délimite généralement une zone homogène de l'image selon un voire plusieurs critères bas niveaux tels que la couleur, la texture ou le mouvement. La figure 1.1.b présente une ROI homogène en mouvement. Est à noter la non correspondance des frontières de la ROI avec les contours contenus dans l'image.

(a) Image originale

(b) Région d'intérêt

(c) Objet vidéo

FIG. 1.1: Région d'intérêt *vs* objet vidéo

1.2.2 De l'objet réel à l'objet vidéo

La définition de la notion d'objet vidéo s'avère être une tâche difficile. Tout d'abord, un objet vidéo se rapporte à un objet réel $3D$ projeté sur un plan. Cet objet vidéo ne peut donc pas être considéré comme une instance ou une apparence prise à un instant donné de l'objet réel. Il représente de manière abstraite l'objet réel pendant toute la période où ce dernier est présent dans la séquence. L'apparence $2D$ d'un objet vidéo à un instant donné est appelée *plan de l'objet vidéo* (VOP : Video Object Plan). Il est possible de différencier l'objet vidéo capturé à un instant donné et le plan objet résultant : dans le cas d'une occultation à un instant t, l'objet vidéo est la projection de l'objet réel dans le plan, tandis que le VOP est la partie apparente (visible dans l'image) de l'objet vidéo. Ce phénomène est illustré dans la figure 1.2.

1.2.3 Objet d'intérêt

Les objets réels dont nous allons tenter de fournir une représentation par un ensemble de VOPs par exemple, sont dits *d'intérêt*.

L'objet d'intérêt désigne ici un objet sémantique que l'image ou le plan d'une séquence tente de mettre en valeur. Ainsi, une image peut contenir de nombreux objets sémantiques mais seulement un voire quelques objets d'intérêts. Les objets sémantiques restants sont considérés comme appartenant au fond. Cette décomposition *fond/objet d'intérêt* est relativement subjective et dépend de ce que recherche l'utilisateur ou le spectateur et de ce que le créateur (photographe, réalisateur) veut montrer.

(a) Image originale (b) Objet vidéo (c) Plan objet vidéo

FIG. 1.2: Objet vidéo *vs* plan objet vidéo

Certains domaines de recherche tentent de définir l'intérêt d'un point de vue physiologique avec par exemple les cartes de saillances [23]. Ces cartes ont pour but de localiser les points qui sont supposés stimuler davantage la rétine et sur lesquels l'attention du spectateur se focalise. Notre approche ici sera différente et propose d'utiliser les caractéristiques intrinsèques à la prise de vue tel que le mouvement pour définir les objets d'intérêt.

Cette notion d'intérêt introduit inévitablement la notion de *modèle* dont le rôle est de définir quels sont les objets extractibles par les algorithmes que nous allons présenter tout au long de ce manuscrit.

1.2.4 Modèle d'objets

Afin de détecter et d'extraire automatiquement des VOPs, il est nécessaire de définir un modèle d'objet auquel l'objet vidéo se rapporte et répond correctement. Ce modèle peut être un modèle physique (corps humain, visage) ou de mouvement (affine, quadratique, ...). Il est donc nécessaire de définir le modèle afin de savoir quels types d'objets l'algorithme peut traiter.

Le pouvoir de description du VOP vis à vis de l'objet réel d'intérêt est directement lié à la richesse du modèle utilisé. Intuitivement, un objet vidéo peut être défini par sa texture, sa forme, sa couleur ou encore son mouvement, *etc.* Il peut être rigide ou non. La notion d'objet vidéo induit une notion inévitablement sémantique en comparaison à une simple région d'intérêt comme il a été vu précédemment. Le paragraphe suivant présente la notion de VOP sémantique.

1.2.5 Plan objet vidéo sémantique

Dans cette étude, la qualité d'un VOP réside dans deux aspects. Le premier est la qualité de segmentation qui concerne la fidélité du masque du VOP par rapport au VO. Le deuxième aspect concerne la portée sémantique du VOP en lui-même.

D'une manière générale, la sémantique (du Grec *semantikós* provenant de *sema*, signe) est l'étude du *sens* des symboles et expressions. Les objets vidéo et leurs VOPs appartiennent effectivement à ce cadre symbolique puisqu'ils sont une représentation $2D$ de l'objet réel filmé. Un VOP est issu d'une certaine perception du monde réel : celle de la caméra. Ensuite cette capture ou représentation est présentée à l'utilisateur qui en a, à son tour une nouvelle perception. Pour tenter de définir la notion de VOP sémantique nous allons procéder à la

comparaison entre un mot qui sert à nommer un certain objet réel et un VOP qui représente ce dernier dans une séquence vidéo.

La discipline introduite par Alfred Korzybski, connue sous le nom de *sémantique générale* semble adaptée pour définir cette notion. Cette discipline dépasse le cadre symbolique de la sémantique. En effet, il s'agit ici de considérer le *sens* de façon opérationnelle, par la manière dont notre organisme réagit à son environnement. La sémantique générale englobe certes la sémantique comme cas particulier, mais s'oriente autant et davantage vers la neurophysiologie, la psychiatrie ou les théories de la communication. C'est bien sûr ces dernières qui vont nous intéresser ici.

« Ce qui peut être montré, ne peut être dit. » Ludwig Wittgenstein, Tractus.

Notre représentation du monde s'effectue par des perceptions ou interactions ayant :

- leurs limites (par exemple, le proche infrarouge invisible à l'oeil)
- leurs pertes (un son masqué par un autre)
- leurs éléments non-conscients (taux d'oxygène dans le sang)
- et d'autres enfin peuvent être sans rapport avec l'objet perçu (hallucinations, illusions d'optique, acouphènes...).

Notre perception du « réel » demeure partielle et personnelle. A l'inverse, les « objets » qui nous entourent pourraient également être décrits par des jeux de molécules, atomes, etc. en perpétuelle évolution, sans que notre compréhension globale y gagne. Notre esprit est donc amené à construire des « représentations » internes du monde extérieur (*cartes*) à l'aide d'informations *filtrées*. Ces cartes, symboliques (désignation verbale par exemple) ou non, ne prétendent nullement dupliquer exactement l'objet réel, dynamique et unique car : *« Quoi que vous disiez qu'une chose* ***est****, elle ne l'****est*** *pas ! »*.

- Une carte n'est pas le territoire qu'elle représente : les mots comme les images que l'on associent aux objets ne sont pas les objets réels. le mot « chien » comme l'objet vidéo chien ne mord pas, *etc.*
- Une carte ne recouvre pas tout le territoire qu'elle représente : le symbole omet de représenter certains « attributs » de l'« objet » qu'il représente.

Les vidéos ou les images sont déjà elles-mêmes, des représentations du monde réel. Les objets vidéo ainsi que leurs VOPs peuvent donc être associés à des cartes intermédiaires des objets réels définies par la sémantique générale. Ils fournissent à l'utilisateur une représentation non sans défaut ou limitation mais qui reste interprétable par ce dernier. Ainsi, un VOP considéré comme sémantique doit correspondre de manière optimum à la carte que l'utilisateur se fait de l'objet d'intérêt pour que ce dernier puisse reconnaître voire nommer l'objet d'intérêt représenté. Un objet vidéo sémantique doit privilégier la compréhension globale sans chercher à apporter des informations superflues.

1.2.6 Objet non rigide

Le terme d'objet *non rigide* s'applique aux objets déformables. Ce qui implique que leur forme n'est pas constante au cours du temps. C'est le cas de la plupart des objets naturels : végétaux, animaux, personnes. A ce type d'objets est associée la notion de mouvement non rigide qui rend compte de la déformation.

La reconnaissance et le suivi automatique de tel objet est une tâche extrêmement complexe qui nécessite souvent l'utilisation de modèles eux-mêmes très complexes. Cependant, la majeure partie des objets d'intérêt présents dans les séquences réelles sont non rigides. Il

nous paraissait donc important que nos méthodes spatio-temporelles puissent gérer ce type d'objets.

1.3 Domaines d'applications

De nombreux traitements peuvent être améliorés ou facilités par une connaissance du contenu de l'image. C'est pourquoi, la décomposition en objets des images motive de nombreux travaux de recherche. Nous allons voir grâce aux paragraphes suivants les domaines très diverses qui peuvent être concernés par une telle décomposition.

1.3.1 Vidéo surveillance et vidéo assistance

L'un des principaux domaines d'applications associé à l'extraction d'objets, mais non le moins controversé, est celui de la vidéo-surveillance. L'utilisation de caméras dédiées à la surveillance s'est considérablement développée en France. Ces équipements ont longtemps été fixes. Cependant, les nouvelles technologies intègrent des caméras mobiles mises en réseaux qui permettent de couvrir un plus large champ et de procéder à des suivis de véhicules ou de passants d'une caméra à l'autre. Les algorithmes sont soumis aux contraintes de temps réels. De plus, dans ce type d'application, la qualité des masques définissant l'objet n'est pas une priorité. L'objectif recherché est la robustesse face au bruit et au changement de conditions d'éclairement de la scène filmée.

En revanche, dans le cadre de la détection automatique d'évènements en vidéo-assistance, il est intéressant d'avoir des masques de qualité des personnes filmées afin de pouvoir reconnaître avec précision leur posture.

1.3.2 Manipulation du contenu

Composition d'image

Les images numériques sont de plus en plus utilisées à l'instar des images argentiques. Elles donnent, aussi bien aux professionnels qu'au grand public, la possibilité de modifier en post-traitement le rendu et le contenu des images. La décomposition d'une image en objets de qualité permet de faciliter la *composition d'image*. C'est un procédé dont le but est de combiner plusieurs parties d'images de sources diverses afin de former une seule nouvelle image. On distingue deux types de compositions : dans le cas où l'objectif est de représenter une scène qui pourrait avoir existé, le résultat s'appelle la photo-montage. En revanche, si l'image est destinée à véhiculer une idée nouvelle par la juxtaposition des différents éléments, on se trouve dans le cadre du collage. Les éléments principaux de la composition consistent en l'extraction des éléments des images sources et le positionnement de ces derniers dans l'image résultat. La figure 1.3 présente un exemple simpliste de collage[1]. Bien sûr, la sélection et l'extraction des différents éléments ne suffisent pas à réaliser une composition. Les aspects plus fins de l'intégration exigent fusion, assortiment des couleurs, et attention générale au détail mais cela ne concerne pas notre étude.

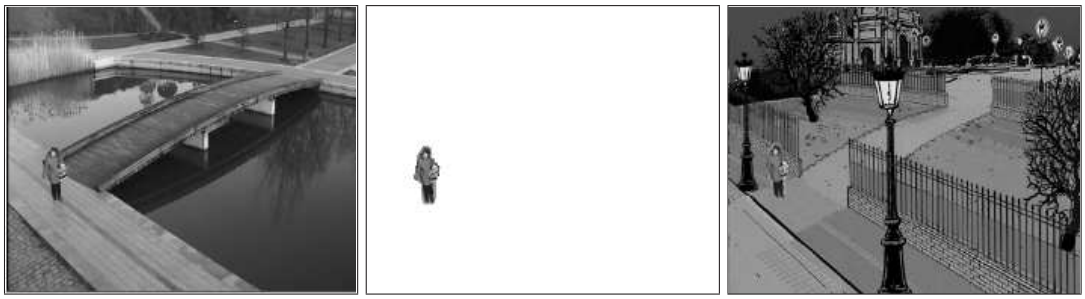
Réalité mixte

De nombreuses applications ont été développées afin de faciliter l'intégration d'éléments du monde réel dans un monde virtuel. Le principe est d'extraire une entité d'une séquence naturelle, par exemple une personne, et de l'intégrer au sein d'un environnement de synthèse [7]. La figure 1.4 illustre ce principe par un résultat obtenu dans le cadre d'un projet européen

[1] Objets extraits à l'aide de notre application de segmentation interactive : ExtraK'Obs

FIG. 1.3: Exemple de composition d'image par simple collage

de l'IST[2] : *Art Live*[3], auquel a participé le LIS. Ce type de procédé se retrouve dans les jeux vidéo ou dans les équipements de télécommunication grand public tels que les *webcams*.



(a) Image originale (b) Extraction (c) Intégration

FIG. 1.4: Exemple de réalité mixte (extrait de [7])

1.3.3 Compression vidéo - standard MPEG-4

Le standard MPEG-4 est une norme de compression de données numériques audio/vidéo créée par le comité MPEG (*Motion Picture Experts Group*). Il intègre les caractéristiques de compression vidéo des standards MPEG-1 et MPEG-2 comme la compensation de mouvement et la transformée en cosinus discrète (DCT). L'enjeu prioritaire de ce format n'est pas d'améliorer les taux de compression par rapport à ces normes mais plutôt d'évoluer vers une compression adaptée au contenu sémantique de la vidéo. Ainsi, des débits élevés de données favorisant la qualité peuvent être accordés aux zones d'intérêt. Tandis que les zones non significatives nécessitant une qualité moindre, bénéficient d'un débit de données plus faible résultant d'une plus forte compression. Pour plus de détails sur les normes de compression

[2]Information Society Technologies
[3]Architecture and authoring Tools Prototype for Living Images and Videos Experiment

vidéo MPEG-1 et MPEG-2 qui ne font pas l'objet de cette étude, le lecteur pourra se reporter à ce site internet[4].

Ainsi, la principale innovation apportée par le MPEG-4 en rapport avec nos travaux, réside dans le codage fondé sur les objets qui permet de prendre en compte le contenu de la vidéo. Pour cela, ce standard intègre la notion d'objet vidéo (VO : *Video Object*) et adopte une représentation des images par un ensemble de plans objets vidéo (VOP : *Video Object Plan*). Conformément à ce qui a été présenté précédemment, un VOP est une représentation un instant donné d'un VO. Les VOPs sont donc de formes quelconques et de plus variables. Ainsi, comme le montre la figure 1.5, le codage d'un VOP traite l'évolution de sa texture et de sa forme au cours de la séquence.

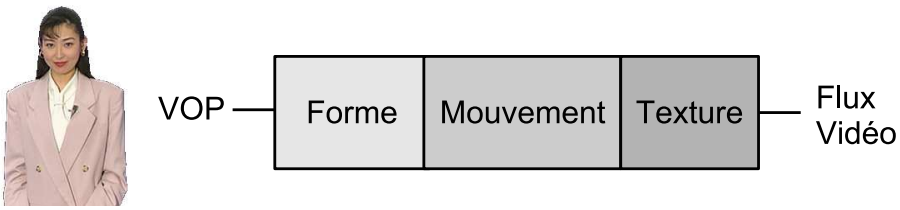


Fig. 1.5: Codage D'un VOP par le standard MPEG-4

L'approche objet permet donc d'améliorer la compression en effectuant un codage (compression) et un décodage (reconstruction) adaptés à chaque type de VO. Mais elle permet également d'augmenter considérablement l'interactivité avec l'utilisateur puisque chaque VO est manipulable indépendamment. En effet, le but est de permettre à l'utilisateur de pouvoir supprimer des informations qu'il ne désire pas ou bien d'accéder à des informations supplémentaires dissimulées. L'utilisateur peut ainsi modifier les attributs de la scène en changeant la position des objets, les rendant visibles ou invisibles, *etc.*

La norme MPEG-4 ne spécifie encore aucun standard d'extraction d'objets. Il s'agit donc d'un problème largement ouvert mais qui constitue un défi technologique important.

1.3.4 Indexation

Les nouvelles technologies de télécommunications et leur intégration omniprésente dans nos équipements microélectroniques portables (téléphones portables, *smartphones*, assistants numériques personnels, ...) rendent extrêmement simple l'acquisition de séquences vidéo et leur diffusion sur des réseaux locaux ou sur l'Internet à des fins professionnelles ou non. Cet engouement pour ce type de technologies engendre une masse considérable de documents audio-visuels, ce qui impose le développement de nouveaux outils de recherche automatiques d'images ou de séquences vidéo dans ces bases de données sans cesse croissantes. La difficulté réside dans l'extraction des attributs et la caractérisation automatique du contenu. La représentation de vidéos ou d'images fondée sur les objets constitue un moyen intéressant pour un procédé d'indexation orienté sur le contenu. Le standard MPEG-7 a été introduit dans le but de standardiser la description d'un document audiovisuel.

Standard MPEG-7

La norme MPEG-7 permet d'intégrer la notion d'objet en utilisant des descripteurs (Ds) bas-niveaux s'intéressant à la syntaxe d'une représentation selon certains critères (forme, couleur, texture, ...). De plus, elle définit des schémas de description (*Description Schemes DSs*)

[4] *http : //www.chiariglione.org/mpeg/*

qui peuvent regrouper plusieurs Ds en intégrant des liens entre eux permettant une description plus haut-niveau voire sémantique. Son objectif est de faciliter l'indexation et la recherche des documents multimédia dans de larges bases de données. La norme définit également un langage de description des contenus multimédia : la DLL (Description Definition Language) qui est un dérivé du format de langage XML [5] et qui permet de gérer et de faire évoluer les descripteurs et les schémas de description. Cependant, cette norme ne s'intéresse ni à la définition exacte des descripteurs ni aux algorithmes permettant d'extraire les attributs. Les technologies à utiliser pour extraire le contenu ne sont donc pas encore déterminées. Pour obtenir plus d'informations sur cette norme il est possible de consulter cette documentation en ligne [1].

Le principe de l'analyse orientée sur le contenu et plus particulièrement sur les objets est de fournir aux procédés d'indexation une définition de l'objet toujours plus précise afin de permettre une recherche efficace.

Interrogation de bases d'images

Les systèmes d'interrogation de bases d'images par le contenu (CBIR : *Content-based Image Retrieval*) sont généralement fondés sur l'extraction de caractéristiques bas-niveau tels que la couleur, la texture, ... Ces caractéristiques extraites des images servent à orienter la recherche. C'est par exemple l'approche des procédés de recherche d'images dans de larges bases de données tels que QBIC développée par P. Flickner *et al.* [38] ou VisualSEEK [109] présentée par J. Smith.

Les auteurs de [54] proposent de solliciter l'utilisateur pour définir un objet à partir d'une première partition de l'image obtenue automatiquement sur des critères bas-niveau. L'objet permet d'orienter la recherche sur un critère beaucoup plus sémantique.

De même G.-A. Bilodeau *et al.* proposent dans [12] un système nommé PLASTIQUE de recherche d'images fondée sur les objets. Cependant, le principe est ici de fournir de manière automatique une représentation de l'objet qui est ensuite utilisée pour guider la recherche. La figure 1.6 présente l'interface du système avec en haut à gauche la requête et à droite les résultats de la recherche.

Zhong *et al.* présentent dans [133] un principe de recherche d'objet à partir d'un objet requête. La différence avec les autres méthodes est que la base de données dans laquelle est effectuée la recherche contient uniquement des objets qui peuvent être de différents types (personnes, animaux, véhicules...).

Le principe de la majorité de ces systèmes est de fournir à l'utilisateur un ensemble de résultats contenant les objets ou les images les plus ressemblants à la requête parmi les éléments de la base de données.

Création de documents vidéo structurés et interactifs

L'objectif de la création de documents vidéo interactifs est d'apporter à une vidéo des fonctionnalités supplémentaires permettant à l'utilisateur de récupérer facilement des informations sur le contenu et l'organisation de la séquence. Cette structuration doit bien sûr être la plus automatisée possible. Cependant, dans les systèmes actuels, elle nécessite encore de nombreuses interventions manuelles afin d'obtenir le résultat escompté.

[5] Extensible Markup Language (« langage de balisage extensible ») généralement abrégé en XML, est un standard du World Wide Web Consortium qui sert de base pour créer des langages de balisage : c'est un « méta-langage ». En ce sens, XML permet de définir un vocabulaire et une grammaire associée sur base de règles formalisées. Il est suffisamment général pour que les langages basés sur XML, appelés aussi dialectes XML, puissent être utilisés pour décrire toutes sortes de données et de textes. Il s'agit donc partiellement d'un format de données. L'extensibilité de XML est principalement assurée par la notion d'espace de nommage.

FIG. 1.6: Recherche d'images fondée sur les objets (extrait de [12])

L'INRIA a développé, en partenariat avec Alcatel Corporate Research Center (CRC), un prototype d'environnement de création de documents "hypervidéo" appelé *VideoPrep* et d'exploitation de ces vidéos structurées et interactives appelé *VideoClic* [8]. Les travaux ont été menés par les équipes de Roger Mohr du projet MOVI de l'INRIA Rhône-Alpes et de Patrick Bouthemy du projet VISTA à l'INRIA Rennes. L'environnement *VideoPrep* introduit des traitements automatiques, tout en offrant une interface appropriée d'édition des résultats (ajout, élimination, correction). Il permet de structurer les documents vidéo en 3 étapes :

1. Découpage de la séquence en plans. L'interface propose de visualiser les plans déterminés automatiquement à l'aide de miniatures (cf. partie droite de la figure 1.7.a) et d'un bandeau temporel (cf. bas de la figure 1.7.a).
2. Édition des objets vidéo contenus dans le plan. Le logiciel propose une détection automatique des régions du premier plan notifiées à l'utilisateur par un détourage superposé à la vidéo. L'utilisateur peut ensuite éditer ces objets visualisables à droite de l'écran (cf. figure 1.7.b) en modifiant, créant ou supprimant des zones directement sur l'image. Il peut également leur associer un nom et/ou une description.
3. Indexation des objets. L'utilisateur groupe les objets dans des classes données par simple déplacer-coller avec la souris (cf. partie droite de la figure 1.7.c).

VideoClic permet à l'utilisateur d'interagir avec la vidéo : rechercher la séquence ou le plan dans lequel apparaît l'objet ou d'obtenir des informations précises sur l'objet sélectionné par simple clic sur l'image.

1.3.5 Des projets novateurs

Une équipe de *Microsoft* basée en Asie, travaille actuellement sur un projet novateur nommé *Photo2Search*[6]. Le principe est de fournir à l'utilisateur des informations sur un objet, un lieu à partir de sa photo prise par un système nomade (téléphone portable, PDA, ...). Pour cela, la photo est soumise à une base de données du service par l'intermédiaire des réseaux de télécommunications. La philosophie de système est bien résumée par les propos de Xing Xie,

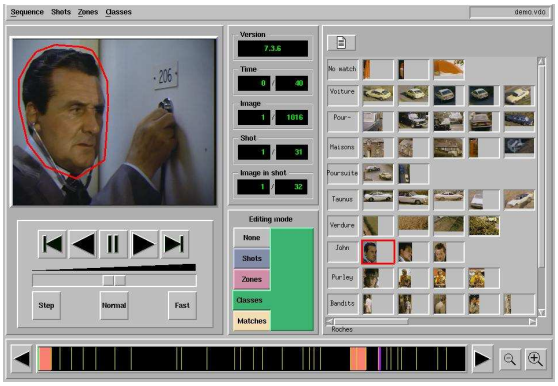
[6] http ://research.microsoft.com/displayArticle.aspx ?id=1434

l'un des ingénieurs en charge du projet : « *Vous voyez un objet dans le monde réel et vous souhaitez trouver les informations qui le concernent dans le monde numérique - par exemple, son prix sur le Web, des commentaires d'utilisateurs ou des sites qui lui sont consacrés. [. . .] Utiliser une image de l'objet est la méthode la plus commode à employer* »

Une meilleure compréhension du contenu et une décomposition en objets des images et des vidéos constituent un véritable défi technologique mais les possibilités que cela pourrait apporter sont colossales. Les projets novateurs et ambitieux de ce type n'ont pas fini de se développer dans les années futures.

(a) Découpage en plans

(b) Édition des objets



(c) Indexation des objets

FIG. 1.7: Fonctionnement de l'application *Videoprep* [7]

Chapitre 2

De la segmentation à l'extraction d'objets d'intérêt

Sommaire

Ce chapitre présente un état de l'art des méthodes concernant la représentation fondée sur les objets contenus dans les images. Son objectif principal est de montrer l'utilisation de la segmentation dans le cadre de l'extraction d'objet. L'étude se porte tout d'abord sur des procédés de segmentation interactifs puis s'étend aux méthodes automatiques spatiales et spatio-temporelles fondées sur les objets. Cependant, avant de présenter ces différentes techniques, nous allons nous intéresser au problème de la sur-segmentation inhérent aux méthodes classiques de segmentation afin de mettre en évidence la difficulté de l'extraction d'objet.

2.1 Problématique de la sur-segmentation

Ce chapitre traite de méthodes de segmentation d'images dont la définition générique est la suivante :

Soit I une image et E une partition de I constituée de sous-ensembles connexes :

$$E = R_1,\ R_2, \ldots,\ R_n \ avec\ R_i \neq \emptyset\ et\ R_i\ connexes\ (\forall i = 1 \ldots n)$$
$$\forall (i,\ j)\ avec\ i \neq j,\ R_i \cap R_j = \emptyset$$
$$I = \cup R_i\ (pour\ i = 1 \ldots n)$$

Soit P un prédicat d'homogénéité appliqué aux pixels. E est une segmentation de I, selon le prédicat P, ssi :

$$\begin{cases} P(R_i) = \ vrai\ \forall i = 1 \ldots n \\ P(R_i \cup R_j) = \ faux,\ \forall i \neq j\ tels\ que\ R_i\ et\ R_j\ sont\ adjacents \end{cases}$$

Il existe de nombreuses méthodes de segmentation parmi lesquelles il est possible de distinguer deux familles principales : (1) les méthodes orientées régions et (2) les méthodes orientées contours. La première famille utilise les critères d'homogénéités tandis que la deuxième est fondée sur les critères d'hétérogénéités de l'image.

Problématique : les méthodes de segmentation spatiales classiques fournissent généralement une sur-segmentation de l'image car elles s'attachent à partitionner l'image en régions homogènes selon un, voire seulement quelques critères.

C'est le cas des méthodes de classification telles que l'agrégation autour de centres mobiles (ou de nuées dynamiques), appelées *k-means* [85] par les anglophones. Il en est de même pour les méthodes de la ligne de partage des eaux (*watershed*) [88, 123] ou des C-moyennes floues (*fuzzy C-means*) [35, 11]. Généralement, elles ne peuvent pas répondre à une approche objet de la segmentation sans extension adaptée. La figure 2.1[1] confronte le phénomène de sur-segmentation (fig. 2.1.b) par rapport au but recherché (certes utopique) qui est l'extraction de l'objet d'intérêt[2] (fig. 2.1.c).

Ce chapitre présente un état de l'art des méthodes de segmentation dont l'objectif est de limiter le phénomène de sur-segmentation et qui proposent une partition constituée d'un

[1] Photo de l'auteur *saigoncine* disponible sur le site *http* : *//www.flickr.com.* Distribution *Creative Commons* avec paternité et sans utilisation commerciale

[2] Extraction réalisée de manière semi-automatique avec notre application ExtraK'obs

(a) Bouddha de Kamakura

(b) Résultat d'une segmentation (quelques centaines de régions)

(c) Segmentation idéale du Bouddha

FIG. 2.1: Illustration du problème de sur-segmentation

ensemble de régions mais dont la plupart sont assimilables à un contenu sémantique de l'image. On parlera ici de méthode d'extraction de régions d'intérêt ou de segmentation fondée sur les objets. Nous n'évoquerons pas ici les méthodes faisant appel à des modèles développés pour des applications précises. Nous nous intéresserons au méthodes plus génériques qui tentent de répondre au problème de l'extraction d'objets au niveau pixel dans les images naturelles sans connaissance *a priori* du contenu.

Le système visuel humain est capable de définir un objet comme un regroupement d'entités plus ou moins complexes et de lui associer un sens. Les méthodes de segmentation automatiques classiques fondées sur des critères bas-niveau ont du mal à traiter de l'aspect sémantique et se heurtent très vite au côté inévitablement subjectif des résultats. C'est pourquoi, dans un premier temps, l'état de l'art porte sur des méthodes interactives où l'apport de l'expertise de l'homme est crucial et permet de palier ce problème. La suite du chapitre présente des méthodes automatiques, d'abord spatiales puis spatio-temporelles, permettant d'approcher une segmentation objet.

2.2 Extraction interactive d'objets dans les images fixes

Ce paragraphe présente les méthodes d'extractions interactives d'objets. L'expertise de l'homme est sollicitée soit à l'initialisation soit au cours de l'algorithme afin de remettre en cause les décisions de l'application. Ainsi, l'utilisateur peut injecter de l'information *a priori* et du sens dans les résultats de segmentation en conservant l'automatisation d'une majeure partie des traitements. Un état de l'art récent des méthodes d'extractions interactives de régions du premier plan est présenté par A. Rother dans [102].

2.2.1 Masque binaire

G. Foret fait allusion dans sa thèse [40] à une technique d'Amal Mahboubi *et al.* qui proposent dans [79] de laisser le soin à l'utilisateur de générer un masque binaire représentant grossièrement l'objet d'intérêt. Ce masque binaire est ensuite appliqué sur une partition obtenue automatiquement par segmentation spatiale afin de classer les régions en 3 classes différentes qui sont : la classe *objet*, la classe *fond* et la classe *indéterminé* regroupant les pixels ambigus du bord de l'objet. La classification des régions de cette dernière classe est remise en cause et n'intervient que plus tard dans l'algorithme. La représentation de chaque objet est obtenue par fusion hiérarchique des régions fondée sur leur mouvement respectif. Cette méthode

permet d'obtenir une partition sémantique tout en ne demandant qu'une faible participation à l'utilisateur.

2.2.2 *Magic Wand*

Le plus populaire des outils d'extraction de régions du premier plan est la *Magic Wand* [115] fondée sur le principe de croissance de région. Il est implanté dans la plupart des logiciels de traitements d'image. L'algorithme est initialisé à partir d'un pixel ou une région de pixels sélectionné par l'utilisateur. L'objectif est d'étendre la zone de sélection en absorbant tous les pixels voisins répondant à une certaine tolérance sur les statistiques couleur de la région spécifiée. Cette tolérance est ajustable mais dans la plupart des cas se révèle compliquée à définir pour obtenir le résultat escompté. Ce type de méthode est adapté à l'extraction d'objets plutôt homogènes et bien contrastés avec le fond.

2.2.3 Ciseaux intelligents (*Intelligent scissors*)

Les ciseaux intelligents [91] constituent un outil interactif de segmentation. Cette technique a été développée par Eric N. Mortensen et Wiliam A. Barrett dans le but de faire de la composition d'images. Le principe est de détecter un contour d'un objet potentiel dans la zone où se trouve le curseur de la souris dirigé par l'utilisateur. Ainsi, pour guider l'extraction de l'objet à l'aide des ciseaux intelligents, l'utilisateur doit suivre grossièrement son contour avec le curseur.

Cet outil est conçu pour pouvoir extraire un objet quelle que soit la complexité du fond. L'avantage est que l'utilisateur voit en temps réel ce que donne le contour qu'il est en train de tracer par l'intermédiaire des ciseaux intelligents. Il peut ainsi corriger et guider davantage le procédé. La plupart des méthodes présentées dans ce chapitre s'efforcent de trouver un contour optimal en fonction de l'initialisation de l'utilisateur puis de le raffiner, tandis que les ciseaux intelligents permettent de choisir un contour parmi tous les contours optimaux issus d'un seul germe.

Cette méthode aborde la programmation dynamique (DP : Dynamic Programming) appliquée à la définition d'un contour dans une image, comme un problème de recherche dans un graphe à 2 dimensions. Le but est de trouver le chemin optimal entre un nœud de départ et un nœud d'arrivée. Ainsi, un nœud représente un pixel et les arcs sont créés entre chaque pixel voisin ; le voisinage étant défini en 8-connexité. Le chemin est optimal lorsque sa fonction de coût correspondante est elle aussi, minimale. Pour cela, cette fonction est définie comme étant la somme pondérée de plusieurs fonctions de localisation de contours qui sont :

Laplacian Zero-Crossing : f_Z qui localise les contours.

Gradient Magnitude : f_G qui renseigne sur l'intensité du contour.

Gradient Direction : f_D qui impose une contrainte de régularité au contour.

Soit $l(p, q)$ la fonction de coût local entre un pixel p et un pixel q voisin :

$$l(p, q) = \omega_Z.f_Z(q) + \omega_G.f_G(q) + \omega_D.f_D(p, q) \tag{2.1}$$

Où chaque ω représente le poids correspondant à la fonction de localisation des contours.

Les arêtes du graphe sont affectées de ces valeurs, le but étant alors de trouver les chemins les plus courts entre le point de départ et chaque pixel de l'image selon une méthode de type Dijkstra [34]. Une fois que les chemins associés aux coûts minimaux reliant le pixel de départ et chaque pixel de l'image ont été calculés, l'outil de segmentation détermine le chemin optimal à partir d'un point « libre » spécifié par la position du curseur dirigé par l'utilisateur.

Deux principes, le *cooling* et le *on-the-fly-training*, sur lesquels nous ne nous attarderons pas ici, ont été développés pour optimiser l'algorithme. Le premier se charge de définir automatiquement des germes à la place de l'utilisateur au cours de la définition du contour. Tandis que le deuxième est là pour optimiser le temps de calcul de l'algorithme qui est à l'origine très important.

2.2.4 *Bayes Matting* et *Knockout* 2

L'outil *Bayes Matting* nécessite 3 zones définies par l'utilisateur : les 2 zones connues du fond et de l'objet et une troisième zone indéfinie où cohabitent par « transparence » les deux entités. Ce découpage de l'image est appelé la *Trimap* et définit les trois zones T_B, T_F et T_U. Nous conservons ici les notations anglaises pour garder la cohérence avec la littérature :

- B pour Background (fond)
- F pour Foreground (premier plan ou objet)
- U pour Undefined (zone indéfinie)

La méthode est fondée sur une modélisation probabiliste orientée couleur. L'évaluation du contour par α-*matting* [26] permet de séparer l'objet du fond. Les contraintes pour obtenir un résultat de qualité sont : une zone indéfinie étroite et des distributions couleur bien séparées entre les 2 éléments.

L'outil *Knockout* 2 utilise le même type de méthode et fournit des résultats similaires voire de moins bonne qualité selon les cas, d'après [26].

2.2.5 *GraphCut*

Le *GraphCut* est une technique très puissante qui utilise le même type d'initialisation que l'outil *Bayes Matting* (trimap) et utilise également une modélisation probabiliste de la couleur. Elle est capable de gérer l'extraction dans des cas où l'objet est similaire au fond (phénomène appelé camouflage) et où la distribution couleur de l'un et de l'autre n'est pas distinctement séparée. Le principe général est de modéliser la distribution caractéristique des pixels de l'objet et celle du fond à l'aide de respectivement, T_F et T_B. Les pixels de T_U sont attribués à l'objet ou au fond selon la probabilité qu'ils appartiennent à l'une ou l'autre des distributions. Deux interactions sont possibles : soit l'utilisateur place manuellement les zones T_B et T_F, soit, comme le proposent les auteurs de [129], l'utilisateur place de manière approximative la zone T_U formant un bande fermée sur le contour de l'objet.

Dans [17], l'image est modélisée par un tableau de niveaux de gris à une dimension $z = (z_1, \ldots, z_n, \ldots, z_N)$ indexé par l'index n. La segmentation de l'image est modélisée comme un tableau de valeurs d'opacité $\underline{\alpha} = (\alpha_1, \ldots, \alpha_n, \ldots, \alpha_N)$ en chaque pixel. Généralement $0 \leq \alpha_n \leq 1$, afin de traduire l'opacité de l'objet par rapport au fond. Cependant pour réaliser une segmentation franche : $\alpha_n \in \{0, 1\}$, avec 0 symbole du fond et 1 de l'objet. Les paramètres $\underline{\theta}$ représentent les distributions des niveaux de gris du fond et de l'objet. Ils s'apparentent à des histogrammes de niveaux de gris, un pour le fond et un pour l'objet :

$$\underline{\theta} = \{h(z, \alpha),\ \alpha = 0, 1\} \tag{2.2}$$

Les histogrammes sont générés directement à partir des pixels étiquetés des régions issues de la *trimap* T_B et T_F. La segmentation consiste à estimer les variables d'opacité $\underline{\alpha}$ à partir des données z et du modèle $\underline{\theta}$ dans T_U.

On définit une fonction E dont le minimum doit correspondre à une segmentation de qualité. Pour cela elle doit être guidée par les histogrammes de niveaux de gris du fond et de l'objet en faisant en sorte que l'opacité reste cohérente, c'est-à-dire respectant le fait que les objets sont solides et connexes. Ceci se traduit par la fonction d'énergie E dont la forme de

Gibbs est donnée par :

$$E(\underline{\alpha},\ \underline{\theta},\ z) = U(\underline{\alpha},\ \underline{\theta},\ z) + V(\underline{\alpha},\ z) \tag{2.3}$$

Le terme U évalue la correspondance de la distribution de l'opacité $\underline{\alpha}$ avec les données z, selon le modèle donné $\underline{\theta}$ construit à partir des histogrammes. Il est défini de la façon suivante :

$$U(\underline{\alpha},\ \underline{\theta},\ z) = \sum_{n} -log(h(z_n,\ \alpha_n)) \tag{2.4}$$

Le terme de régularité s'exprime par l'équation suivante :

$$V(\underline{\alpha}, z) = \gamma \sum_{(n,m),m\in V_n} \delta_{\alpha_m}^{\alpha_n} \cdot exp^{-\beta} \cdot \frac{1}{dis(m,\ n)} \tag{2.5}$$

Où $\delta_{\alpha_m}^{\alpha_n}$ est le symbole de Kronecker et V_n détermine l'ensemble des pixels voisins du pixels n. Le voisinage est défini en 8-connexité entre pixels adjacents. $dis(\cdot)$ représente la distance Euclidienne des pixels voisins. Cette énergie favorise la cohérence entre régions de niveaux de gris similaires. Lorsque la constante β est nulle, la régularité est identique en tout point dont le degré est déterminé par la constante γ. D'après Boykov [17], la régularité est plus performante lorsqu'elle est relâchée dans les régions de fort contraste. La constante β est donc donnée par :

$$\beta = \frac{(z_m - z_n)^2}{2\sigma^2} \tag{2.6}$$

En effet, cette fonction pénalise fortement les discontinuités entre pixels similaires ($(z_m - z_n) < \sigma$). Tandis que dans le cas où les pixels sont différents ($(z_m - z_n) > \sigma$), elle induit une pénalité faible. Intuitivement, cette fonction peut être interprétée comme la distribution d'un bruit parmi les pixels voisins d'une image et σ peut être considéré comme le bruit induit par la prise de vue. La constante γ a été obtenue en maximisant les performances sur 15 images. La valeur est fixée à 50. Selon [13], cette valeur permet de s'adapter à un grand nombre de catégories d'images différentes. Finalement, la segmentation revient à estimer le minimum global de cette fonction d'énergie :

$$\underline{\hat{\alpha}} = \underset{\underline{\alpha}}{\operatorname{argmin}}\, E(\underline{\alpha}, \underline{\theta}) \tag{2.7}$$

La minimisation est réalisée à l'aide d'un algorithme de coupure de graphe appelé *max-flow* présenté dans [18].

2.2.6 *GrabCut*

La technique du *GrabCut*[102] est une optimisation du *GraphCut*. Premièrement, l'image n'est plus modélisée par des histogrammes de niveaux de gris mais par un mélange de Gaussiennes calculé dans l'espace couleur RGB. Deuxièmement, l'algorithme de coupure de graphe réalisé auparavant en une passe est remplacé par un algorithme itératif alternant des phases d'estimation et d'apprentissage. Troisièmement, Les interactions de l'utilisateur sont allégées puisqu'il initialise la *trimap* en positionnant uniquement la zone T_B autour de l'objet à l'aide des outils classiques de sélection (rectangle, lasso...). Ainsi, $T_F = \emptyset$ et $T_U = \overline{T_B}$. Les itérations successives permettent de déterminer les pixels de T_U appartenant au fond ou à l'objet.

Les variables d'opacité $\underline{\alpha}$ sont initialisées de la manière suivante :

$$\begin{cases} \alpha_n = 0 \ pour\ n \in T_B \\ \alpha_n = 1 \ pour\ n \in T_U \end{cases} \tag{2.8}$$

Afin d'adapter le *GraphCut* à la couleur, chaque pixel z_n est considéré dans l'espace RGB. Chaque ensemble de pixels de T_B et T_U est modélisé par un mélange de Gaussiennes à K composantes (Classiquement, $K = 5$). Un vecteur supplémentaire $k = (k_1, \ldots, k_n, \ldots, k_N)$ est donc introduit dans l'algorithme avec $k_n \in \{1, \ldots, K\}$. k affecte une unique composante à chaque pixel. La composante est choisie dans le mélange associé au fond ou à l'objet en accord avec la valeur $\alpha_n = 0$ ou 1. La fonction d'énergie de Gibbs E donnée à l'équation 2.3 devient alors :

$$E(\underline{\alpha},\ k,\ \underline{\theta},\ z) = U(\underline{\alpha},\ k,\ \underline{\theta},\ z) + V(\underline{\alpha},\ z) \tag{2.9}$$

L'expression de U doit également intégrer le nouveau modèle couleur. Ainsi, l'expression de U devient :

$$U(\underline{\alpha},\ k,\ \underline{\theta},\ z) = \sum_n D(\alpha_n,\ k_n,\ \underline{\theta},\ z_n) \tag{2.10}$$

Où $D(\alpha_n,\ k_n,\ \underline{\theta},\ z_n) = -logp(z_n|\alpha_n,\ k_n,\ \underline{\theta}) - log\pi(\alpha_n,\ k_n)$ et $p(\cdot)$ la distribution de probabilité Gaussienne et $\pi(\cdot)$ les poids des mélanges.

$$\begin{aligned} D(\alpha_n,\ k_n,\ \underline{\theta},\ z_n) &= -log\pi(\alpha_n,\ k_n) + \frac{1}{2} logdet\left(\Sigma(\alpha_n,\ k_n)\right) \\ &+ \frac{1}{2}\left[z_n - \mu(\alpha_n,\ k_n)\right]^T \Sigma(\alpha_n,\ k_n)^{-1}\left[z_n - \mu(\alpha_n,\ k_n)\right] \end{aligned} \tag{2.11}$$

Les paramètres du modèle d'estimation symbolisés par $\underline{\theta}$ deviennent :

$$\underline{\theta} = \{\pi(\alpha,\ k),\ \mu(\alpha,\ k),\ \sigma(\alpha,\ k),\ \alpha = 0, 1,\ k = 1 \ldots K\} \tag{2.12}$$

C'est-à-dire : les poids π, les moyennes μ et les matrices de covariances Σ des $2K$ composantes des Gaussiennes des 2 mélanges du fond et de l'objet. Le terme de régularité V reste inchangé à part la distance Euclidienne qui est maintenant exprimée dans un espace $3D$.

$$\beta = \frac{\|z_m - z_n\|^2}{2\sigma^2} \tag{2.13}$$

Algorithme 1 : Minimisation itérative

1. Assignation de chaque pixel de T_U à une composante du mélange de couleur :

$$k_n := \underset{k_n}{\operatorname{argmin}}\, D_n(\alpha_n,\ k_n,\ \underline{\theta},\ z_n)$$

2. Mise à jour des paramètres du mélange de Gaussiennes :

$$\underline{\theta} := \underset{\underline{\theta}}{\operatorname{argmin}}\, U(\underline{\alpha},\ k,\ \underline{\theta},\ z)$$

3. Estimation de la segmentation :

$$\min_{\{\alpha_n : n \in T_U\}} \min_k E(\underline{\alpha}, k, \underline{\theta}, z)$$

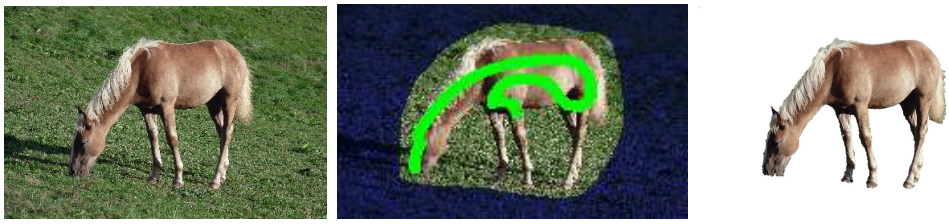
4. Retour à l'étape 1 jusqu'à convergence

La minimisation itérative contient 4 étapes exposées dans l'algorithme 1. Ces différentes étapes permettent de raffiner les paramètres d'opacité α_n en utilisant à chaque itération les pixels nouvellement étiquetés (comme fond ou objet) de la région T_U pour affiner le modèle de mélanges de Gaussiennes $\underline{\theta}$. Pour plus de détails sur la méthode de minimisation nous proposons au lecteur de se reporter aux travaux de C. Rother [102].

Une fois le traitement effectué, l'utilisateur peut agir sur les zones où la classification a causé des erreurs et ainsi affiner le résultat. Pour cela trois outils lui sont proposés : le *foreground brush*, le *background brush* et le *matting brush*. Les deux premiers outils servent à consolider le modèle utilisé en intégrant des pixels supplémentaires étiquetés de manière certaine par l'utilisateur. Le troisième outil permet de lisser les contours.

Selon [42], le *GrabCut* dépasse tous les outils précédents. Il existe deux inconvénients à cette méthode. Le premier réside dans le fait qu'elle ne peut sélectionner qu'un seul objet à la fois. Le second est que l'algorithme minimise une fonction de coût global qui ne peut distinguer correctement le bruit de détails fins. Ainsi, l'algorithme est nettement moins performant pour les images contenant beaucoup de détails.

2.2.7 SIOX



(a) Image originale (b) Interactions (c) Résultat

FIG. 2.2: Illustration du principe d'extraction d'objet de SIOX[3]

Nous tenions à évoquer une méthode interactive très intéressante introduite par G. Friedland [43]. Elle fait l'objet du projet SIOX[4]. Le fonctionnement de l'algorithme peut se résumer en 6 étapes principales qui sont les suivantes :

1. Récupération des pixels de confiance représentant le fond signifiés par l'utilisateur (pixels bleutés dans la figure 2.2.b)
2. Récupération des pixels de confiance appartenant à l'objet signifiés par l'utilisateur (pixels vert dans la figure 2.2.b)
3. Création des 2 signatures couleurs de l'objet et du fond par classification des pixels renseignés par l'utilisateur : S_O et S_F
4. Classification des pixels restants de l'image par une recherche pondérée des plus proches voisins dans les ensembles S_F et S_O.
5. Filtrage morphologique
6. Récupération de la composante connexe comportant le plus de pixels de confiance.

Lorsque le résultat ne satisfait pas l'utilisateur, celui-ci peut affiner de manière intuitive la classification initiale et ainsi améliorer la classification automatique.

La méthode de segmentation est adaptée des travaux exposés dans [103] dont l'objectif est d'utiliser la signature couleur et une distance appelée distance d'*Earth Mover* dans le cadre

[4]Sous-projet de E-Chalk dirigé par Raul Rojas développé au département informatique de l'université libre de Berlin (*Freie Universität Berlin*). Pour plus d'informations consulter le site internet *http* : *//www.siox.org/*

de la recherche d'images (*image retrieval*). L'approche de SIOX, résumée par les six étapes précédentes, consiste à construire 2 signatures (étape 3), une pour le fond et une pour l'objet à l'aide des pixels signifiés comme tels par l'utilisateur (étapes 1 et 2). Ces signatures sont ensuite utilisées pour classer les pixels de l'image en deux catégories : fond et objet (étape 4). Pour cela, chaque échantillon du fond et de l'objet est découpé en plusieurs sous-espaces de tailles égales puisque dans l'espace $L^*a^*b^*$, la taille du sous-espace est synonyme d'une certaine cohérence perceptive entre les points contenus. Ce découpage est réalisé par l'algorithme du *kd-tree* [5] modifié par [103]. La règle de séparation est alors simplement la division d'un espace donné en deux sous-espaces de tailles égales (et non pas de séparer l'espace en sa médiane). Le découpage s'arrête lorsque l'espace entre les nœuds devient plus petit que le diamètre permis pour un groupe[5]. Il est alors possible qu'un nœud appartienne à plusieurs groupes. Les nœuds sont donc recombinés en construisant un autre *kd-tree* utilisant uniquement les centroïdes des groupes obtenus dans la phase précédente.

Afin de classer un pixel donné, l'objectif est de parcourir l'arbre afin de savoir s'il appartient à un groupe du fond ou non. Si un pixel n'appartient ni au fond, ni au premier plan, il est affecté à la classe donnant la distance Euclidienne la plus faible entre le pixel et chaque centroïde des groupes de la classe.

L'inconvénient de la méthode est qu'elle ne peut extraire correctement un objet sur un fond dont la distriution de couleurs est proche de celle de l'objet ; même dans le cas où la frontière entre les deux est contrastée. Notons que SIOX dispose d'un *plugin* pour l'application *GIMP*[6]. La figure 2.3 présente un comparatif visuel des méthodes présentées précédemment.

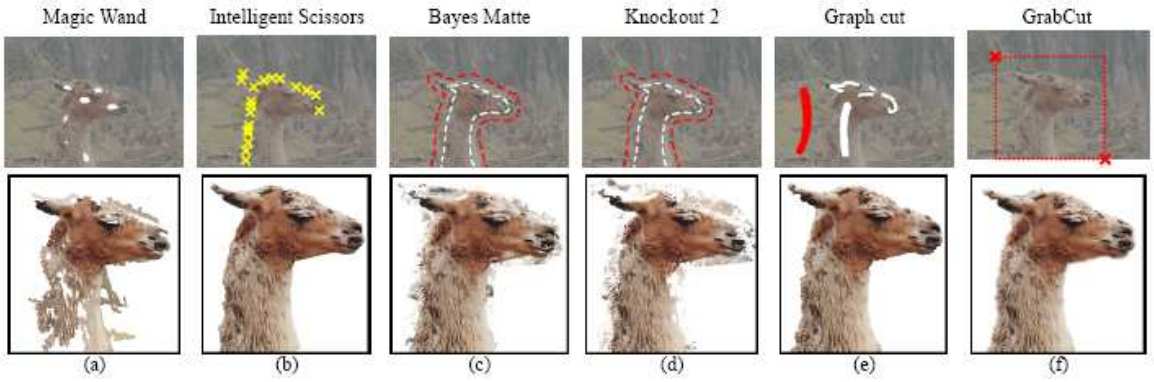


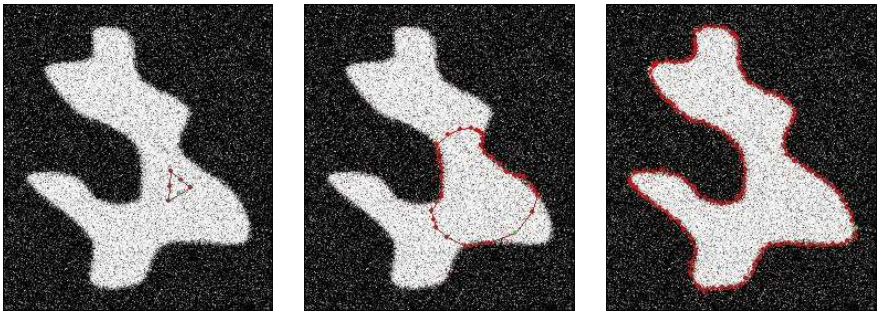
Fig. 2.3: Comparaison de différents outils d'extraction de régions du premier plan (extrait de [102])

2.2.8 Les contours actifs : *snakes*

Il n'était pas possible de réaliser cet état de l'art sans évoquer la famille des méthodes fondées sur les contours actifs également appelées *Snakes*. Initialement proposées par M. Kass *et al.* [125], ce sont les méthodes les plus classiques de détection de contours. Ces méthodes nécessitent une initialisation qui consiste à placer un contour grossier à l'intérieur ou à l'extérieur de l'objet qui par la suite, au cours du traitement, est ajusté au contour de l'objet en minimisant une fonction d'énergie (cf. figure 2.4). Il en résulte un seul contour optimal. C'est une méthode qui a été très largement utilisée dans le domaine de l'imagerie médicale.

[5]Plusieurs tailles de groupes sont utilisées selon les axes L, a et b. Par défaut, 0,66 pour L, 1,25 pour a et 2,50 pour b. Ces valeurs peuvent être modifiées selon la diversité des couleurs perçues sur chaque axe

[6]Logiciel libre de traitement d'image appartenant au projet GNU



(a) Initialisation du contour (b) Evolution du contour (c) Résultat

FIG. 2.4: Évolution d'un *Snake*

La fonction d'énergie

La fonction d'énergie à minimiser combine deux types de forces :

1. Des forces internes qui sont des forces élastiques et qui traduisent d'une part la flexibilité (contrainte de régularité) du *Snake* et d'autre part sa capacité à s'étendre ou à se contracter (contrainte sur le périmètre).
2. Des forces externes intrinsèques à l'image, comme par exemple son gradient.

Les limites des snakes

Le *Snake* traditionnel est très sensible à sa position initale à cause de la non-convexité de la fonction d'énergie qui le fait converger vers des minima locaux. Pour réduire ce problème, plusieurs méthodes ont été adoptées.

Tout d'abord, le *Snake* peut être couplé à des techniques de programmation dynamique (DP : *Dynamic Programming*). Ces dernières ont l'avantage de garantir un minimum global et d'être plus stables numériquement, cependant elles s'accompagnent aussi d'une importante quantité de données, ce qui implique des problèmes de stockage et des temps de calcul élevés.

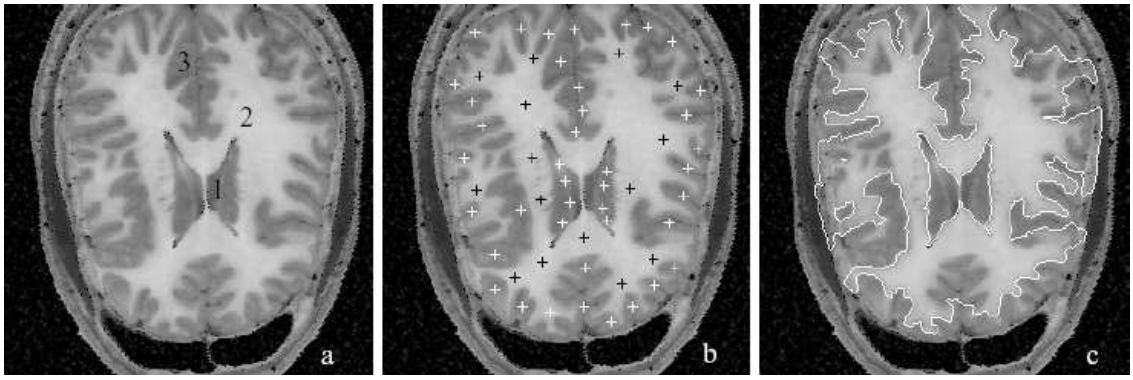
Une autre méthode appelée *Dual-Snake* [45] consiste en l'utilisation de deux contours, l'un se rapprochant du contour de l'objet par l'intérieur et l'autre par l'extérieur. En comparant les deux contours obtenus, il est alors possible de rejeter les minima locaux.

Les *Snakes* traditionnels ont comme autre inconvénient le fait qu'ils ne peuvent pas prendre en charge des changements topologiques du contour initial comme par exemple des coupures ou des fusions. Le *T-Snake* [84] a été développé dans l'optique de répondre à ce problème. C'est une forme discrète du *Snake* classique.

2.2.9 La ligne de partage des eaux interactive

Le principe de la méthode de la ligne de partage des eaux (*Watershed*) [10] est la croissance de régions (*Region growing*) à partir de points d'amorçage, aussi appelés germes, correspondant aux minima du gradient de l'image. Le problème est que généralement une image réelle contient beaucoup moins d'objets sémantiques que de régions homogènes. Pour pallier le problème de la sur-segmentation, J. Cutrona et N. Bonnet ont décidé d'abandonner l'automatisation complète du procédé classique et sont revenus à un procédé interactif [30] où les germes ne sont plus choisis suivant les minima du gradient mais par l'intermédiaire de clics effectués par l'utilisateur sur l'image. Une étiquette correspond à chaque ensemble de germes appartenant

à un objet. A la fin du traitement tous les pixels sont classés suivant l'étiquette des germes de la région à laquelle ils appartiennent (cf. figure 2.5).



(a) Image originale. Les objets d'intérêts sont marqués de 1 à 3

(b) Points d'amorçage étiquetés suivant les différentes régions

(c) Résultat du *Watershed* utilisant les points d'amorçage de (b)

FIG. 2.5: Résultat du *Watershed* (extrait de [30])

Le principal avantage de cette méthode est qu'elle ne nécessite aucun autre paramètre en plus des points d'amorçage. Elle n'utilise notamment aucun seuillage. Cependant, l'initialisation consistant à positionner de nombreux germes par des clics sur l'image, constitue une importante interaction de la part de l'utilisateur sachant qu'il n'est pas toujours évident d'anticiper le résultat.

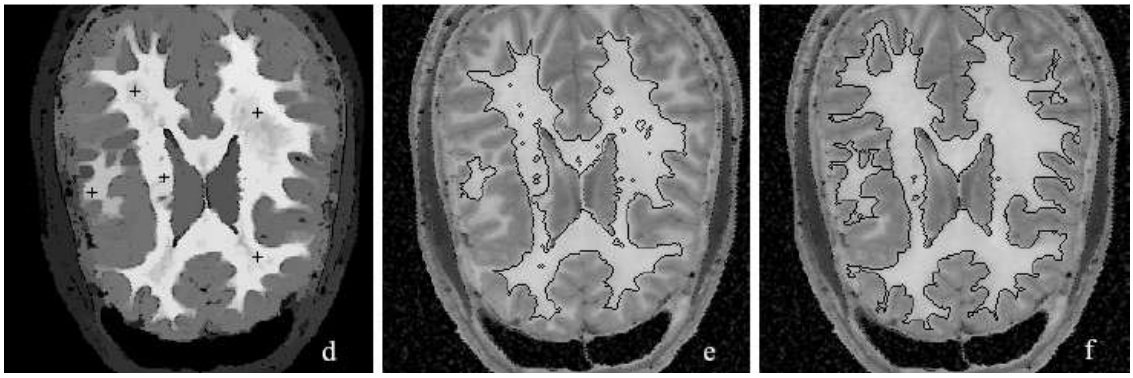
2.2.10 La connexité floue

La notion de la connexité floue a été introduite par Rosenfeld [101], Dellepiane *et al.* [39] et Udupa et Samarasekara [120] qui ont été les premiers à prendre en compte le caractère flou des images naturelles dans un algorithme de segmentation. L'idée générale est de générer une carte des connexités de chaque pixel de l'image originale avec un pixel spécifique de l'objet d'intérêt, désigné manuellement. L'initialisation se fait, là encore, avec quelques clics sur l'image.

Cette technique permet de détecter un seul objet à la fois. Le principal inconvénient de cette méthode est le nombre et la complexité des paramètres que doit gérer l'utilisateur. Notamment pour la création de la carte des connexités. Nous pouvons voir le résultat de segmentation d'un tel algorithme dans la figure 2.6.

Pour simplifier l'utilisation d'une telle méthode J. Cutrona et N. Bonnet [30] ont intégré la notion de *compétition* entre les points d'amorçage du *Watershed* dans leur méthode fondée sur la connexité floue. En effet, en ce qui concerne la vision artificielle, il est naturel de penser qu'une région n'existe pas intrinsèquement mais relativement à une autre région ou par rapport au fond.

Dans [30], les auteurs expliquent comment de ce fait, ils évitent tout seuillage et que les résultats sont généralement améliorés par la prise en compte simultanément de plusieurs objets d'intérêt. Nous pouvons visualiser le type de résultats obtenus par cette méthode sur la figure 2.7.

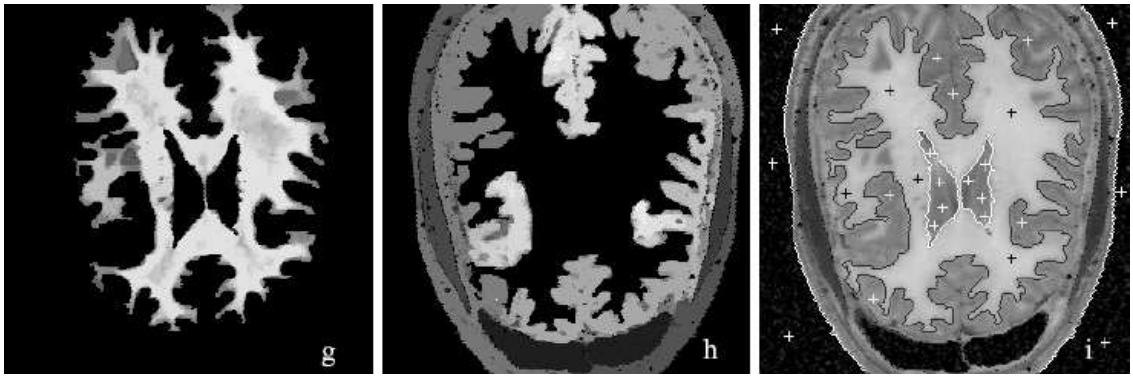


(a) Initialisation avec 5 points d'amorçage

(b) Segmentation en deux régions avec un seuil de 0.75

(c) Segmentation en deux régions avec un seuil de 0.5

FIG. 2.6: Application de la méthode de connexité floue (extrait de [30])



(a) Carte des connexités de deux régions

(b) Carte des connexités de 3 régions

(c) Résultat de la segmentation en 4 régions

FIG. 2.7: Application de la méthode de connexité floue compétitive (extrait de [30])

2.3 Extraction automatique d'objets dans les images fixes

2.3.1 Segmentation couleur et texture

Les méthodes de segmentation classiques telles que les techniques de croissances de régions, de partition de graphes ou de la ligne de partage des eaux sont plus adaptées à la segmentation en régions homogènes en couleur. Cependant, la plupart de images naturelles comportent de nombreuses couleurs et textures. L'analyse en texture nécessite des modèles dont les paramètres ne sont pas toujours simples à estimer. Pour pallier ces problèmes Y. Deng et B. S. Manjunath présentent dans [32], une méthode de segmentation d'image en couleur et en texture. Cette technique permet de limiter grandement le nombre de régions présentes dans la partition finale. L'approche se divise en deux étapes principales. La première consiste à limiter le nombre de couleurs présentent dans l'image. En effet, une image codée sur 24 bits peut contenir des milliers de couleurs, qui sont difficiles à gérer lors de l'étape de segmentation. C'est pourquoi, les auteurs de [32] proposent de réduire ce nombre à 10 ou 20 couleurs à l'aide d'une quantification. Selon eux, ce nombre suffit pour conserver une bonne représentation des

images naturelles. La quantification ne tient pas compte de la répartition spatiale des couleurs et est fondée sur la perception visuelle humaine. Elle permet d'affecter une classe à chaque pixel. La deuxième étape consiste à segmenter l'image quantifiée avec une méthode originale :

Soit P l'ensemble des pixels de l'image quantifiée. Soient p un pixel donné assimilé à ses coordonnées : $p = (x,\ y) \in P$ et g le centre de gravité des pixels de P. Supposons que les couleurs de P ont été quantifiées en C classes P_i avec $i = 1, \ldots, L, \ldots, C$. Soit g_i les centres de gravité des pixels de chaque classe P_i. Le critère de segmentation est défini par l'expression suivante :

$$J = \frac{(S_T - S_W)}{S_W} = \frac{S_B}{S_W} \tag{2.14}$$

Où S_T et S_W sont définis de la manière suivante :

$$S_T = \sum_{p \in P} \|p - g\|^2 \tag{2.15}$$

$$S_W = \sum_{i=1}^{C} \sum_{p \in P_i} \|p - g_i\|^2 \tag{2.16}$$

S_W est la dispersion des positions des pixels au sein de leur classe. J confronte les distances données par S_B, existant entre les classes avec les distances données par S_W, séparant les membres de chaque classe.

Dans le cas d'une image constituée de plusieurs régions homogènes en couleur, les classes couleur sont séparées les unes des autres et la valeur de J est alors élevée. Au contraire, si les classes couleur sont uniformément distribuées dans l'image entière, la valeur de J tend à être faible.

Ce critère est appliqué dans une fenêtre de calcul restreinte à un voisinage pour chaque pixel de l'image quantifiée, afin de générer la *J-image*. Ainsi, une valeur J est attribuée à chaque pixel. Les valeurs élevées et faibles représentent respectivement les contours et les centres des régions. Ensuite, une méthode de croissance de régions effectuée dans la J-image permet de déterminer la segmentation finale. La figure 2.8 présente un aperçu des résultats[7] obtenus avec la méthode J-SEG.

(a) Image originale issue de la séquence *Flowers*

(b) Segmentation

(c) Image originale issue de la base *Corel*

(d) Segmentation

FIG. 2.8: Exemples de segmentation par la méthode J-SEG

2.3.2 Critère photométrique

Certains travaux [82, 130, 94] utilisent les propriétés photométriques des objets pour fusionner les régions qui les composent. Le fait est que des points voisins appartenant à une

[7] Exemples issus de site internet *http : //vision.ece.ucsb.edu/segmentation/jseg/jsegcolor.html*

surface incurvée de façon progressive ont des conditions d'illumination et des orientations de surface similaires. Ces propriétés sont bien sûr capturées dans l'image. S. N. Nayar *et al.* proposent dans [94] une expression du coefficient de réflexion qui traduit le changement d'albédo[8] entre deux pixels voisins. Ce coefficient de réflexion dépend de l'orientation de l'éclairement et du matériau de l'objet.

M. Yazdi et A. Zaccarin proposent dans [130] de reconstruire les objets présents dans l'image avec une méthode fondée sur l'analyse de la compatibilité de forme entre régions utilisant le coefficient de réflexion. Leur méthode nécessite une première étape de segmentation couleur. Celle-ci est réalisée par croissance de régions utilisant les composantes de chrominance H et S de l'espace HSV. La deuxième étape consiste à tester la compatibilité des formes de régions adjacentes en étudiant les variations du changement d'albédo le long de leur frontière. Puisque le coefficient de réflexion est sensible aux changements d'éclairement dus aux variations de formes ou de matériaux, si les deux régions appartiennent au même objet le changement d'albédo calculé le long de la frontière doit fournir une variance faible. Au contraire, dans le cas où les régions n'appartiennent pas au même objet, il est fort probable que l'éclairement dû, par exemple, à une inclinaison différente de la surface engendre une forte variance du coefficient de réflexion. Le test de compatibilité est donc fondé sur la variance du coefficient de réflexion calculé avec les couples de pixels pris de part et d'autre de la frontière des régions à tester. Les auteurs de [130] rajoutent également un test de compatibilité de la surface de l'intérieur des régions en les modélisant à l'aide du profil de l'intensité. Une continuité des profils induit une fusion des régions adjacentes.

2.3.3 Groupement de régions

Les méthodes de groupement de régions sont souvent utilisées pour obtenir des partitions comportant peu de régions. Elles utilisent généralement une partition initiale présentant une sur-segmentation obtenue par une méthode de segmentation classique. La figure 2.9 présente le principe de ce type de techniques.

J. Luo et C.-E. Guo présentent dans [77] une méthode de groupement sans connaissance *a priori* du contenu de l'image, fondée sur la pertinence perceptive des régions. La méthode utilise un modèle probabiliste définissant les régions comme un champ aléatoire de Markov. Les auteurs définissent une série de fonctions de coût rendant compte de la pertinence, d'une part des régions de la partition et d'autre part des groupements entre régions adjacentes. Les critères de pertinence des régions sont fondés sur la convexité, la compacité, l'aire et la variance de la couleur. Les critères de fusion entre deux régions sont évalués, quant à eux, sur la différence des couleurs moyennes, l'amplitude du contour de la frontière entre les régions, la variance de la couleur du groupement et la continuité des contours.

Le problème est formalisé de la manière suivante. Soit R un ensemble de régions :

$$R \;=\; \{R_i, i = 1, 2, \ldots, N\} \tag{2.17}$$

Soit θ l'ensemble des paramètres utilisés. Soit $\hat{R}$ un ensemble de régions résultant du groupement des régions de R :

$$\hat{R} = \{R_j, i = 1, 2, \ldots, M\} \;\; M < N \tag{2.18}$$

A partir d'une partition R^{obs}, l'objectif est d'estimer le meilleur groupement $\hat{R}$ vérifiant :

$$\hat{R} = \underset{\hat{R}}{\operatorname{argmax}}\, p(\hat{R}|R^{\text{obs}}; \theta) \tag{2.19}$$

[8] L'albédo est le rapport de l'énergie lumineuse réfléchie par une surface sur l'énergie lumineuse incidente

Les régions étant modélisées par un champ aléatoire de Markov, on peut adopter la formulation de Gibbs. Ainsi :

$$p(\hat{R}|R;\theta) = \prod_{j=1}^{M} p(\hat{R}_j|R_{j1}, R_{j2}, \ldots, R_{jn};\theta) \tag{2.20}$$

D'où :

$$p(\hat{R}|R;\theta) = \frac{1}{Z} \cdot exp(\sum_{j=1}^{M} E(\hat{R}_j|R_{j1}, R_{j2}, \ldots, R_{jn};\theta)/T) \tag{2.21}$$

Avec $E(\hat{R}_j|R_{j1}, R_{j2}, \ldots, R_{jn};\theta)$, la fonction d'énergie concernant le groupement j et dépendant des paramètres θ. Z est le coefficient de normalisation et T la température. Pour plus de détails sur la construction des fonctions d'énergie, nous invitons le lecteur à se reporter à l'article [77].

A. Pardo propose de combiner plusieurs méthodes afin de répondre au problème de la segmentation sémantique d'image. Il utilise dans [98] une méthode de groupement de régions organisées dans un BPT (*Binary Partition Tree*) [105] fondée sur une métrique perceptive introduite par [97]. Cette métrique permet de fusionner et de hiérarchiser les régions en évaluant leur pertinence visuelle reposant sur plusieurs critères tels que la compacité, le contraste, *etc.* De surcroît, il tire partie du BPT qui offre une intéressante représentation des données et permet une décomposition sémantique des objets.

(a) Image originale (b) Segmentation initiale (c) Résultat du groupement (d) Image originale (e) Segmentation initiale (f) Résultat du groupement

FIG. 2.9: Principe de groupement de régions (extraits de [77])

2.3.4 Opérations morphologiques

L'inconvénient des opérateurs morphologiques classiques tels que les ouvertures ou les fermetures par un élément structurant, sont la dégradation des contours présents dans l'image. Afin de répondre à ce problème, P. Salembier *et al.* présentent dans [104, 106] une méthode fondée sur une famille d'opérations de morphologie appelée *opérateurs connexes*. Ces opérateurs permettent de filtrer le signal image en fournissant uniquement des fusions de zones plates. Ceci permet de ne pas introduire de contours supplémentaires. En effet, une zone plate ne peut pas être déformée ou scindée. Ces opérateurs agissent indépendamment sur chaque composante connexe d'une image binaire. L'avantage de ce type d'outils est qu'il permet l'élimination des composantes connexes qui seraient également supprimées par une simple érosion tout en maintenant intactes les autres composantes connexes. Cette approche se révèle donc

particulièrement adaptée aux applications nécessitant la conservation des contours importants de l'image. L'objectif est la simplification de l'image utilisant différents critères comme la taille, le contraste, la forme des composantes connexes tout en préservant les contours.

2.4 Extraction automatique d'objets dans les séquences vidéo

2.4.1 Différence d'images

Dans le cas d'une vidéo issue de caméra fixe, il est possible d'utiliser une famille de techniques très connue fondée sur la différence entre 2 images. La première approche est l'utilisation d'une image de référence. La deuxième effectue la différence entre 2 images successives.

Image de référence

Une première famille de méthodes d'extraction d'objet par différence d'images regroupe les techniques fondées sur l'utilisation d'une image de référence notée I_{ref}. L'objectif est d'acquérir ou de construire une image donnant une reconstitution du fond la plus fidèle possible sans objet en mouvement. Il est alors possible, par différence pixel à pixel entre l'image du fond et l'image comportant l'objet en mouvement, d'extraire l'objet.

Dans le cas où l'acquisition du fond sans objet en mouvement est rendue possible, l'obtention de l'image de référence est relativement simple. Se pose alors tout de même le problème des variations d'éclairement dans les environnements non contrôlés et donc la nécessité de mettre à jour I_{ref}. Le problème de la mise à jour de I_{ref} peut se modéliser de la manière suivante :

$$I_{ref}(x,\ y,\ t+1) = \alpha \cdot I(x,\ y,\ t) + (1-\alpha) I_{ref}(x,\ y,\ t) \tag{2.22}$$

Où α représente le paramètre qui traduit le compromis de mémoire (ou d'apprentissage) dans la construction de l'image de référence.

Dans le cas où I_{ref} ne peut être acquis directement il faut l'estimer. Les méthodes les plus classiques sont l'intégration temporelle donnée par l'équation 2.23 ou le remplacement de l'objet mobile [53].

$$I_{ref} = \frac{1}{N} \sum_{i=1}^{N} I_i \tag{2.23}$$

Une approche différente introduite par C. Stauffer *et al.* dans [111], consiste à modéliser les différents états d'un pixel d'une image au cours de la séquence par un mélange de Gaussiennes. Pour cela, l'historique des niveaux de gris ou des couleurs de chaque pixel de l'image, représenté sous forme d'histogramme, est modélisé par un mélange de Gaussiennes mis à jour au cours de la séquence. La composante de poids le plus fort du mélange est considérée comme représentative de l'état *fond* et les autres composantes modélisent chacune un état *objet*. L'image de référence est alors reconstruite et mise à jour à chaque image en affectant la valeur moyenne des Gaussiennes de poids le plus fort en chaque pixel. Des traitements morphologiques sont nécessaires après le calcul de la différence d'images pour définir une composante connexe à partir des pixels considérés comme appartenant à l'objet.

Ce type de méthode est très utilisé dans la surveillance de trafic routier. Un des avantages de cette approche est le temps de calcul très faible rendant ainsi possible des applications temps réel pour des technologies embarquées.

Différence d'images successives

Dans le cas où la construction d'une image de référence est impossible, on a souvent recours à la technique de différence d'images successives. La différence pixel à pixel entre deux images

successives permet d'isoler les changements temporels dans la vidéo. Cette méthode repose sur l'hypothèse selon laquelle les variations du fond sont de faibles amplitudes. Soit $I_t(x,\ y)$ l'intensité du pixel de coordonnées $(x,\ y)$ dans l'image considérée au temps t. La différence se calcule donc de la manière suivante :

$$D(x,\ y,\ t) = |I_t(x,\ y) - I_{t-1}(x,\ y)| \tag{2.24}$$

Les valeurs de D en chaque pixel permet de construire l'*image des différences*. A partir de cette image, il est possible de générer une carte binaire B définie comme suit :

$$B(x,\ y,\ t) = 1\ si\ D(x,\ y,\ t) > seuil \tag{2.25}$$

$$B(x,\ y,\ t) = 0\ si\ D(x,\ y,\ t) \leq seuil \tag{2.26}$$

Le seuil est défini empiriquement afin de s'adapter à la vidéo traitée. Les changements temporels dus aux mouvements des objets définissent donc 3 types de zones auxquelles est assignée dans B la valeur 1 et qui sont :

1. la zone de glissement : intersection des zones recouvertes par l'objet au temps t et $t-1$.
2. la zone de découvrement (ou écho) : zone de l'image occupée par l'objet uniquement à $t-1$.
3. la zone recouverte : zone recouverte par l'objet uniquement à t.

Les zones d'écho n'appartiennent pas à l'objet et ne doivent donc pas être prises en compte. Ainsi la reconstruction de l'objet est réalisée en effectuant un ET logique entre deux cartes binaires de changements temporels successives : $B(x,\ y,t)$ & $B(x,\ y,t+1)$. L'objet est alors défini uniquement par la zone de glissement et la zone recouverte.

La gestion des variations d'éclairement peut améliorer la détection des changements temporels. On suppose ici que les variations d'éclairement sont lentes et de faibles amplitudes. Une première méthode consiste à normaliser les niveaux de gris des images et la seconde utilise les propriétés des dérivées partielles :

- Normalisation des niveaux de gris :
  La prise en compte des caractéristiques statistiques des 2 images I_t et I_{t+1} permet de limiter les erreurs dues aux variations de changement d'éclairement entre ces 2 images. La normalisation I_t^N de l'image I_t est donnée par l'expression suivante :

$$I_t^N(x,\ y) = (I_t(x,\ y) - \mu_t) \cdot \frac{\sigma_{t+1}}{\sigma_t} + \mu_{t+1} \tag{2.27}$$

- Méthode des dérivées partielles :
  Considérons un modèle linéaire *a priori* de la variation de l'éclairement sur une région R :

$$I_t^R(x,\ y) = a + b\Delta x + c\Delta y \tag{2.28}$$

  L'utilisation des dérivées partielles spatiales dans le calcul de l'image des différences permet d'éliminer les variations de l'éclairement affectant la région R. Seules les variations de l'intensité dues aux mouvements des objets sont détectées :

$$F_t(x,\ y) = \frac{\partial I_t(x,\ y)}{\partial x} + \frac{\partial I_t(x,\ y)}{\partial y} \tag{2.29}$$

$$D(x,\ y) = \sum_{x,y \in R} F_t(x,\ y) - F_{t+dt}(x,\ y) \tag{2.30}$$

L'inconvénient de ce type de méthode est sa grande sensibilité au bruit. Ses performances sont bonnes lorsque l'objet à extraire est animé de mouvements rapides. Cependant, pour des objets lents il est préférable qu'ils soient texturés. En effet, si l'objet contient des zones homogènes, le problème d'ouverture se pose alors et l'image des différences risque de contenir des valeurs faibles dans les zones de recouvrement (car similaire d'une image à l'autre). L'objet n'est alors extrait que partiellement et des trous apparaissent dans le masque binaire le représentant.

Afin de détecter des objets homogènes, une technique consiste à utiliser les images de gradient obtenues en appliquant un filtre sur les images originales (Prewitt, Sobel...). L'image des différences est alors calculée à partir des images de gradient [64].

2.4.2 Construction de mosaïques

Dans le cas de mouvements panoramiques ou de travelling (latéral ou vertical) de la caméra, certaines techniques permettent d'estimer le fond et ainsi de discriminer les objets présents dans le premier plan. Ces techniques reposent sur la création de mosaïques [107]. Le procédé s'effectue généralement en deux étapes [121] : (1) calcul du modèle du mouvement global entre deux images successives et (2) composition des images en une seule image panoramique selon les paramètres estimés de la caméra. La prise en compte des informations contours permet de renforcer la détection des objets du premier plan [52].

(a) Image originale (issue de la séquence *nhkvideo7*)

(b) Reconstitution du fond

(c) Soustraction de l'image originale avec la mosaïque

Fig. 2.10: Extraction d'objets par construction d'une mosaïque (extrait de [52])

2.4.3 Segmentation de mouvement

La segmentation de mouvement regroupe les techniques fondées sur l'estimation des paramètres du mouvement calculés à partir des informations couleur et/ou de luminance. Chaque image est découpée en régions homogènes en mouvement. Plusieurs approches sont utilisées afin de parvenir à cette segmentation : la fusion [15], le test Bayésien [16], l'agrégation autour de centres mobiles [3]. Les méthodes de segmentation de mouvement dépendent largement de la qualité de l'estimation de mouvement. Ainsi, elles sont soumises aux problèmes classiques d'occultation et d'apparition qui ont pour effet de dégrader la qualité des contours des objets. Certaines méthodes traitent donc conjointement l'estimation et la segmentation [118]. De plus, les méthodes d'estimation du mouvement sont soumises au paradoxe selon lequel : pour obtenir un champ de vecteurs précis, il est intéressant de connaître les frontières du mouvement mais cette localisation n'est possible qu'à partir d'une segmentation du champ de vecteurs. Ainsi, les auteurs de [127] proposent d'effectuer une sélection judicieuse des pixels et des critères servant à estimer de manière précise le champ de vecteurs.

Un inconvénient des méthodes de segmentation de mouvement réside dans le fait qu'un objet est souvent représenté par un ensemble de régions cohérentes en mouvement mais animées de mouvements affines différents. Afin de gérer ce problème, les auteurs de [128] proposent d'utiliser des informations cette fois spatiales, afin de regrouper les régions adjacentes appartenant aux mêmes objets.

Les objets en mouvement sont souvent caractérisés par un mouvement distinct de celui du fond, ce qui rend le critère du mouvement très utile dans l'extraction d'objets vidéo sémantiques. La compensation de mouvement permet de discriminer les objets appartenant au premier plan [53]. Cette compensation de mouvement repose sur l'estimation d'un modèle de mouvement global. Il existe de nombreux modèles de complexités différentes permettant de gérer plus ou moins de mouvements de caméra [51]. Généralement, une simple compensation de mouvement ne permet pas une extraction précise des contours de l'objet du premier plan.

2.5 Conclusion

Pour conclure nous pourrions noter qu'à part les méthodes interactives qui gèrent relativement bien un grand nombre de cas, les méthodes automatiques ne répondent que partiellement aux problèmes inhérents à l'extraction d'objets dans les images naturelles. Bien que le phénomène de sur-segmentation soit atténué, il apparaît de manière évidente que la décomposition de l'image en objets sémantiques est loin d'être effective. Les méthodes spatio-temporelles fondées sur la différence d'images fournissent des résultats de grande qualité. Cependant, les performances diminuent grandement dans le cas de séquences vidéo issues de caméras mobiles. Malheureusement, c'est ce type de vidéos qui est le plus utilisé dans le domaine du multimédia.

D'après cet état de l'art, nous retiendrons quelques points pour la suite de notre étude :

- la notion de conservation des contours des objets d'intérêt au cours de la simplification de l'image apparaît cruciale. C'est un caractère que nos méthodes présentées par la suite se devront de respecter.
- les méthodes de groupements semblent attirantes de part le nombre restreint de régions qu'elles proposent dans la partition finale. Cependant, il paraît inévitable que la qualité du résultat final dépende fortement de la qualité de la partition initialisant le procédé de groupement.
- l'information du mouvement et notamment la séparation du mouvement du premier plan par rapport au mouvement global apporte des informations essentielles pour l'extraction de régions d'intérêt à caractères sémantiques. Le problème qui se pose alors est la localisation précise des contours.

Chapitre 3

Extraction de régions d'intérêt par segmentation locale

Sommaire

Ce chapitre présente plusieurs méthodes de segmentation fondée sur les objets reposant sur la base d'une même méthode de segmentation appelée *segmentation locale*. Le but de ce chapitre est de sensibiliser le lecteur au problème complexe de l'extraction d'objets sémantiques dans une image sans connaissance *a priori* sur le contenu de celle-ci. Ainsi, la distinction entre extraction de régions d'intérêt et extraction d'objets est cruciale et sera faite aussi souvent que nécessaire. A chaque fois que l'expertise sémantique de l'utilisateur sera intégrée dans le procédé de segmentation, il sera alors question d'extraction d'objets d'intérêt. Tandis que, dans le cas d'un procédé totalement automatisé exempt de supervision, il sera question de partitionnement de l'image en régions d'intérêt ou d'extraction de régions d'intérêt.

Dans un premier temps, une approche interactive est étudiée, permettant de mettre en évidence l'efficacité de la segmentation locale lorsqu'elle est initialisée manuellement par un utilisateur. On parlera donc d'extraction d'*objets* d'intérêt. L'expertise essentielle de l'homme permet de localiser dans l'image les informations sémantiques. Le rôle des méthodes développées sur la base de la segmentation locale est de simplifier et de limiter les interactions de l'utilisateur tout en fournissant un résultat précis.

Dans un deuxième temps, ce chapitre traite des méthodes automatiques occultant toute intervention de l'utilisateur. Le problème réside alors dans l'évaluation de la pertinence visuelle voire sémantique des régions d'intérêt constituant les partitions obtenues.

La fin du chapitre étend le problème aux séquences vidéo. Ceci permet d'introduire l'information temporelle et de montrer son importance cruciale dans une telle méthode fondée sur les objets où la sémantique des résultats est un point central.

Mais tout d'abord, présentons la méthode de segmentation locale par pyramide de graphes irrégulière.

3.1 La pyramide de graphes irrégulière

Dans cette section, je présente l'outil générique de segmentation en régions largement utilisé dans mes travaux[a].

[a] Ce type de cadre servira tout au long des chapitres relatant mon travail à introduire l'idée générale des paragraphes. L'idéogramme juxtaposé à ces cadres se traduit par *idée générale*

La pyramide de graphe [90, 9] est une technique utilisée à GIPSA-Lab (DIS) pour faire de la segmentation spatiale d'images couleur. Une version plus évoluée permet de localiser le traitement de segmentation sur des zones bien précises de l'image afin de limiter la sur-segmentation des partitions obtenues. Elle peut être également étendue à un traitement spatio-temporel afin d'extraire et de suivre un ou plusieurs objets d'intérêt dans les séquences vidéo [41]. Cette pyramide décrit sous la forme d'une arborescence de graphes, les adjacences entre régions qui composent une image ainsi que les attributs de chacune de ces régions. La représentation par graphe est très adaptée aux relations d'adjacence qui relient les régions.

Comme le montre la figure 3.1, l'objectif principal de la pyramide irrégulière est de regrouper des régions voisines considérées comme similaires selon certains critères. Ces fusions sont effectuées d'une part, en parallèle et d'autre part, de manière itérative d'un étage à l'autre de la pyramide. A la base de la pyramide constituée par l'image d'origine, aucun pixel n'est encore regroupé. Il existe alors autant de régions que de pixels, puis plus on gravit les étages plus ce nombre de régions diminue par fusion de ces dernières comme le montre la figure 3.2.

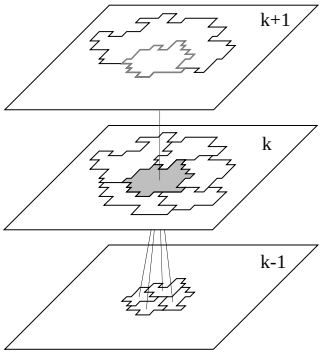


FIG. 3.1: Principe de la pyramide (d'après [6])

Les fusions sont effectuées jusqu'à ce que soit atteint l'apex[1] où le nombre de régions est alors minimal selon les critères choisis.

Le critère de similarité actuellement utilisé est orienté sur la couleur. Chaque pixel est localisé dans l'espace couleur YUV (voir annexe A) par 1 composante de luminance Y et 2 composantes de chrominances U et V. Dans le cas où la pyramide irrégulière est utilisée dans les vidéos, le format utilisé est le format YUV 4 : 2 : 0. Ceci traduit le fait que les composantes de chrominances sont sous-échantillonnées dans toute l'image à la hauteur d'un pixel sur deux, en ligne et en colonne. Ainsi, il existe 4 échantillons de luminances pour 1 échantillon de chaque chrominance. Ce sous-échantillonnage est justifié par une particularité du système visuel humain qui se trouve être plus sensible aux informations de luminance qu'à celles de la couleur.

L'utilisation d'un autre espace couleur est bien sûr envisageable et dépend de l'application visée (cf. annexe A). Cependant, l'utilisation de l'espace YUV a déjà fait ses preuves pour la segmentation d'images naturelles [40]. C'est pourquoi, notre travail ne s'est pas porté sur le choix de l'espace couleur le plus approprié pour la segmentation.

FIG. 3.2: Regroupement des régions d'étage en étage

Les paragraphes suivants présentent le fonctionnement de la pyramide irrégulière utilisée

[1] L'apex est le niveau le plus élevé de la pyramide. Ce terme est préféré au terme de *sommet* qui est utilisé par la suite pour décrire les nœuds d'un graphe

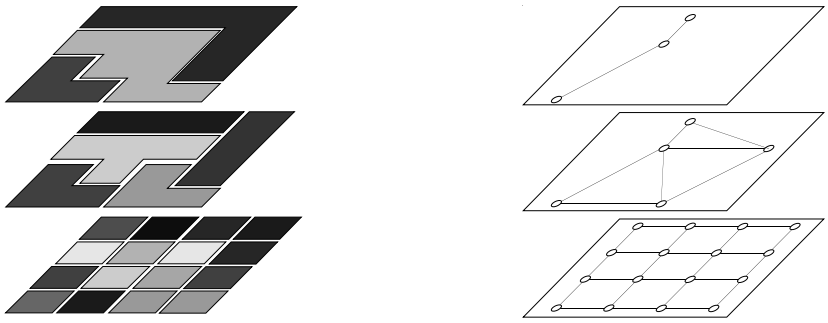
dans une application de segmentation d'image couleur. Pour plus de détails voir [6].

3.1.1 Principes de la pyramide irrégulière

La pyramide irrégulière répond à plusieurs critères caractéristiques des structures pyramidales qui sont les suivants [6] :

- Chaque niveau de la pyramide représente l'image originale à une certaine résolution. La résolution décroît de la base qui représente l'image originale vers l'apex.
- Le nombre de régions de la base correspond au nombre de pixels de l'image originale.
- Le niveau $k+1$ est construit uniquement à partir du niveau k.
- Chaque niveau est représenté par un graphe d'adjacence et une partition de l'image. Chaque région est modélisée par un sommet du graphe d'adjacence.
- Tous les niveaux de la pyramide ont la taille de l'image originale. La différence de résolution résulte de la réduction du nombre de régions du niveau.
- Les traitements des pixels sont locaux et effectués en parallèle sur tout le niveau.
- Une région à un niveau k distinct de la base et de l'apex, possède un nombre irrégulier non nul de voisins appartenant au même niveau, un parent unique au niveau $k+1$ et un nombre quelconque d'enfants au niveau $k-1$.

La composante connexe représentée par un sommet à un niveau donné est appelée *champ récepteur* (voir figure 3.3). Afin d'adapter la représentation de la pyramide irrégulière à une application de type segmentation, l'ensemble des pixels appartenant à un champ récepteur se voit attribuer la même étiquette ou une couleur moyenne (dans l'espace YUV) calculée à partir du contenu de cette composante connexe. L'ensemble des champs récepteurs constitue alors une partition de l'image d'origine.



(a) Pyramide de champs récepteurs : 16, 5 puis 3 régions

(b) Pyramide de graphes d'adjacences des régions.

FIG. 3.3: Correspondance entre les champs récepteurs et les sommets des graphes d'adjacences (d'après [6])

3.1.2 Structure de données

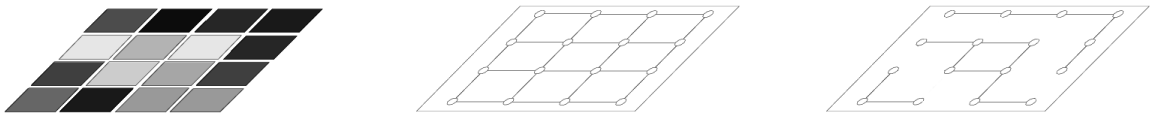
Classiquement, l'image d'origine entière est utilisée afin de construire le graphe d'adjacence. Ainsi, au départ, chaque pixel est représenté par un sommet. Au fur et à mesure des fusions et de la création des différents étages, les sommets représentent des régions par l'intermédiaire du champ récepteur. Les attributs descriptifs associés à chaque sommet (ou région) sont paramétrables à volonté. Il est tout à fait possible d'en ajouter selon l'application désirée.

Voici les plus classiques dont les 4 premiers sont indispensables :

- Nombre de voisins
- Lien vers chacun des voisins
- Booléen indiquant si le sommet a au moins un voisin survivant
- Booléen indiquant si le sommet est survivant
- Couleur moyenne dans l'espace YUV
- Coordonnées du sommet
- Surface
- Seuil local de similarité (présenté dans le paragraphe 3.1.4)
- Écart-type des niveaux de gris

3.1.3 Construction de la pyramide

La base de la pyramide constitue une relation d'adjacence de chaque pixel. De par la constitution d'une image, il est naturel d'utiliser un voisinage du type 4-connexité voire 8-connexité entre les pixels répartis en maille carrée. Le niveau zéro est un graphe d'adjacence non orienté qui contient donc autant de sommets que de pixels. Un arc entre deux sommets indique que les pixels sont voisins. Les niveaux supérieurs sont obtenus par des traitements parallèles qui ont pour but d'affecter un sommet non plus à chaque pixel mais à des champs récepteurs recouvrant un ensemble de pixels. La conséquence immédiate étant la diminution du nombre de sommets du graphe. Cette diminution du nombre de sommets est effectuée d'une part selon une notion d'adjacence et d'autre part en utilisant la similarité des voisins. Deux sommets sont déclarés similaires si leur distance dans l'espace couleur YUV vérifie un certain critère. Ce critère permet de construire un *graphe de similarité* à partir du graphe d'adjacence (voir figure 3.4). Ainsi, le graphe de similarité contient le même nombre de sommets que le graphe d'adjacence. Cependant, chaque sommet se voit lier à un voisin uniquement si ce dernier vérifie le critère de similarité développé au paragraphe 3.1.4. Le graphe de similarité est donc un ensemble de sous-graphes du graphe d'adjacence.



(a) Champ récepteur (b) Graphe d'adjacence (c) Graphe de similarité

FIG. 3.4: Comparaison du graphe de similarité avec le graphe d'adjacence (d'après [6])

Voici les différents traitements effectués en parallèle à un niveau donné k :

1. Construction du graphe de similarité. Ce graphe détermine quels sont les voisins qui sont susceptibles de fusionner avec un sommet s, selon un seuil de similarité $T_l(s)$ qui lui est propre (voir paragraphe 3.1.4 pour plus de détails sur le calcul du seuil $T_l(s)$).
2. Décimation des sommets du graphe de similarité à l'aide d'une méthode itérative. Lorsque le nombre de sommets survivants reste constant entre deux itérations, le procédé de décimation est stoppé.
3. Gestion des sommets survivants dont l'objectif est de rattacher les sommets non survivants aux sommets survivants forcément voisins.
4. Mise à jour des attributs de chaque sommet survivant à partir des caractéristiques de leur nouveau champ récepteur contenant les régions absorbées.

5. Construction du nouveau graphe d'adjacence qui traduit le voisinage entre les sommets survivants.

3.1.4 Construction du graphe de similarité

Dans le cadre de la segmentation d'image par pyramide irrégulière, la décimation n'a pas lieu dans le graphe d'adjacence mais dans un graphe de similarité. Ce dernier permet de prendre en compte les caractéristiques de l'image telles que la luminosité ou la couleur de manière globale et locale.

Dans ce graphe un sommet s_i est relié à un autre sommet s_j si s_i et s_j vérifient un certain critère de similarité. Ce dernier utilise deux seuils : un seuil local $T_l(s)$ dépendant du sommet s et un seuil global T_g.

Seuil global de similarité

Le seuil global de similarité T_g permet de définir la distance maximale pouvant séparer deux sommets similaires dans l'espace couleur YUV. Selon ce critère global, les sommets s_i et s_j sont déclarés dissemblables si la distance entre les composantes moyennes dans l'espace YUV est plus élevée que le seuil T_g. Ce dernier est fixé par l'utilisateur. Ainsi, deux sommets s_i et s_j ne pourront être reliés par un arc dans le graphe de similarité s'ils vérifient la relation suivante :

$$\delta(s_i,\ s_j) > T_g \tag{3.1}$$

Où δ est défini comme suit :

$$\delta(s_i,\ s_j) = \sqrt{(Y(s_i) - Y(s_j))^2 + (U(s_i) - U(s_j))^2 + (V(s_i) - V(s_j))^2} \tag{3.2}$$

Ce seuil ne permet pas de prendre en compte les particularités locales de l'image. C'est pourquoi la pyramide irrégulière utilise un seuil calculé pour chaque sommet à partir de son voisinage.

Seuil local de similarité

Les auteurs de [2] insistent sur l'importance de la mise en œuvre d'un seuil de similarité local afin de hiérarchiser au mieux la segmentation selon les particularités de l'image. G. Foret utilise dans [40] un seuil local calculé à partir des niveaux de gris. Nous l'avons ici, étendu à la couleur. Considérons la liste ordonnée des distances couleur δ_i évaluée entre s et ses n voisins ($i \in [\![1,\ n]\!]$). Nous avons donc :

$$\delta_1 \leq \delta_2 \leq \cdots \leq \delta_l \leq T_l(s) < \cdots \delta_t < T_g < \delta_n \tag{3.3}$$

L'objectif est de déterminer le seuil $T_l(s)$ permettant d'ordonner les fusions d'un étage à l'autre de la pyramide de manière à prendre en compte le voisinage de chaque pixel. $T_l(s)$ doit correspondre au pas le plus significatif dans la séquence des distances δ_i avec $i \in [\![1,\ t]\!]$.

Pour cela, il est nécessaire de partitionner en 2 groupes la séquence des distances tout en maximisant l'inertie inter-groupes. Pour chaque voisin j de s, on calcule les inerties suivantes :

$$U_j = \frac{\sum_{i=1}^{j} \delta_i}{j}, \quad V_j = \frac{\sum_{i=j+1}^{t} \delta_i}{t-j}, \quad j \leq t-1 \tag{3.4}$$

Le seuil $T_l(s)$ est alors défini de la manière suivante :

$$T_l = \underset{j}{\operatorname{argmax}}(V_j - U_j) \tag{3.5}$$

Comme $T_l(s)$ est calculé selon les voisins s_i du sommet s, généralement nous avons :

$$\delta(s,\ s_i) \leq T_l(s) \nRightarrow \delta(s,\ s_i) \leq T_l(s_i) \tag{3.6}$$

Ainsi, le seuil local $T_l(s)$ permet de construire un graphe de similarité *orienté*. Ce seuil favorise des fusions localement dans des directions privilégiées. Ce qui se révèle très utile pour la gestion de dégradés. Le seuil global, quant à lui, est utilisé pour contraindre le seuil local à ne pas dépasser une valeur critique. En effet, dans le cas où un sommet n'a que des voisins très différents de lui, le seuil local pourrait prendre une valeur excessive.

En résumé, le sommet s_i est déclaré similaire et peut fusionner avec le sommet s_j au niveau suivant de la pyramide s'il vérifie les 2 propriétés suivantes :

$$\delta(s_i,\ s_j) \leq T_l(s_i) \textbf{ et } \delta(s_i,\ s_j) \leq T_g \tag{3.7}$$

3.1.5 Décimation du graphe de similarité

L'obtention d'un nouvel étage de la pyramide s'obtient par contraction du nombre de sommets présents dans l'étage précédent. Cette étape s'appelle la décimation. Elle intervient dans le graphe de similarité afin de fournir une partition de l'image orientée segmentation couleur. Il en résulte une croissance en parallèle de régions dans la carte des champs récepteurs. Le problème consiste à réduire le nombre de sommets du graphe de façon homogène dans tout l'ensemble, sans favoriser une partie du graphe. Trois règles s'imposent alors d'elles-mêmes :

Règle 1 : *Deux sommets reliés par un arc de similarité bi-directionnel au niveau k ne peuvent pas survivre tous les deux au niveau $k+1$.*

Règle 2 : *Deux sommets reliés par un arc de similarité uni-directionnel au niveau k ne doivent pas* si possible *survivre tous les deux au niveau $k+1$*

Règle 3 : *Au niveau k, tout sommet non-survivant doit posséder au moins un sommet de son voisinage qui survit au niveau $k+1$.*

Les règles 1 et 2 permettent une décroissance significative et répartie du nombre de sommets tandis que la règle 3 garantit que l'ensemble des sommets constitue bien une partition de l'image.

La décimation du graphe de similarité est dite *adaptative* [60, 6] : le choix des survivants n'est pas réalisé de manière stochastique comme il est proposé par Meer et Connelly [86] mais s'adapte au contenu de l'image dans le cadre particulier d'une segmentation. En effet, à un niveau donné de la pyramide, il coexiste des régions de tailles très différentes. Il est donc préférable de conserver les régions de grandes tailles et que ce soient ces dernières qui survivent en absorbant les régions voisines de petites tailles. Chaque sommet se voit donc associer un score qui est proportionnel à l'aire en pixel de la région qu'il représente. Un sommet survit s'il maximise le score par rapport à son voisinage. Dans le cas où plusieurs voisins fournissent le même score local, il en résulte un choix cette fois aléatoire pour décider du survivant. D'autres caractéristiques déterminant le score sont envisageables. Il peut tout à fait être adapté en fonction de l'application dans laquelle est utilisée la pyramide. Toutefois, l'aire des régions semble être un critère bien adapté à une segmentation générique.

3.1.6 Relaxation

Les partitions obtenues avec la pyramide irrégulière comportent généralement une multitude de petites régions parasites au voisinage des contours. Ces dernières sont devenues trop dissemblables de leurs voisines et ne peuvent plus être fusionnées sur le critère classique de similarité. L'objectif de la relaxation est de simplifier la partition en forçant la fusion de ces régions de petites tailles avec l'une de leurs voisines. Cette relaxation peut être appliquée à

l'apex de la pyramide ou pendant la création des niveaux intermédiaires afin d'accélérer le traitement.

Ainsi, une région dont la taille est inférieure à un certain seuil t_{min} fixé par l'utilisateur, est absorbée par la région adjacente la plus similaire en couleur. Ceci intervient pendant la construction du graphe de similarité. Soit s un sommet correspondant à une région dont la taille est inférieure à t_{min} et v_i ses n voisins ($0 < i \leq n$). Un arc est créé entre s et le sommet v_k qui vérifie :

$$k = \min_i \delta(s,\ v_i) \tag{3.8}$$

3.1.7 Pyramide locale

概念

Dans cette section, je présente l'extension de la pyramide irrégulière orientée sur l'extraction de régions d'intérêt.

Il n'est pas rare que les objets d'intérêt n'occupent que très peu de place dans une image. Par exemple dans la figure 3.5, les animaux ne représentent qu'une partie de l'image. Il est intéressant de pouvoir focaliser le procédé de segmentation uniquement sur ces objets d'intérêt, voire sur leur frontière avec d'autres objets ou le fond. Ceci permet de limiter les erreurs, d'accélérer le traitement et de simplifier les résultats en fournissant moins de régions. La localisation des contours des objets peut être réalisée manuellement avec précision ou de manière plus grossière avec une méthode automatique présentée par la suite.

FIG. 3.5: Objets d'intérêt dans une image

Pour effectuer cette segmentation locale, il faut indiquer à la pyramide irrégulière quelles sont les zones indéfinies qui sont à segmenter au niveau pixel et quelles sont celles qui sont supposées appartenir à une entité d'intérêt et qui doivent donc être conservées telles quelles. Pour ceci, nous proposons la construction d'une *pyramide locale* où seul un nombre réduit de pixels de l'image est identifié à des sommets. Ce sont ces pixels qui seront segmentés. Tandis que les n composantes connexes restantes sont identifiées respectivement à n sommets (figure 3.6.a et b). Ces sommets sont appelés les *racines* et représentent des régions qui appartiendront au résultat final tel que le fond par exemple.

Cette optimisation de la segmentation nécessite une étape cruciale d'étiquetage des différentes zones de l'image. C'est l'objectif du paragraphe suivant.

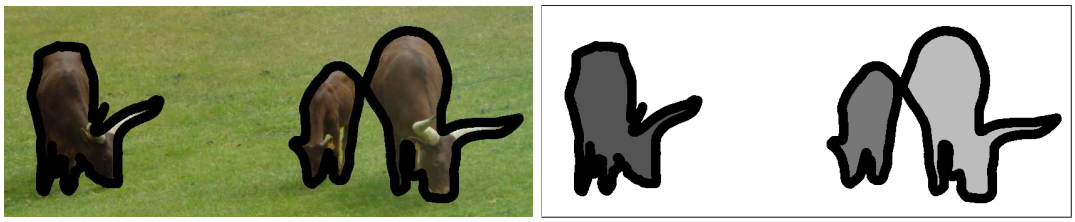
Étiquetage de l'image

Afin de déterminer les zones de l'image qui doivent être segmentées, nous allons avoir recours à l'étape d'étiquetage des pixels dans l'image. Nous considérons ici que la segmentation est focalisée sur les contours des objets d'intérêt de l'image 3.7 qui sont situés sous le ruban de pixels noirs.

(a) Champs récepteurs (b) Graphe correspondant

FIG. 3.6: Exemple d'initialisation d'une pyramide locale

Les pixels des zones délimitées par les rubans portent une étiquette propre à chaque objet. Les pixels extérieurs se voient attribuer une étiquette marquant leur appartenance commune au fond. Chaque composante ainsi définie est associée à une racine. Dans la figure 3.7.b les composantes connexes correspondant aux différentes racines sont affichées en nuances de gris. Les pixels des rubans, quant à eux, constituent les zones indéfinies de l'image qui doivent à l'issue du procédé de segmentation, récupérer les étiquettes *ad hoc* des racines environnantes. Ces zones ne portent donc pas d'étiquette à l'initialisation de la segmentation. Nous allons maintenant étudier la construction de la zone de segmentation à partir de cet étiquetage qui permet d'optimiser le procédé de segmentation et d'effectuer la propagation des étiquettes dans les pixels du ruban.



(a) Initialisation des régions d'intérêt

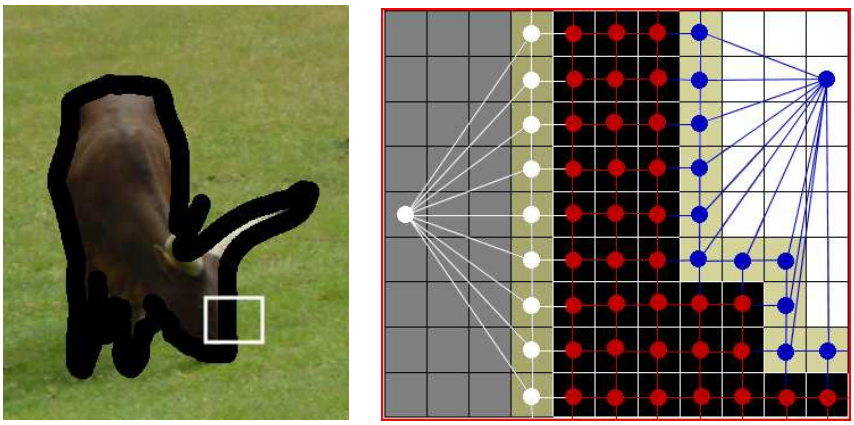
(b) Étiquetage initial des racines induites par (a). Les zones noires n'ont pas d'étiquette

FIG. 3.7: Etiquetage des objets d'intérêt

Étiquetage de la zone de segmentation

Une fois l'étiquetage effectué, nous procédons à la détermination de la zone de segmentation. Le problème est que si la segmentation est limitée uniquement à la zone indéfinie, les régions résultantes pourront difficilement fusionner avec les racines sur le critère de similarité donné par l'équation 3.1. En effet, ce critère est fondé sur des valeurs moyennes faisant intervenir l'intégralité des régions. Ainsi, il n'est pas envisageable de comparer des attributs des régions d'aires faibles avec ceux des racines qui sont généralement d'aires plus imposantes. Il faut absolument respecter une certaine équité entre la taille des régions afin que les critères de similarité entre ces régions soient pertinents.

Pour éviter ce phénomène, une fine couche de pixels de part et d'autre du ruban est séparée des racines et regroupe autant de sommets portant l'étiquette de la racine correspondante (cf. figure 3.8). Ce sont ces deux couches de liaison qui vont permettre de propager les étiquettes des deux racines dans la zone indéfinie. La propagation est obtenue par fusion des pixels étiquetés avec les pixels non étiquetés. Durant la construction de la pyramide locale, les



(a) Le rectangle est zoomé dans (b)

(b) Graphe d'adjacence de la zone marquée dans (a)

FIG. 3.8: De l'initialisation manuelle d'une zone d'intérêt à la définition d'un graphe d'adjacence où sera effectuée la segmentation

fusions s'effectuent toujours selon le critère de similarité couleur, mais répondent également à des règles qui assurent la propagation cohérente des étiquettes à travers la zone indéfinie.

Règles de propagation

Le regroupement de sommets se fait localement selon les phases classiques de la pyramide irrégulière qui sont la décimation suivie de la phase de rattachement (les pixels des sommets disparus se rattachent au sommet voisin survivant le plus similaire). Le nombre de sommets se regroupant en un nouveau sommet n'est pas fixé *a priori*. Les phases de décimation et de rattachement nécessitent que la similarité soit évaluée pour tout couple de sommets. La notion de similarité est capitale ; rajouter de nouvelles contraintes dans les règles de construction de la pyramide peut simplement être vu comme modifier le critère de similarité. Pour prendre en compte les nouvelles contraintes induites par la segmentation locale, on modifie le critère vu au paragraphe 3.1.4 et on rajoute des règles concernant la propagation des étiquettes. Par la suite, nous noterons $R(l)$ une région portant l'étiquette l et R est une région non étiquetée ; $\sim$ est le symbole de similarité.

- **Règle de similarité :** $R_i(l_i) \nsim R_j(l_j)$ si $l_i \neq l_j$ ou si R_i et R_j ne vérifient pas les relations de similarité couleur vues précédemment à l'équation 3.7. Deux régions portant deux étiquettes différentes ne peuvent pas fusionner, ceci pour éviter la fusion de différents objets.
- **Règle de propagation 1 :** Si $R_i \sim R_j(l)$, $R_i \cup R_j(l) \rightarrow R_k(l)$. Une région non étiquetée fusionnant avec une région étiquetée l donne naissance à une région portant l'étiquette l.
- **Règle de propagation 2 :** Si $R_i(l) \sim R_j(l)$, $R_i(l) \cup R_j(l) \rightarrow R_k(l)$. La fusion de plusieurs régions d'étiquette l donne naissance à une région portant l'étiquette l.
- **Règle de propagation 3 :** Si $R_i \sim R_j$, $R_i \cup R_j \rightarrow R_k$. Des régions non étiquetées fusionnant entre elles donnent naissance à une région non étiquetée.

Lorsque plus aucune fusion n'est possible, il se peut qu'il reste tout de même des sommets non étiquetés trop dissemblables par rapport à ses voisins. Deux options se présentent :

1. on force la fusion de ces régions avec la racine voisine la plus similaire en couleur. Ceci

revient à relâcher la contrainte de similarité.

2. ces régions sont conservées telles quelles pour être classées manuellement par l'utilisateur.

Dans une méthode automatique, on préférera la première option. La partition finale comporte alors autant d'objets que de racines. Tandis que dans le cas d'une méthode interactive, c'est la seconde option qui sera privilégiée. La partition finale comporte alors des racines et les régions non étiquetées attendant une classification manuelle ultérieure comme le montre la figure 3.9.

FIG. 3.9: Résultat de la segmentation locale. Les régions non étiquetées sont en vert

Nous venons de présenter le fonctionnement global de la pyramide irrégulière locale. Nous allons maintenant nous intéresser à la manière dont peuvent être localisées les zones de segmentation.

3.2 Initialisation interactive - le cas idéal

概念

Dans cette section, je présente l'utilisation de la pyramide locale dans un contexte interactif. Cette méthode permet de segmenter avec précision les contours d'un objet préalablement localisé grossièrement par l'utilisateur.

La segmentation précise et efficace d'images naturelles en objets d'intérêt est difficile à effectuer automatiquement. C'est pourquoi le recourt à la segmentation interactive est souvent nécessaire. L'objectif est de proposer à l'utilisateur un outil simple et pratique permettant d'obtenir des résultats exploitables. Ce paragraphe présente deux types d'outils utilisant deux initialisations différentes de la pyramide locale irrégulière. Nous avons intégré nos deux outils dans une interface fonctionnelle appelée ExtraK'Obs[2].

3.2.1 Localisation de l'objet d'intérêt - boîte d'extraction

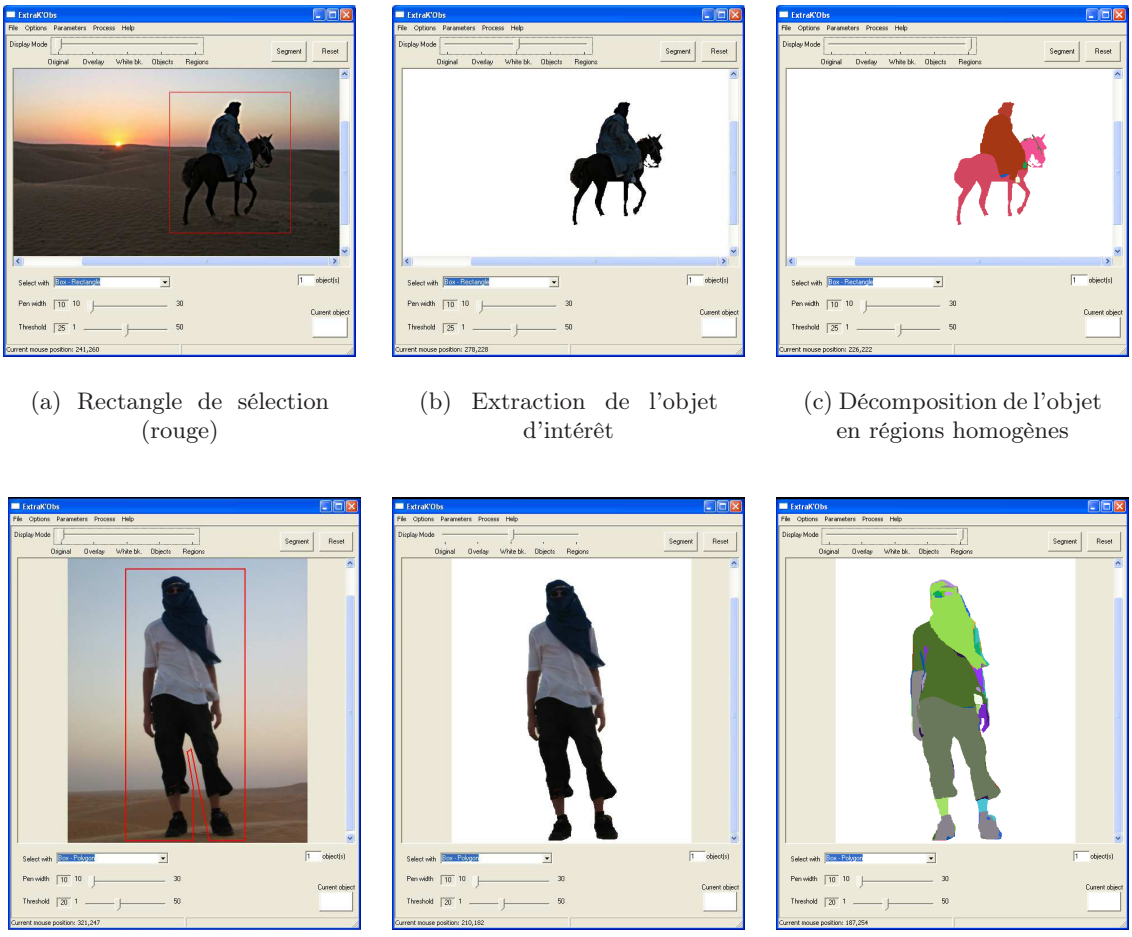
概念

Je présente, ici, notre premier outil interactif permettant de focaliser rapidement la segmentation sur l'objet d'intérêt.

Dans cette première version, l'initialisation est réalisée à partir d'une boîte englobante appelée *boîte d'extraction*, constituant une forme fermée contenant l'intégralité de l'objet à extraire (cf. figure 3.10.a). Les formes classiquement utilisées sont des rectangles, des ellipses mais une forme quelconque peut également être envisagée (cf. figure 3.10.d). Cependant, cette dernière implique une interaction plus importante de la part de l'utilisateur.

Cette initialisation permet de définir tous les pixels se trouvant à l'extérieur de la boîte comme appartenant à la racine représentant le *fond*. Ce dernier est alors associé à un seul

[2] *http : //www.lis.inpg.fr/pages_perso/bertolino/software.php*



(a) Rectangle de sélection (rouge)

(b) Extraction de l'objet d'intérêt

(c) Décomposition de l'objet en régions homogènes

(d) Polygone de sélection (rouge)

(e) Extraction de l'objet d'intérêt

(f) Décomposition de l'objet en régions homogènes

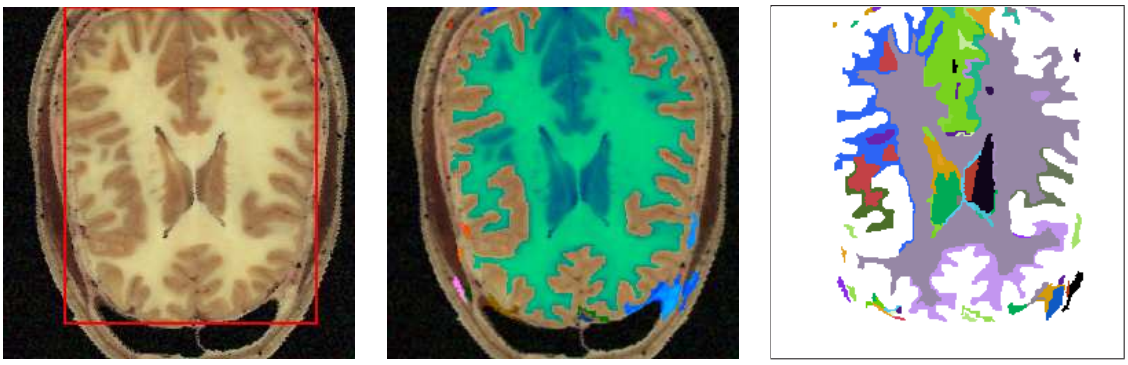
FIG. 3.10: Extraction d'un objet à l'aide de la boîte d'extraction

nœud du graphe de la pyramide. L'intérieur de la boîte est quant à lui considéré comme une *zone indéfinie*. Seule une partie des pixels situés à l'intérieur de cette zone appartient effectivement à l'objet qui doit être extrait. Pour définir cet ensemble il faut segmenter la zone indéfinie. Ainsi, pendant la segmentation, une partie des pixels fusionne avec la racine du fond tandis que le reste des pixels fusionne entre eux afin de générer plusieurs régions homogènes contrastées avec le fond. Ce procédé, en plus d'extraire l'objet (cf. figures 3.10.b et 3.10.e), fournit une segmentation en régions homogènes de l'objet comme le montre les figures 3.10.c et 3.10.f.

Cette méthode donne des résultats de qualité lorsque les objets sont relativement bien contrastés avec le fond. Si cette contrainte n'est pas respectée sur toute la périphérie de l'objet, il se peut qu'il y ait un risque de « fuites » de régions de l'objet vers le fond. La robustesse des résultats est d'autant plus importante lorsque les objets se trouvent sur un fond avec des zones homogènes. Toute zone hétérogène contenue dans la zone de segmentation et appartenant au fond peut, à tort, être considérée comme un objet. Quant aux objets eux-mêmes, ils ont peu de contraintes : ils peuvent être très texturés ou homogènes, cela n'a pas de réel impact sur le résultat.

Comparaison de la boîte d'extraction avec la connexité floue compétitive

Nous effectuons ici la comparaison avec la méthode de la connexité floue compétitive [30]). Nous avons segmenté l'objet central (jaune clair) de l'image 3.11.a en utilisant la boîte d'extraction avec un seuil global de 19. Le temps de calcul pour extraire un objet de cette taille est de l'ordre de la seconde.



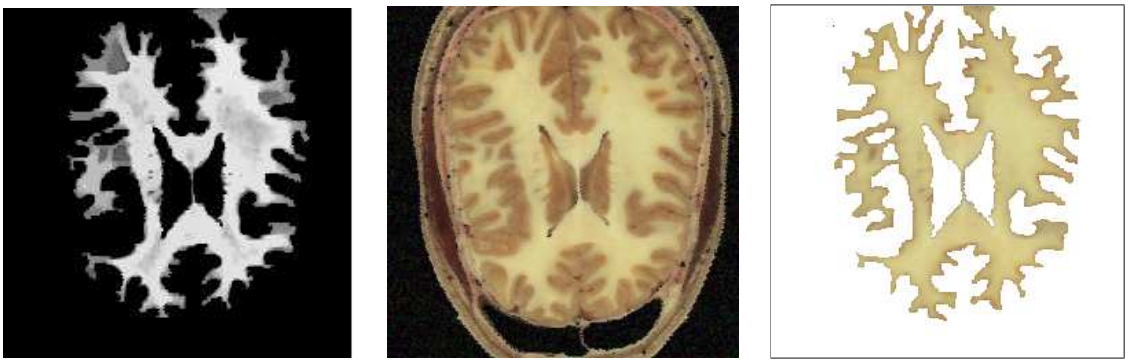
(a) Sélection de l'objet (cadre rouge)

(b) Extraction de l'objet d'intérêt

(c) Décomposition de l'objet en régions homogènes

FIG. 3.11: Exemple d'extraction d'objet pour une application biomédicale

Une brève retouche du masque brut consistant à fusionner avec le fond une dizaine de régions non connexes à l'objet d'intérêt et quelques régions non désirées incluses dans celui-ci, permet d'obtenir un résultat de qualité. La figure 3.12 permet de comparer ce dernier au résultat obtenu par la méthode de la connexité floue compétitive.



(a) Méthode de la connexité floue compétitive

(b) Image originale

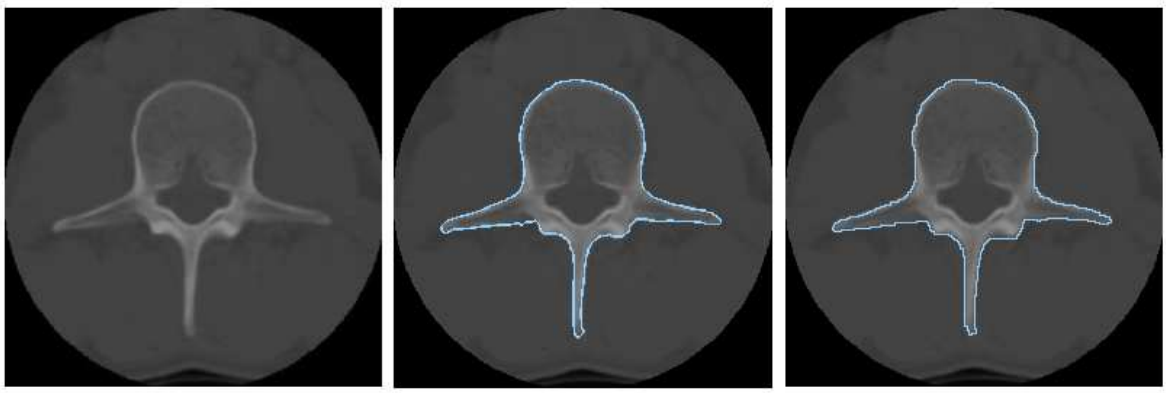
(c) Méthode de la pyramide locale

FIG. 3.12: Connexité floue compétitive *vs* pyramide locale

La sélection de la zone de segmentation est réalisée sans difficulté et de surcroît très rapidement. La décomposition de l'objet en régions homogènes permet de manipuler des régions incluses dans le masque d'extraction. Ainsi, il est possible après le traitement de classer manuellement des régions de manière précise afin d'obtenir une partition sémantique de qualité.

Comparaison de l'outil de la boîte d'extraction avec la méthode des ciseaux intelligents

Nous allons une fois de plus nous intéresser à une application d'imagerie médicale mais cette fois pour comparer la pyramide locale initialisée par la boîte d'extraction avec les ciseaux intelligents (ou *intelligent scissors*) [91]. Pour effectuer la comparaison, nous nous intéressons ici à la localisation du contour de l'objet. Nous avons donc extrait le contour à partir de la région obtenue avec la boîte d'extraction. Les résultats sont regroupés dans la figure 3.13.



(a) Image d'une vertèbre

(b) Contour obtenu avec les ciseaux intelligents (d'après [91])

(c) Contour obtenu avec la boîte d'extraction

FIG. 3.13: Comparaison des ciseaux intelligents avec la boîte d'extraction

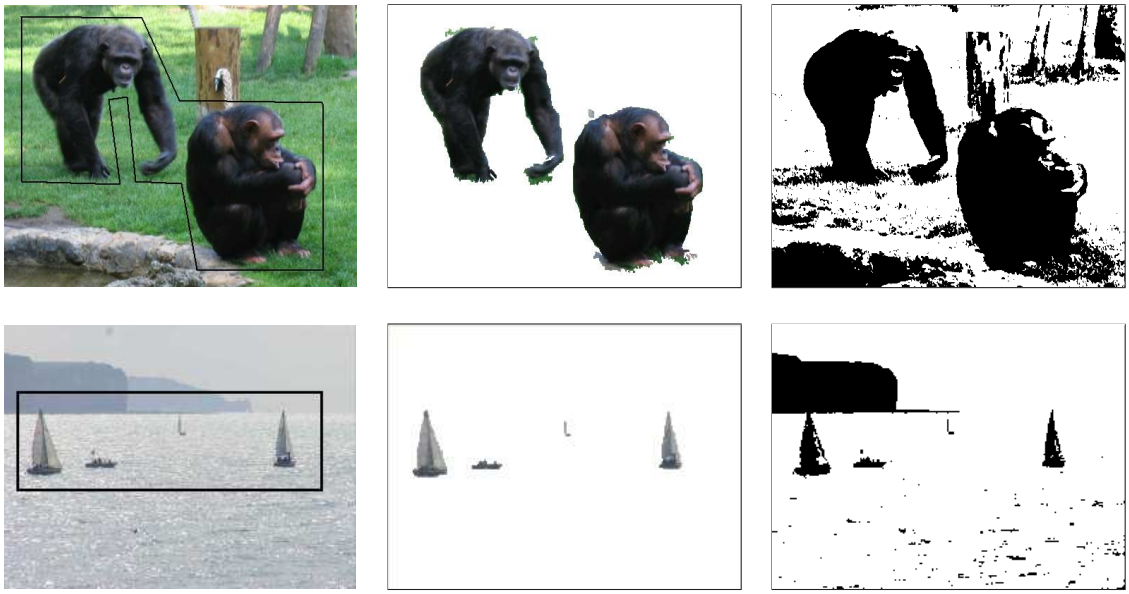
Le contour obtenu avec les ciseaux intelligents est de qualité similaire à celui obtenu à l'aide de notre outil. Cependant, la boîte d'extraction bénéficie d'une meilleure maniabilité devant celle des ciseaux intelligents.

Extraction multi-objets

Cet outil peut être utilisé afin d'extraire plusieurs objets d'intérêt simultanément. C'est ce qu'illustre la figure 3.14. Les résultats sont comparés aux images originales seuillées afin de mettre en évidence la non-trivialité de l'extraction. Même si l'objet semble bien contrasté avec le fond, son extraction n'en est pas obligatoirement triviale.

Conclusion

Cet outil se révèle pratique et efficace lorsque nous sommes en présence d'objets contrastés par rapport à un fond comportant des zones homogènes étendues. Cependant, dans le cas d'images naturelles, il est rare que ces contraintes soient respectées. C'est pourquoi nous avons mis au point un outil plus robuste et dont le résultat est plus contrôlable, fondé sur une initialisation plus précise de la pyramide locale.



(a) Sélection des objets (b) Extraction (c) Seuillage binaire

FIG. 3.14: Extraction multi-objets à l'aide d'un polygone englobant

3.2.2 Localisation du contour - ruban d'extraction

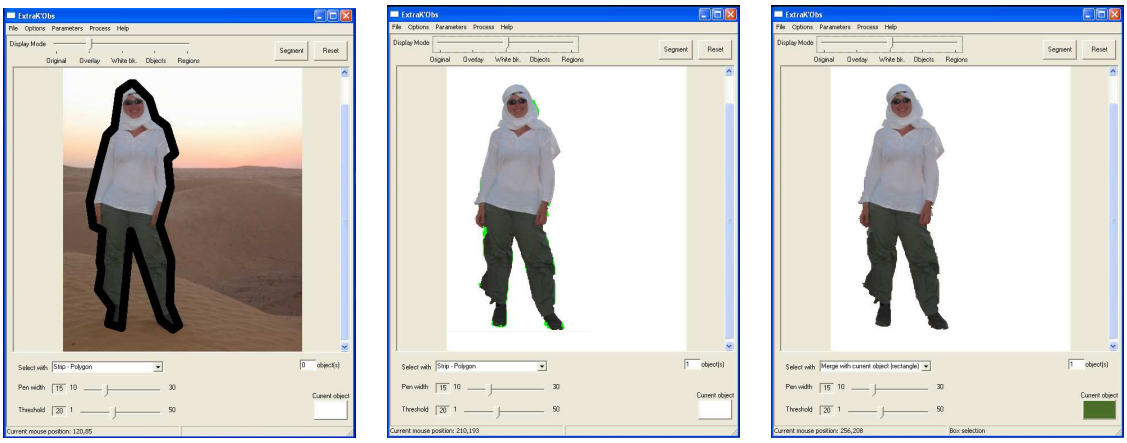
概念

Cette section présente notre deuxième outil interactif, plus précis que le précédent, permettant à l'utilisateur de focaliser la segmentation sur le contour de l'objet d'intérêt.

Cette approche nécessite d'avoir une connaissance grossière de la position du contour de l'objet (cf. figure 3.15.a). L'utilisateur est invité à localiser de manière approximative le contour, en traçant sur ce dernier un ruban fermé. Ce ruban induit trois zones : l'extérieur, l'intérieur et la zone définie par le ruban lui-même. Nous retrouvons ici une initialisation similaire à l'outil *Bayes matting* présenté dans le chapitre précédent.

Les pixels de l'extérieur appartiennent à la racine du fond. Ils sont alors modélisés par un seul sommet dans le graphe de segmentation. Quant aux pixels situés à l'intérieur, ils appartiennent à l'objet d'intérêt lui-même. Tous les pixels du ruban font partie de la zone indéfinie. Ces derniers seront segmentés afin d'être fusionnés avec l'une ou l'autre des racines.

Le principe de l'extraction consiste en la propagation des étiquettes du fond et de celle de l'objet dans la zone indéfinie. Ainsi, les régions de la zone indéfinie similaires à celles de la racine objet fusionnent avec l'objet tandis que les autres régions sont attribuées au fond. La racine de l'objet est un élément essentiel du masque puisqu'elle détermine la région minimale correspondant à l'objet après l'extraction. Il se peut que des régions indéfinies trop différentes de leurs voisines persistent après la segmentation comme le montre la figure 3.15.b. La classification de ces régions généralement peu nombreuses est confiée à l'utilisateur. Notre application Extrak'Obs propose des outils pour faciliter cette classification manuelle et ainsi obtenir une partition finale comportant autant de racines que d'objets d'intérêt (cf. figure 3.15.c).



(a) Tracer du ruban de segmentation

(b) Extraction de l'objet avec des régions indéfinies (vert clair)

(c) Fusion manuelle des régions indéfinies avec l'objet *ad hoc*

FIG. 3.15: Exemple d'utilisation du ruban d'extraction

3.2.3 Résultats

Il est naturel que la méthode donne des résultats de qualité dans le cas d'objets homogènes sur un fond homogène. Beaucoup plus intéressant, la méthode fonctionne bien lorsque les objets ont une colorimétrie hétérogène sur un fond relativement homogène (ou inversement) au niveau de la zone indéfinie, même lorsque le signe du gradient s'inverse à la frontière entre le fond et l'objet. Les dégradés sont aussi bien pris en compte. Cette approche est intéressante notamment lorsque les objets ont une forme et une composition complexes puisqu'elle se focalise uniquement sur le contour.

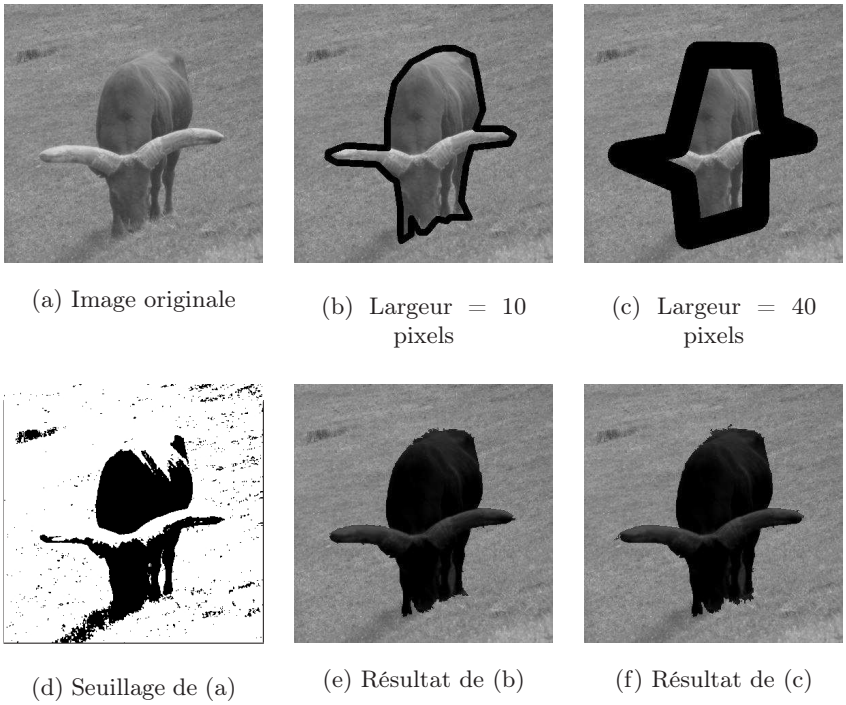
Les résultats présentés par la suite ne sont pas post-traités, et sont obtenus avec le même seuil global de similarité T_g. La figure 3.16.d montre comment il est souvent impossible avec des méthodes simples (ici un seuillage) d'extraire correctement un objet, soit à cause de sa texture, de son ombre portée ou du manque de contraste localement. On peut remarquer que le contour du dos de l'animal (figure 3.16.a) n'est pas du tout contrasté, ni en couleurs, ni en niveaux de gris.

Robustesse aux variations de la zone de segmentation

Les résultats (figures 3.16 et 3.17) montrent que notre méthode est peu sensible à la façon dont l'utilisateur initialise le traitement : en effet, une variabilité importante de l'épaisseur du ruban, de son positionnement (plutôt intérieur ou extérieur), de sa forme (plus ou moins régulière), fournit des résultats très similaires. Le ruban peut être tracé soit à l'aide de segments (figures 3.16.c,e, 3.17.a,c), soit à main levée (figure 3.17.e,g).

Les figures 3.16.d et f présentent deux résultats obtenus avec deux épaisseurs différentes (10 et 40 pixels). Le ruban de la figure 3.16.e est obtenu avec 10 clics de la souris seulement ; en revanche, il fournit des détails précis de la tête et des jambes de l'animal.

Dans la figure 3.17.a, la segmentation est obtenue avec un ruban positionné plutôt dans l'objet alors que dans la figure 3.17.c, il est plutôt dans le fond. Bien que ces résultats ne soient pas exactement identiques, ils diffèrent assez peu. Bien sûr, il est préférable d'inclure dans la racine la majeure partie des pixels connus comme appartenant à l'objet d'intérêt, afin de s'assurer qu'ils appartiennent effectivement au résultat à l'issue du traitement.



(a) Image originale (b) Largeur = 10 pixels (c) Largeur = 40 pixels

(d) Seuillage de (a) (e) Résultat de (b) (f) Résultat de (c)

FIG. 3.16: Une variabilité importante de l'épaisseur du ruban entraîne une faible variabilité des résultats

Dans les figures 3.17.e et g, c'est la forme du ruban qui a varié : l'utilisateur peut effectivement sélectionner dans le ruban des agrégats de pixels qui contiennent de nombreux détails de contour afin de les extraire correctement, comme le montre la jambe arrière dans la figure 3.17.g.

Extraction mutli-objets

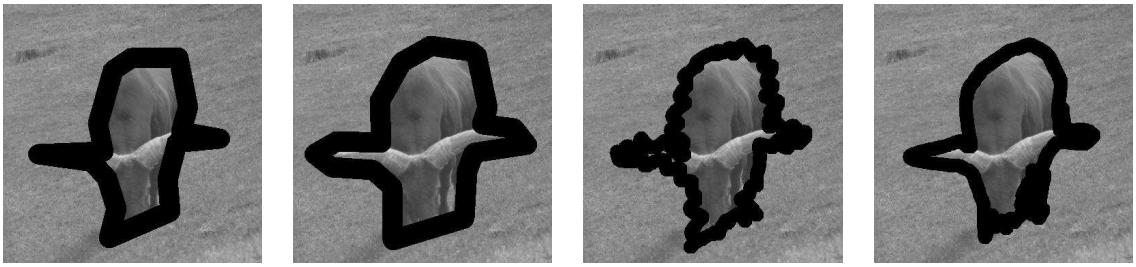
Enfin, la méthode présentée peut être utilisée telle quelle, sans modification, avec un nombre quelconque de racines. Le ruban fermé peut avoir une topologie quelconque et être connexe à un nombre quelconque de racines. Comme le montre l'exemple de la figure 3.18, l'approche proposée garantit un nombre d'objets final égal au nombre de racines. Il est donc important que le ruban soit fermé pour que l'intérieur puisse se distinguer de l'extérieur.

3.2.4 Discussion

Nous proposons une méthode bien adaptée aux images complexes où le contour d'un objet peut successivement prendre des configurations très différentes : gradient positif, négatif, contours multiples. La figure 3.19 montre un résultat obtenu avec notre application Extra-K'Obs pour un objet qui se distingue assez difficilement du fond.

La texture du fond et/ou de l'objet est, bien entendu, un problème important auquel doivent faire face la majorité des méthodes de segmentation et notamment, celle que nous présentons. La reconnaissance de présence de textures (objet ou fond) et de contours pourrait servir à favoriser certaines fusions lors de la propagation concurrente d'étiquettes. Il semblerait par exemple judicieux de favoriser la propagation dans les zones peu texturées qui pourraient alors s'étendre jusqu'à être stoppées par des zones plus texturées.

La localisation de la frontière finale n'est pas liée géométriquement à la localisation et à l'épaisseur du ruban. Ces deux paramètres peuvent néanmoins influencer le résultat final dans



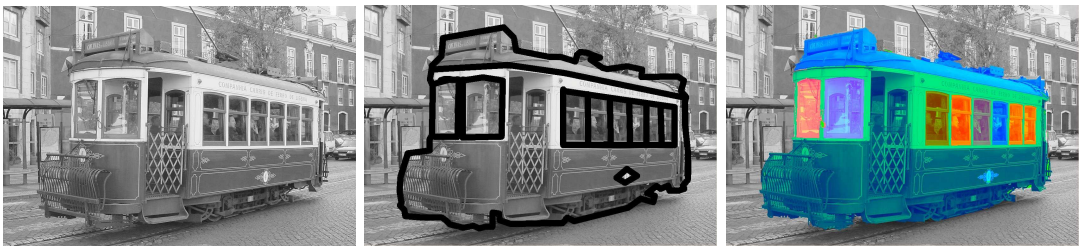
(a) Ruban intérieur (b) Ruban extérieur (c) Ruban bruité (d) Ruban irrégulier

(e) Résultat de (a) (f) Résultat de (b) (g) Résultat de (c) (h) Résultat de (d)

FIG. 3.17: Une variabilité importante du positionnement et de la régularité du ruban entraîne une faible variabilité des résultats

(a) Image originale (b) Initialisation (c) Partition obtenue

FIG. 3.18: Segmentation de plusieurs régions d'intérêt



(a) Image originale (b) Initialisation (c) Partition obtenue

FIG. 3.19: Exemple de résultat obtenu avec l'application ExtraK'Obs

le sens où une localisation et une largeur mieux adaptées limiteront les erreurs de segmentation (faux contour, fuites, ...). La frontière finale correspond localement à la frontière entre deux régions qui ont crû puis acquis une étiquette. Ainsi, la croissance de régions non encore étiquetées est un élément essentiel que nous devons étudier de manière plus approfondie.

Après avoir exploré le domaine interactif, nous allons nous intéresser à l'initialisation automatique de la pyramide locale. L'objectif est de supprimer toute interaction de l'utilisateur en fournissant tout de même des partitions proches de partitions sémantiques.

3.3 Initialisation spatiale automatique

概念

Cette section présente une initialisation spatiale et une extension orientée sur des critères perceptifs adaptés à la pyramide locale visant à automatiser l'extraction de régions d'intérêt.

L'étude du paragraphe précédent sur la robustesse des résultats obtenus par segmentation locale permet d'envisager de définir la zone de segmentation de manière automatique. L'objectif est de localiser les contours des objets d'intérêt sans l'intervention de l'utilisateur. L'absence de supervision de la part d'un utilisateur permettant d'injecter du sens dans la segmentation fait que nous parlerons maintenant d'extraction de régions d'intérêt ou de partition pertinente de l'image et non plus d'extraction d'objets d'intérêt. Une partition sera considérée comme pertinente si elle se rapproche d'une partition sémantique.

Nous allons présenter notre méthode de segmentation spatiale fondée sur les objets reposant sur une analyse des hétérogénéités de l'image. Son objectif est de fournir des partitions pertinentes de l'image permettant de résumer au mieux le contenu sémantique de l'image.

3.3.1 Localisation des contours par carte d'homogénéité

概念

La segmentation locale fondée sur les objets nécessite de connaître grossièrement la position des contours de l'objet pour l'extraire. Dans cette section, j'explique comment automatiser la localisation des contours pour initialiser la pyramide locale.

Il existe plusieurs mesures d'hétérogénéités d'une image. Feng Jing *et al.* présentent dans [59] une méthode originale qui localise les homogénéités en textures d'une image. Leur méthode de segmentation est dérivée de celle présentée par Y. Deng et B. S. Manjunath dans [32]. Cette méthode réside dans la construction d'une carte d'homogénéité appelée la *H-image* qui fournit un résultat très intéressant et facilement exploitable pour l'initialisation automatique de la pyramide locale.

Construction de la H-image

Avant d'aller plus loin reprenons les quelques notations inspirées de [59]. Soit M le motif permettant de calculer l'homogénéité. Classiquement, M est une fenêtre carrée de largeur $(2N+1)$ exprimée en pixels. Soit $c = (x_c,\ y_c)$ le pixel central du motif dont le critère est I_c. Ce critère peut être l'intensité du pixel mais aussi la couleur ou tout autre critère dont on

recherche l'homogénéité dans l'image. Chaque pixel $p_i = (x_i,\ y_i)$, $i \in [\![1,\ (2N+1)^2]\!]$ dans M correspond à un vecteur dont l'expression est la suivante :

$$cp_i = \begin{pmatrix} x_i - x_c \\ y_i - y_c \end{pmatrix} \tag{3.9}$$

Ce dernier permet de construire un nouveau vecteur f_i de même orientation mais dont la norme correspond à la différence des critères I_i et I_c des pixels p_i et c.

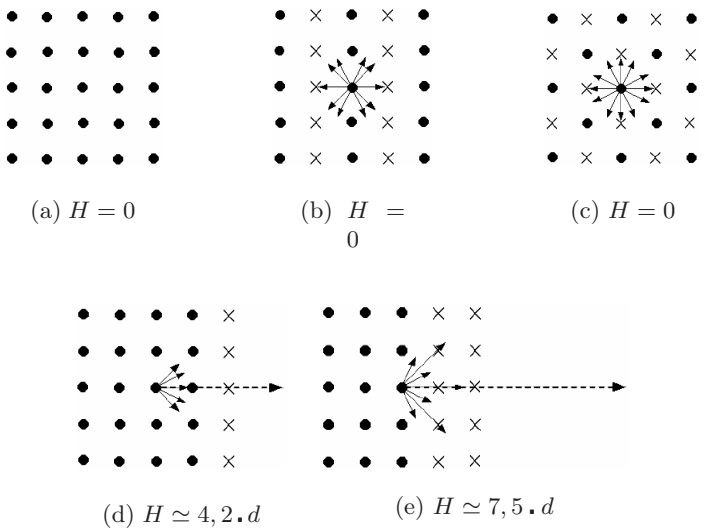
$$f_i = (I_i - I_c) \cdot \frac{cp_i}{\|cp_i\|} \tag{3.10}$$

Soit f la somme de l'ensemble des vecteurs f_i de M :

$$f = \sum_{i=1}^{(2N+1)^2} f_i \tag{3.11}$$

Ces vecteurs permettent de définir la valeur H comme étant la norme du vecteur f : $H = \|f\|$.

La figure 3.20 montre l'évolution de la valeur selon le contenu homogène ou hétérogène de M. Dans cette figure, '•' et '×' indiquent deux valeurs différentes du critère de deux pixels. La valeur '×' est supérieure à celle de '•' et leur distance est donnée par d ($d > 0$). Les lignes continues représentent f_i tandis que les lignes en pointillés représentent f.



(a) $H = 0$ (b) $H = 0$ (c) $H = 0$

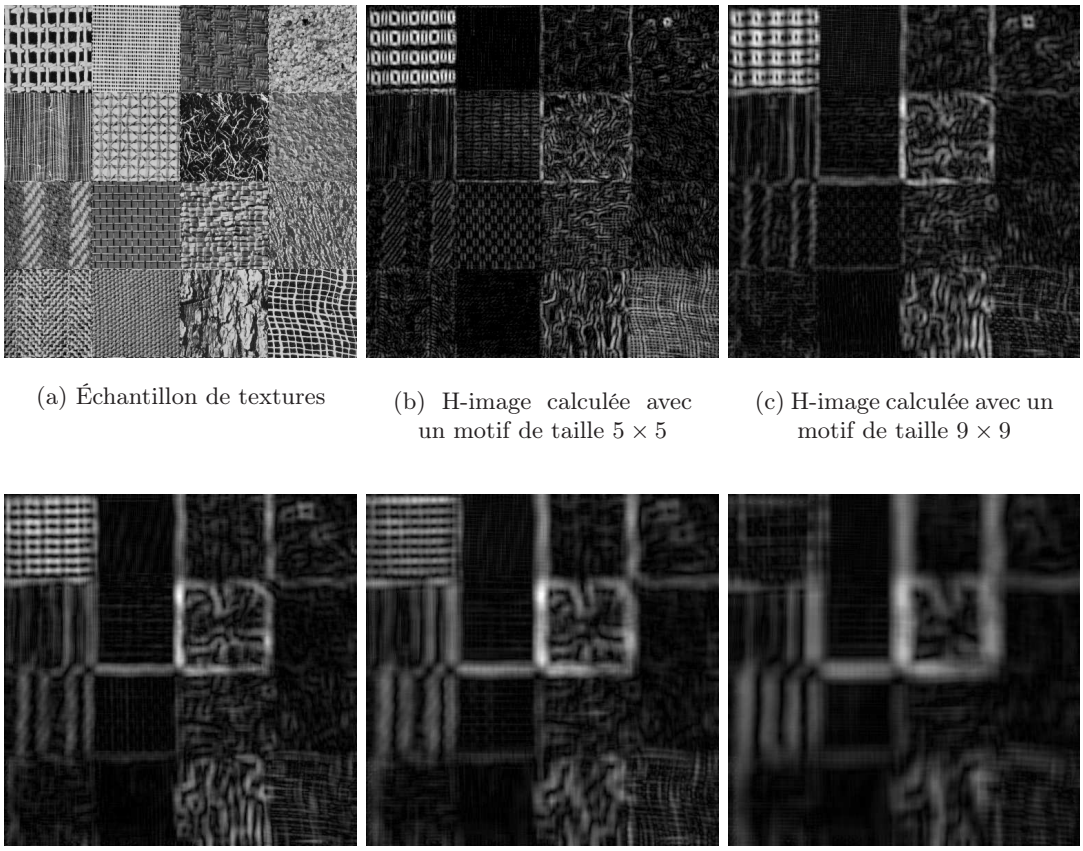
(d) $H \simeq 4,2 \cdot d$ (e) $H \simeq 7,5 \cdot d$

FIG. 3.20: Calcul de la valeur H dans une fenêtre de recherche (d'après [59])

Ainsi, les valeurs de H sont très faibles pour des zones homogènes comme le montrent les figures 3.20.a, 3.20.b et 3.20.c. Tandis que pour des zones hétérogènes (pouvant contenir plusieurs régions homogènes), les valeurs de H sont élevées. C'est l'exemple présenté dans les figures 3.20.d et 3.20.e. Ces deux figures illustrent également le fait que la valeur de H augmente avec la proximité d'un contour : plus le pixel central de M est proche du contour plus la valeur de H est élevée. La valeur de H dépend également du contraste entre les régions contenues dans M. Un contraste fort implique une valeur de d élevée. Ce qui a pour effet d'augmenter la valeur H.

La *H-image* est une image en niveaux de gris dont l'intensité des pixels est donnée par la valeur H calculée à partir du motif M centré sur ces pixels. Les parties sombres représentent les régions homogènes tandis que les parties lumineuses indiquent les contours de ces régions.

Un problème de cette méthode est la détermination de la taille du motif M. Un motif de petite taille a tendance à détecter des contours selon le critère choisi (intensité, couleur ...). Au contraire, un motif de grande taille détecte des contours de textures. La figure 3.21 montre des H-images calculées sur l'intensité (composante Y) d'échantillon de textures. Plus la taille du motif augmente plus les frontières entre les textures différentes apparaissent (cf. figures 3.21.d, e et f). Tandis que les H-images construites avec de petits motifs fournissent des contours du critère d'intensité (cf. figures 3.21.b et c).



(a) Échantillon de textures

(b) H-image calculée avec un motif de taille 5×5

(c) H-image calculée avec un motif de taille 9×9

(d) H-image calculée avec un motif de taille 13×13

(e) H-image calculée avec un motif de taille 17×17

(f) H-image calculée avec un motif de taille 25×25

FIG. 3.21: Effet de la taille du motif de la H-image sur des textures

Certaines techniques multi-échelles telles que celle introduite par J. L. Crowley et A.C. Parker qui se place dans un cadre de détection de points d'intérêt [29], présentent une sélection automatique de l'échelle la plus pertinente pour l'application. Dans notre méthode ce paramètre est choisi par l'utilisateur. En effet, le lien entre l'échelle et la pertinence de la partition qu'elle engendre ne semble pas trivial à mettre en œuvre. Pour des contraintes de temps de calcul, lorsque la taille du motif augmente il est nécessaire d'échantillonner le nombre de pixels utilisés dans la fenêtre en faisant au préalable un filtrage passe-bas de l'image originale. Nous utilisons le même tableau présenté dans [32] renseignant l'échelle, l'échantillonnage et la taille de motif correspondante.

Échelle	Motif (pixels)	Échantillonnage
1	9×9	$1/(1 \times 1)$
2	17×17	$1/(2 \times 2)$
3	32×32	$1/(4 \times 4)$
4	65×65	$1/(8 \times 8)$

Critères d'homogénéité

Une H-image peut être calculée à partir de n'importe quel critère de l'image. Toutefois, l'homogénéité selon la couleur est un critère souvent utilisé pour la détection d'objet. Il existe une multitude d'espaces couleur (cf. annexe A). Cependant, nous avons choisi ceux dont l'indépendance entre les composantes couleur et celle de la luminosité est la plus marquée afin de pouvoir privilégier la couleur. Nous nous sommes donc tournés vers les deux espaces couleur : TLS (Teinte Luminance Saturation) et $L^*a^*b^*$ dont nous comparerons les résultats par la suite.

Classiquement, une H-image est calculée uniquement sur un seul critère. Or, il est intéressant de calculer une H-image rendant compte de l'homogénéité de plusieurs composantes des différents espaces couleur. L'objectif est donc de calculer une H-image par critère, puis de trouver un mélange permettant de générer une H-image globale rendant compte des homogénéités de l'image fidèlement à la perception visuelle humaine.

En ce qui concerne l'espace TLS, souvent énoncé pour ses propriétés perceptives intéressantes, le principe est de calculer une H-image sur chacune des composantes de l'espace couleur : H_T, H_L et H_S, puis de les combiner en une seule H-image globale : H_{TLS}. Une particularité de l'espace TLS est de pouvoir déterminer la pertinence de la teinte (T). En effet, la pertinence de la teinte dépend fortement de la saturation. Lorsque la couleur est fortement saturée, l'information peut être considérée comme fiable. Cependant, lorsque la saturation est faible, la teinte devient très sensible au bruit et sa pertinence en est fortement diminuée. Il est alors préférable de lui substituer l'information de la luminance. Les problèmes d'ombre peuvent par exemple être évités en utilisant ce coefficient de pertinence.

La pertinence de la teinte est donnée par une fonction dépendant de la saturation : $\alpha(S)$. Les propriétés de courbure d'une sigmoïde semblent être adaptées pour ce type de coefficient [22]. La pertinence de la teinte est donc donnée par la relation suivante :

$$\alpha(S) = \frac{1}{1 + e^{(-\beta(S - S_0))}} \tag{3.12}$$

Où S_0 est la saturation permettant de déterminer un niveau moyen de pertinence ($\alpha(S_0) = 0,5$) et β représente la pente en ce point. Cette pente permet de doser le mélange des H-images (T/L). La figure 3.22 présente l'allure d'une telle fonction.

Le coefficient α nous permet de déterminer le mélange H_{TLS} qui est obtenu pour chaque pixel par la relation suivante :

$$H_{TLS}(p) = \sqrt{\alpha(S(p))H_T(p)^2 + (1 - \alpha(S(p)))H_L(p)^2 + H_S(p)^2} \tag{3.13}$$

La saturation est conservée dans le mélange car elle a un fort impact visuel. Elle permet de discriminer des teintes proches qui visuellement sont bien distinctes.

Dans le cas de l'espace $L^*a^*b^*$, le mélange H_{Lab} est donnée par l'équation suivante :

$$H_{Lab}(p) = \sqrt{(H_L(p)^2 + H_a(p)^2 + H_b(p)^2)} \tag{3.14}$$

Le coefficient de réflexion

Il apparaît clairement que l'information couleur est un critère pertinent pour la détection d'objet. Quant à la luminance, elle induit souvent des hétérogénéités parasites au sein d'un

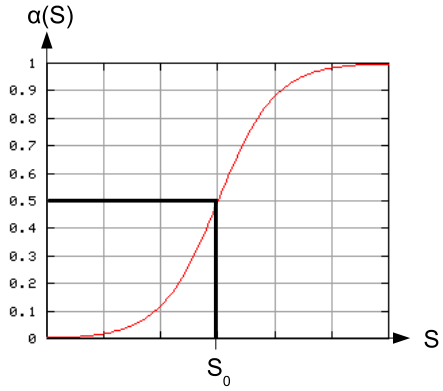


FIG. 3.22: Sigmoïde traduisant la pertinence $\alpha(S)$ de la teinte en fonction de la saturation S

même objet. Il peut être intéressant de substituer l'information de la luminance par le coefficient de réflexion introduit dans [94]. En effet, ce coefficient est un invariant photométrique qui dépend du matériau et de la forme (courbure) de l'objet. Dans [94], S. K. Nayar *et al.* proposent de calculer localement le coefficient de réflexion R à partir de pixels proches p_1 et p_2 et de leur luminance respective I_1 et I_2. R est donné par l'équation 3.15.

$$R(p_1, p_2) = \left(\frac{I_1 - I_2}{I_1 + I_2} \right) \tag{3.15}$$

R permet de mesurer le changement d'albédo[3] entre deux pixels voisins. Considérons plusieurs couples de pixels voisins. Si tous les pixels appartiennent au même objet, la variance des valeurs de R de chaque couple est faible. Tandis que si les pixels au sein d'un même couple appartiennent à des objets différents, il est probable que les conditions d'éclairement et que la courbure ne soient pas similaires. De ce changement résulte une forte variation des valeurs de R de chaque couple.

Nous avons intégré ce critère dans le calcul de la H-image. Au lieu de calculer la différence de luminance entre pixels, nous calculons le changement d'albédo donné par R entre le pixel central et chaque pixel du motif M. L'expression de f_i dans l'équation 3.10 devient :

$$f_i = R(c, p_i) \cdot \frac{cp_i}{\|cp_i\|} \tag{3.16}$$

Le changement d'albédo ne peut être calculé que pour des pixels voisins. Ainsi, la taille du motif M doit rester relativement faible pour conserver cette hypothèse. Lorsque les pixels de M appartiennent à une même surface de l'objet, le coefficient de réflexion reste stable même si l'éclairement varie. La valeur H reste alors faible. Lorsque R varie dans M, H prend des valeurs élevées. Il en résulte que dans une H-image calculée sur le critère du coefficient de réflexion, les valeurs de faibles intensités correspondent à des surfaces homogènes en courbure, tandis que des valeurs élevées traduisent une frontière entre deux surfaces d'éclairement et de formes différentes. L'intégration du coefficient R dans le calcul de la H-image permet d'être moins sensible aux changements de luminosité dus à la courbure de la surface des objets.

Nous verrons dans l'étape suivante de seuillage de la H-image, que l'utilisation du coefficient de réflexion permet de limiter les fausses hétérogénéités. La figure 3.23 présente les différents résultats de seuillage des H-images obtenues à partir des deux composantes de luminances des espaces TLS et $L^*a^*b^*$ et du coefficient de réflexion R.

[3] L'albédo est le rapport de l'énergie lumineuse réfléchie par une surface sur l'énergie lumineuse incidente

Soit H_R la H-image construite à partir du coefficient de réflexion. En remplaçant la composante de luminance L par le coefficient de réflexion R, le mélange des H-images devient dans le cas de l'utilisation des composantes de teinte et de saturation :

$$H_{TRS}(p) = \sqrt{\alpha(S(p))H_T(p)^2 + (1 - \alpha(S(p)))H_R(p)^2 + H_S(p)^2} \tag{3.17}$$

Dans le cas des composantes a^* et b^* :

$$H_{Rab} = \sqrt{(H_R^2 + H_a^2 + H_b^2)} \tag{3.18}$$

Seuillage de la H-image

L'objectif est de séparer les zones hétérogènes des zones homogènes dans les mélanges H_{Rab} et H_{TRS} afin de définir les zones de segmentation vues précédemment au paragraphe 3.1.7. Pour cela, une méthode de seuillage est utilisée. Nous avons opté pour la même approche qui est utilisée dans la méthode introduite par [59]. Ce seuillage permet de prendre en compte des informations à la fois globales et locales. Il s'effectue en 2 étapes :

Étape 1 : Calcul de la moyenne μ et de l'écart-type σ des valeurs H sur l'ensemble des pixels de la H-image.

Étape 2 : Pour chaque pixel, calcul de la moyenne μ_M et de l'écart-type σ_M des pixels voisins. Le voisinage est défini comme une fenêtre carrée de côté $2N + 1$.

Un seuil est défini à partir de ces valeurs :

$$T_M = \min(\max(\mu_M - \alpha_1 \cdot \sigma_M,\ \mu - \alpha_2 \cdot \sigma),\ \mu + \alpha_3 \cdot \sigma) \tag{3.19}$$

où α_i, $i \in [\![1,\ 3]\!]$ sont 3 paramètres modifiables mais placés à des valeurs fixes conformément à [59] : $\alpha_1 = 0,5$ et $\alpha_2 = \alpha_3 = 0,4$.

Si la valeur H d'un pixel p est supérieure à ce seuil T_M, p appartient à la zone indéfinie. Sinon il est destiné à appartenir à une racine. La classification par T_M assure que les pixels des racines ont une valeur faible par rapport à l'ensemble de la H-image ($\cdot \leq \mu - \alpha_2 \cdot \sigma$) mais également vis à vis de leur voisinage ($\cdot \leq \mu_M - \alpha_1 \cdot \sigma_M$). Dans ce dernier cas, on s'assure que les valeurs de H dans le voisinage ne sont pas trop élevées ($\cdot \leq \mu + \alpha_3 \cdot \sigma$).

Une fois les mélanges de H-images seuillés, nous procédons à des traitements de morphologie mathématique afin d'améliorer la robustesse des racines. Ces dernières doivent être, rappelons-le, des composantes connexes entourées par des zones indéfinies (hétérogènes). Afin de définir des composantes connexes, il est par précaution, utile d'effectuer une fermeture sur les zones indéfinies. Les composantes connexes ainsi obtenues permettent de construire les racines finales et de manière duale les zones de segmentation. Cette fermeture supplémentaire permet de délimiter correctement la racine d'un objet dont la frontière serait localement peu contrastée avec le fond.

Chaque racine est ensuite étiquetée et la segmentation locale peut être effectuée. La figure 3.23 montre que l'utilisation du coefficient de réflexion permet de limiter le nombre de composantes connexes qui seraient dues à des changements de luminosité au niveau des courbures des objets par rapport aux composantes L et V.

Limitation de la sur-segmentation de la pyramide locale

A l'issue de la segmentation locale, il est possible de diminuer le nombre de régions constituant l'apex de la pyramide pour deux raisons :

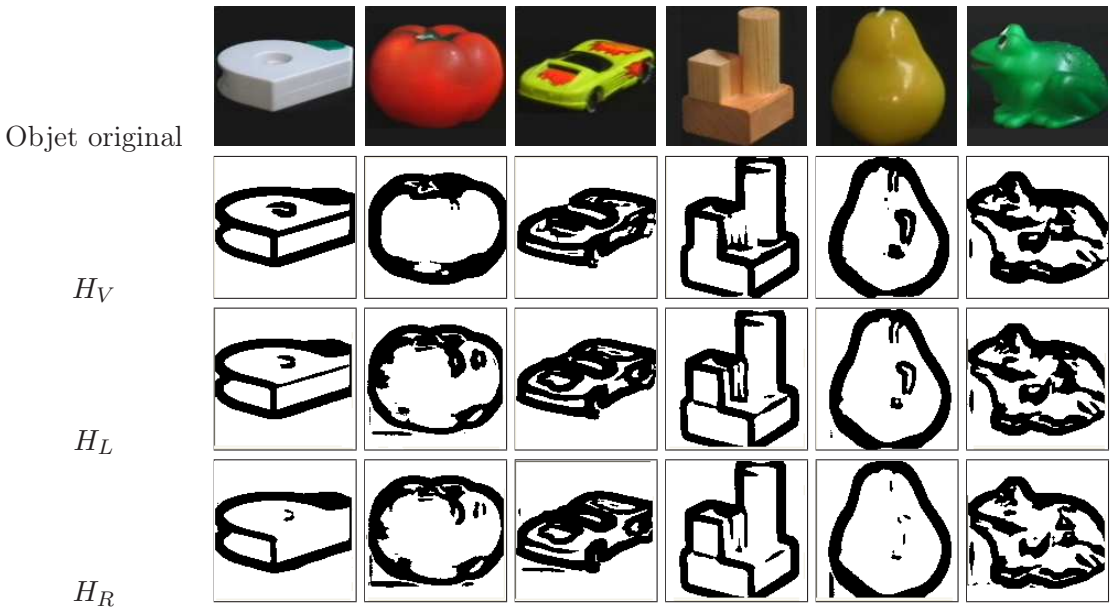


FIG. 3.23: Comparaison des résultats de seuillage des H-images H_V, H_L et H_R

1. L'apex peut contenir des régions parasites de surfaces faibles qui n'ont pas pu fusionner avec des racines. Ces régions favorisent le phénomène de sur-segmentation. Pour palier ce problème, la pyramide irrégulière propose l'étape de relaxation (voir paragraphe 3.1.6) qui permet de fixer la surface minimum des régions appartenant à la partition finale. Cette étape intervenant soit à l'apex soit lors de la construction des derniers niveaux, permet de forcer la fusion de petites régions avec les régions adjacentes les moins dissimilaires en couleur. Une relaxation relativement forte est donc effectuée à l'apex pour des soucis de simplification de la partition : la surface minimum est placée empiriquement à 500 pixels.
2. Le principe de la segmentation locale empêche la fusion des racines puisqu'elles portent des étiquettes différentes. Cependant, il se peut que des racines voisines de couleurs similaires coexistent à l'apex. La contrainte des étiquettes est donc relâchée dans la suite du procédé de segmentation afin de fusionner les racines similaires obtenues à l'apex de la pyramide locale classique. La fusion est alors seulement dirigée par le seuil local T_l et le seuil global T_g.

La combinaison de ces deux étapes permet de diminuer d'environ 70% le nombre de régions obtenues à l'apex. Généralement, la segmentation locale fournit une partition d'une centaine de régions. Après ces deux étapes de simplification, les partitions ne contiennent plus qu'une trentaine de régions. Le risque encouru lors de l'étape de la relaxation est de fusionner de petites régions du contour d'un objet avec le fond lorsque celle-ci appartiennent à un dégradé. Ce type de problème est visible dans les résultats présentés à l'annexe B.

Évaluation des résultats

Nous présentons à l'annexe B un échantillon de résultats de la segmentation locale initialisée par la méthode de la H-image sur les mélanges H_{Rab} et H_{TRS}. Ces résultats sont obtenus avec le même jeu de paramètres pour toutes les images. C'est-à-dire avec la même échelle et le même seuil global. L'annexe présente également les effets du changement d'échelle dans le calcul de la H-image.

(a) Image originale

(b) Résultat de la segmentation locale ($\sim$ 70 régions)

(c) Simplification de la partition ($\sim$ 10 régions)

FIG. 3.24: Limitation de la sur-segmentation

Les images utilisées sont issues de la base de données présentée par Martin *et al.* dans [80]. Cette base de données[4] est constituée de 350 images issues de la base d'images *Corel* et contenant au moins un objet discernable. Chaque image est segmentée manuellement par 1 à 5 personnes. Ces partitions manuelles nous ont permis d'évaluer les deux méthodes de segmentation locale afin de montrer la qualité des partitions obtenues.

Cependant, l'élaboration d'un test d'évaluation n'est pas chose aisée et il est nécessaire de prendre avec prudence les résultats obtenus. En effet, un premier phénomène permettant de souligner la difficulté de l'évaluation automatique de partitions est le fait que des partitions obtenues manuellement à partir d'une même image, peuvent être très différentes d'un sujet à un autre. Que ce soit par rapport au nombre de régions qu'elles contiennent ou par rapport aux différentes régions de l'image sur lesquelles s'est focalisé le sujet. Deux partitions d'une même image peuvent par exemple contenir un nombre similaire de régions mais fournir une description de zones différentes. Ceci provient du fait que la partition découle directement de la perception de la scène qui peut être éventuellement différente d'un sujet à l'autre. De plus, les contours des objets dans les images numériques sont flous à cause de deux raisons principales. La première est due au fait que la résolution est finie. Ainsi l'intensité des pixels du contour est forcément un mélange de l'intensité de l'objet et du fond. La deuxième se traduit par le fait que chaque système optique est caractérisé par une fonction de flou. Ainsi chaque point observé de la scène se projette non pas sur un point mais sur une zone plus ou moins étalée.

Ainsi, comme le montre Martin dans [80], au sein d'une même base de données composée de plusieurs segmentations manuelles pour chaque image, il apparaît clairement des divergences notables dues à la précision des sujets et à leur perception de la scène. Martin présente deux métriques pour évaluer la cohérence entre deux partitions différentes de la même image [80].

Soient P_1 et P_2 ces deux partitions. L'objectif est de définir une valeur comprise dans l'intervalle $[\![0 \dots 1]\!]$ où 0 signifie la parfaite correspondance. Chaque pixel p_i appartient à deux régions R_1 et R_2 appartenant respectivement à P_1 et P_2. R_1 et R_2 sont des ensembles de pixels. Si l'un est un sous-ensemble de l'autre, le pixel p_i est déclaré comme appartenant à une zone de raffinement. L'erreur locale est alors de 0. S'il n'y a pas de relation de sous-ensemble, cela traduit le fait que les régions ne se recouvrent que partiellement. Dans ce cas l'indice d'erreur augmente. Par la suite nous utiliserons les notations suivantes. Soit $\setminus$ le symbole traduisant la différence[5] entre deux ensembles et $|A|$ le cardinal de l'ensemble A.

Si $R(P,\ p_i)$ est l'ensemble des pixels correspondant à la région de la partition P qui

[4] *http : //www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/*

[5] $A \setminus B$ représente l'ensemble dont les éléments sont ceux de A qui n'appartiennent pas à B

contient le pixel p_i, l'erreur de raffinement locale est définie comme suit :

$$E(P_1,\ P_2,\ p_i) = \frac{|R(P_1,\ p_i) \backslash R(P_2,\ p_i)|}{|R(P_1,\ p_i)|} \tag{3.20}$$

$E(P_1,\ P_2,\ p_i)$ n'est pas une mesure symétrique. En effet, dans le cas où P_1 est un raffinement de P_2, elle vaut 0. Tandis que dans le cas inverse elle prend une valeur supérieure à 0. L'erreur de raffinement locale est donc unidirectionnelle. Il découle de cette propriété deux manières de combiner les valeurs obtenues en chaque pixel :

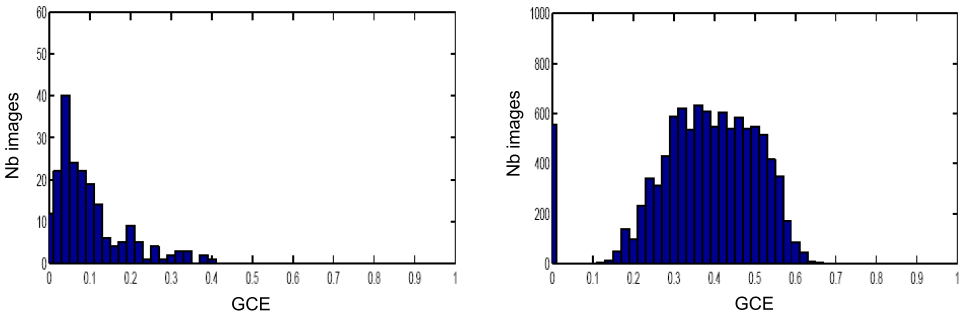
- L'erreur de cohérence globale (*Global Consistency Error* : *GCE*) qui impose la même direction à tous les raffinements.
- L'erreur locale de cohérence (*Local Consistency Error* : LCE) qui permet de considérer des raffinements mutuels dans des zones différentes de l'image.

Soit N le nombre de pixels.

$$GCE(P_1,\ P_2) = \frac{1}{N} \min \left\{ \sum_i E(P_1,\ P_2,\ p_i), \sum_i E(P_2,\ P_1,\ p_i) \right\} \tag{3.21}$$

$$LCE(P_1,\ P_2) = \frac{1}{N} \sum_i \min \left\{ E(P_1,\ P_2,\ p_i), E(P_2,\ P_1,\ p_i) \right\} \tag{3.22}$$

La figure 3.25.a présente les divergences observables entre les partitions manuelles d'une même image et ceci pour chaque image de la banque de données. La figure 3.25.b montre la distribution de la GCE entre partitions concernant des images différentes. Le pic de la distribution de l'erreur entre partitions de mêmes images est plus proche de 0 que la distribution de l'erreur calculée sur des couples aléatoires de partitions.



(a) GCE entre les différentes partitions de chaque image

(b) GCE entre des partitions d'images différentes

FIG. 3.25: Distribution de la GCE calculée sur la base de données de segmentation manuelle (d'après [80])

Ces mesures acceptent une certaine tolérance sur le nombre respectif de régions dans les partitions puisqu'elles prennent en compte le raffinement. En revanche, elles sont d'autant plus significatives que le nombre de régions des 2 partitions à comparer est proche. En effet, il existe deux segmentations triviales induisant en erreur la mesure : celle qui consiste à affecter un pixel par région et celle qui ne contient qu'une seule région. Ainsi la première est un raffinement de n'importe quelle partition, tandis que toutes partitions est un raffinement de la deuxième.

L'inconvénient dans notre cas, est qu'il n'est pas possible de fixer le nombre de régions dans la partition finale de la segmentation locale sans dénaturer complètement le résultat. Généralement le nombre de régions fournies par la segmentation locale est supérieur au nombre de régions obtenues par une segmentation manuelle. Il dépend principalement de la tolérance sur la taille minimale des régions pouvant appartenir à la partition finale. L'objectif du test est donc d'évaluer la faculté de la segmentation locale à fournir les contours de l'objet d'intérêt capturé dans la segmentation manuelle tout en laissant la liberté du nombre de régions décrivant l'objet d'intérêt.

Pour cela, nous avons choisi d'étudier uniquement la distribution de la GCE à partir de 100 images tests. En effet la LCE n'est pas assez discriminante dans le cas où des objets d'intérêt ne sont pas représentés dans la partition automatique. Le prise en compte du raffinement mutuelle peut engendrer une erreur trop faible pour des partitions pourtant relativement différentes.

Les partitions manuelles permettant d'effectuer la comparaison sont séparées en 2 catégories pour le test. La première regroupe les partitions considérées comme grossières ne comportant que très peu de régions (en moyenne 10 régions par partition) (cf. figure 3.26.a et b). La deuxième comporte les partitions détaillées comportant un plus grand nombre de régions (en moyenne 33 régions par partition)(cf. figure 3.26.c et d).

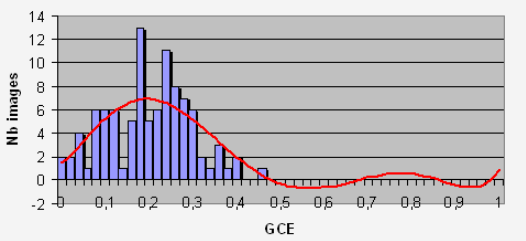
Une forte valeur de la GCE indique soit des fuites de l'objet vers le fond dans la partition automatique soit que l'objet d'intérêt n'a tout simplement pas été détecté. De faibles valeurs indiquent cependant une forte cohérence entre les partitions et une décomposition de l'objet d'intérêt en régions d'intérêt de qualité. La figure 3.26 présente les distributions de la GCE calculée sur 100 images tests pour les 2 mélanges H_{TRS} et H_{Rab}. Les partitions obtenues avec le mélange H_{Rab} contiennent en moyenne 25 régions (cf. figure 3.26.a et c) et 24 pour le mélange H_{TRS} (cf. figure 3.26.b et d).

La figure 3.26.e présente un test effectué sur les partitions obtenues avec le mélange H_{Rab} comparées aléatoirement. L'objectif est de montrer le comportement de la GCE pour des partitions indépendantes qui ne sont *a priori* pas cohérentes. Dans cette dernière figure, les faibles valeurs sont obtenues pour des partitions comportant un nombre de régions très (trop) différent. Ce qui est interprété comme un raffinement par le test. Cependant, nous pouvons constater que la moyenne de la GCE est plus élevée $(0,36)$ et que les valeurs sont plus étalées que pour les 4 histogrammes précédents.

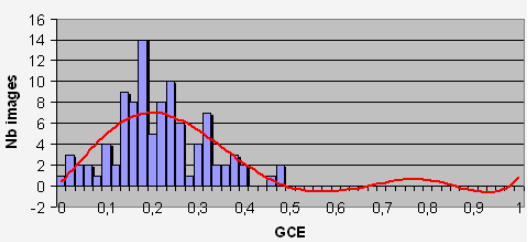
Statistiquement, la qualité de raffinement des partitions obtenues avec le mélange H_{TRS} et celles obtenues avec le mélange H_{Rab} est similaire. L'allure de l'histogramme de la figure 3.26.a est comparable à celui de la figure 3.26.b. Cependant, à GCE égale, ce ne sont pas forcément les mêmes images concernées. Il faut plutôt considérer ces deux espaces comme des espaces complémentaires. C'est pourquoi nous avons préféré présenter les résultats des 2 mélanges.

Conclusion

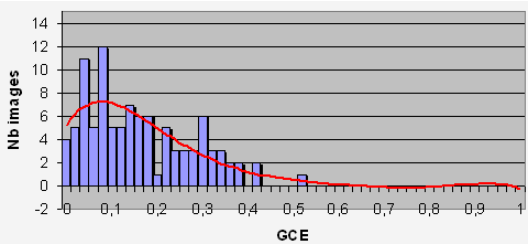
L'utilisation de la segmentation locale initialisée à l'aide de la technique de la H-image permet de limiter grandement la sur-segmentation. Il est rare que des parties d'objets différents fusionnent. Il arrive cependant, lors de la création de la H-image, que l'objet ne soit pas détecté et qu'aucune racine ne le représente. Dans ce cas l'objet est confondu avec le fond. Généralement, même si l'objet n'est pas défini par une seule région, les régions qui le composent correspondent bien à la forme de l'objet et les contours sont de bonne qualité. Cependant, il apparaît encore trop de régions définissant les objets d'intérêt et le fond. Nous proposons d'effectuer, après l'étape de la segmentation locale, une étape supplémentaire de groupements hiérarchiques des régions obtenues. Cette étape fait l'objet du paragraphe suivant.



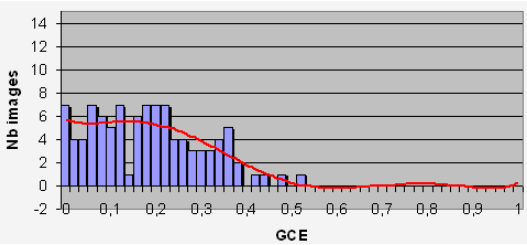
(a) GCE des partitions obtenues avec le mélange H_{Rab} et des partitions manuelles détaillées (moyenne $= 0,21$)



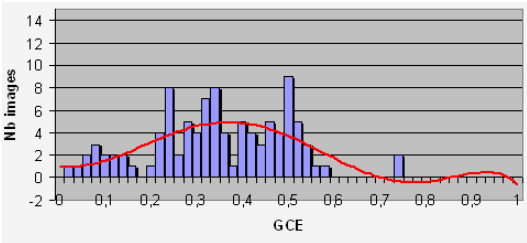
(b) GCE des partitions obtenues avec le mélange H_{TRS} et des partitions manuelles détaillées (moyenne $= 0,23$)



(c) GCE des partitions obtenues avec le mélange H_{Rab} et des partitions manuelles simplistes (moyenne $= 0,17$)



(d) GCE des partitions obtenues avec le mélange H_{TRS} et des partitions manuelles simplistes (moyenne $= 0,19$)



(e) GCE de partitions correspondant à différentes images (moyenne $= 0,36$)

FIG. 3.26: Distribution de l'erreur de cohérence globale (GCE) calculée sur 100 images tests de la base de données de segmentation manuelle de [80] (en rouge : courbe de tendance du troisième ordre)

3.3.2 Groupements hiérarchiques de régions

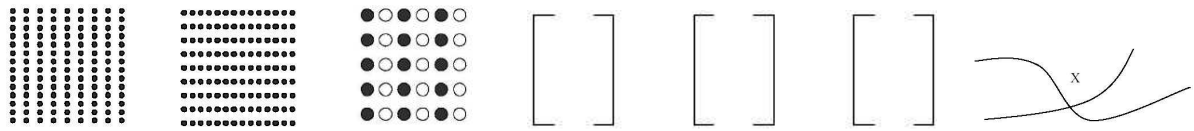
概念

La segmentation locale automatique limite le nombre de régions présentes dans la partition finale. Cependant, les objets d'intérêt sont encore trop souvent décomposés en quelques régions. J'aborde ici une méthode de fusion des régions obtenues par la pyramide locale visant à reconstruire les objets d'intérêt.

Nous proposons ici de nous intéresser aux groupements des régions obtenues après la segmentation locale. La méthode de groupement doit être sans connaissance *a priori* du contenu de l'image. Plusieurs systèmes fondés sur cette approche de généricité ont été développés [81, 122, 98]. Afin de rester générique, de nombreuses méthodes s'inspirent de la théorie du *Gestalt* introduite par Wertheimer [124] selon laquelle la perception visuelle humaine crée, à partir de certains critères, des groupements successifs : les *gestalts*.

Théorie du Gestalt

Les propriétés de base de la théorie du Gestalt sont : la proximité, la similarité, la continuité, la fermeture (d'une courbe), la symétrie ... Ces propriétés sont illustrées dans la figure 3.27. Dans la figure 3.27 (a), les points sont en fait perçus comme une juxtaposition de co-



(a) Proximité (b) Similarité (c) Fermeture (d) Continuité

FIG. 3.27: Illustration des propriétés du Gestalt de regroupement (d'après [135])

lonnes (à gauche) ou de lignes (à droite) grâce à la propriété de proximité. Dans la figure 3.27 (b), nous percevons une série de colonnes de points similaires en couleur. Dans la figure 3.27 (c), bien que les segments soient espacés de la même distance, la propriété de fermeture fait que nous regroupons les segments 2 à 2 de façon à ce qu'ils soient perçus comme des objets fermés. Dans la figure 3.27 (d), nous percevons deux courbes continues qui se croisent en X au lieu de percevoir deux courbes en V se touchant en X.

D'autres principes de groupement ont été étudiés par la suite. Par exemple, Gaetano Kanizsa propose dans [62] une liste de règles de groupement légèrement différente par rapport à celle de Wertheimer. Pour plus de détails voir [33]. Brunswik [20] évoque le fait que les différents critères de groupement issus de la théorie du Gestalt reflètent les statistiques des scènes naturelles. C'est pourquoi il est intéressant de considérer ces propriétés afin de guider un procédé de groupement itératif. Martin *et al.* proposent dans [80] d'étudier les statistiques de ces propriétés de groupement sur leur base de données d'images naturelles segmentées manuellement.

Vasseur [122] et Maßmann [81] proposent un groupement utilisant les contours. Luo [77] et Pardo [98] proposent, quant à eux, d'utiliser une approche régions pour les groupements. C'est cette approche qui nous semble la plus appropriée à notre méthode. Dans [122], la pertinence

des groupements est calculée par la théorie de Dempster-Shafer tandis que dans [81] et [77] la fusion est effectuée selon la méthode des chaînes de Markov aléatoires (*RMF Random Markov Fields*). Dans [98], la pertinence visuelle permet de construire un *BPT* [105] (*Binary Partition Tree*) approchant une représentation sémantique de l'image. L'utilisation du *BPT* permet à l'utilisateur de manipuler et d'améliorer facilement la partition proposée par l'algorithme.

Ainsi, la théorie du Gestalt n'implique aucun modèle d'objet, elle utilise uniquement la pertinence visuelle des régions seules et de leur groupement éventuel. Ainsi cette théorie est bien adaptée aux méthodes génériques d'extraction d'objets, qui se veulent sans connaissance *a priori* du contenu de l'image. Le problème est que de ce fait, une partition visuellement pertinente n'est pas forcément, voire rarement, assimilable à une partition sémantique. Il faudrait pour cela utiliser des critères de plus haut-niveau bien plus complexes à mettre en œuvre.

Cependant, dans le cas où les régions à regrouper décrivent avec précision des sous-parties d'objet, les propriétés de continuité et de symétrie semblent bien adaptées pour guider les fusions permettant de reconstruire l'objet sémantique. Nous proposons donc un procédé de groupement itératif des régions issues de la segmentation locale fondé sur ces deux propriétés. L'objectif est de montrer que les régions issues de la segmentation locale fournissent un support de qualité pour de tels groupements. En effet, le nombre de régions obtenues est généralement assez faible et les régions correspondent relativement bien à des sous parties d'objets sémantiques. La structure en graphes d'adjacences (RAG : *Region Adjacency Graph*) de la pyramide se trouve être en accord avec la propriété de proximité issue du Gestalt. Ainsi, le dernier étage de la pyramide locale et son graphe d'adjacence correspondant, constituent le premier étage de l'étape de groupement fondé sur des critères visuels. A chaque itération le *RAG* est réduit par la fusion des 2 sommets constituant la meilleure fusion au sens des propriétés du Gestalt.

Continuité

La continuité entre deux régions constitue une propriété très intéressante dans un procédé de groupement générique. Une fonction de coût C_c a été élaborée pour favoriser la fusion de régions qui vérifient une jonction en "T"[6] et qui ont des orientations globales similaires.

Dans un premier temps, le principe est d'estimer les angles formés par les contours localisés de part et d'autre de la frontière entre les 2 régions pour déterminer le type de jonction existant entre elles (jonction en "T" ou en "Y"[6]). Pour cela, le contour de chaque région voisin de la frontière est modélisé sur quelques pixels par un segment obtenu par régression linéaire. Une frontière induit 4 segments. Cette modélisation, bien que simpliste, fournit des résultats satisfaisants. Ce principe est illustré par la figure 3.28 où a_1, a_2, a_3 et a_4 sont les 4 angles formés par les 4 couples de segments estimés (en rouge). Les angles sont compris entre 0 et π.

La continuité des couples de segments de chaque jonction est exprimée par le coefficient C introduit dans [77] et donné par l'expression suivante :

$$C = \frac{\sum_{i=1}^{2*N_f} a_i}{2\pi * N_f} \tag{3.23}$$

Où N_f représente le nombre de frontières communes (pointillés dans la figure 3.28) entre les 2 régions. Les valeurs de C sont comprises entre 0 et 1. La valeur 1 indique une forte continuité (jonction en "T"), tandis que la valeur 0 indique un groupement indésirable (jonction en "Y").

[6] La barre verticale du "T" ou du "Y" représente la frontière entre les 2 régions

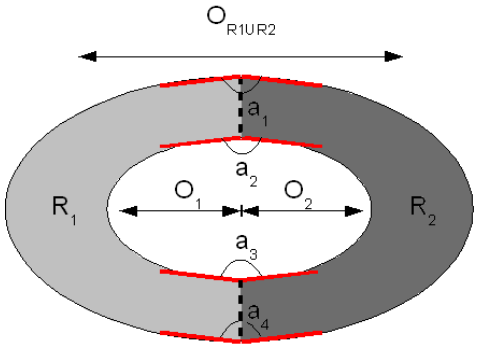


FIG. 3.28: Illustration de la continuité entre 2 régions R_1 et R_2

Dans un deuxième temps, les orientations de chacune des 2 régions permettent de pondérer le coefficient C pour favoriser la fusion de régions orientées selon une direction similaire. Ce poids noté w_o, consiste à évaluer la correspondance des orientations des 2 régions afin de construire la fonction de coût C_c. L'orientation d'une région correspond à l'orientation de l'axe principal de son ellipse d'inertie [58].

Soient O_1 et O_2 les orientations respectives de R_1 et R_2 comprises entre 0 et π. Soit $O_{R_1 \cup R_2}$ l'orientation de la région formée par la réunion de R_1 et R_2. Le poids est donné par la relation suivante :

$$w_o = 1 - \frac{\Delta_o}{\pi/2} \tag{3.24}$$

avec Δ_o donnée par l'équation 3.25 :

$$\Delta_o = \frac{|O_1 - O_{R_1 \cup R_2}| + |0_2 - O_{R_1 \cup R_2}|}{2} \tag{3.25}$$

Le coût d'une fusion entre 2 régions selon le critère de la continuité est donné par l'expression suivante :

$$C_c = w_o \cdot C \tag{3.26}$$

Les valeurs de C_c sont comprises entre 0 et 1. Une faible valeur de C_c indique une forte continuité entre les 2 régions. Tandis qu'une valeur proche de 0 indique une fusion non désirée.

Symétrie

Une deuxième fonction de coût C_s a été développée pour favoriser l'obtention de régions de formes symétriques durant les étapes de groupement.

Nous proposons de quantifier la symétrie du masque d'une région issue d'un groupement afin de construire la fonction de coût C_s. Cette région est tout d'abord modélisée par une ellipse d'inertie [58] donnant l'orientation globale de la région. L'objectif est alors de déterminer si la symétrie de la région est plutôt centrale ou axiale. Dans ce dernier cas on précisera par rapport à quel axe. Le symétrique de chaque pixel de la région est calculé par rapport au grand axe, au petit axe puis au centre de gravité. L'objectif est de déterminer la transformation donnant le plus faible pourcentage de pixels sortant de la surface de la région. Ainsi, on calcule un coefficient s_i pour chacune des 3 symétries ($i \in [\![1,\ 3]\!]$). s_i mesure le pourcentage de pixels possédant son symétrique au sein de la surface de la région selon la $i^{\text{ème}}$ symétrie. La fonction de coût fondée sur la symétrie C_s est donnée par la relation suivante :

$$C_s = \min_i(s_i)\ i \in [\![1,\ 3]\!] \tag{3.27}$$

Sélection des groupements

La pertinence d'un groupement de 2 régions est évaluée à partir de la continuité entre ces régions C_c et de la symétrie C_s de la forme du groupement résultant. L'objectif de cette étape est de sélectionner le meilleur groupement pour effectuer une seule fusion par niveau.

Selon le critère de proximité de la théorie du Gestalt, les groupements ne sont naturellement évalués qu'entre les régions voisines possédant une frontière. Pour cela, on utilise la structure du graphe d'adjacence de la pyramide irrégulière. A chaque arc du graphe d'adjacence est associé un coût de groupement. Ce coût noté C_G, est donné par l'équation suivante :

$$C_G = \sqrt{C_c^2 + C_s^2} \tag{3.28}$$

Du fait que les partitions ne possèdent que très peu de régions, l'étape de décimation du graphe d'adjacence ne procède qu'à 1 fusion par niveau. Cette fusion est bien-sûr donnée par le coût de groupement le plus faible du graphe d'adjacence. Chaque fusion donne une nouvelle partition et donc un niveau supplémentaire dans la pyramide, en effectuant seulement le meilleur groupement global de la partition courante. Une fois la fusion effectuée, les coûts de fusion entre la nouvelle région et ses voisines sont calculés. Ce procédé itératif est effectué jusqu'à qu'il n'y ait plus qu'une seule région dans la partition.

Conclusion et résultats

Nous proposons donc une étape supplémentaire de groupements hiérarchiques initialisée à partir d'une partition issue d'une première segmentation. Le résultat final se trouve être un empilement de partitions correspondant aux fusions successives utilisant la théorie du Gestalt. Une fois le traitement effectué, l'utilisateur peut rechercher parmi les empilements successifs proposés, la partition qu'il jugera la plus sémantique.

La figure 3.29 présente différents résultats de groupements correspondant aux meilleures partitions sémantiques des empilements. Pour que l'on puisse parler de sémantique ici, ces dernières ont été bien sûr choisies manuellement parmi l'empilement de partitions. Nous pouvons remarquer que les deux propriétés de continuité et de symétrie, permettent d'ordonner de manière intéressante les fusions des régions issues de la segmentation locale. Les partitions obtenues ne comportent que très peu de régions. De plus, les objets décomposés en plusieurs régions de qualité par la segmentation locale sont effectivement reconstruits petit à petit durant les fusions successives.

Néanmoins, ce type de méthode comporte plusieurs inconvénients :

Le temps de traitement : Le temps de calcul des caractéristiques de chaque groupement de la partition est généralement important bien qu'il dépende du nombre de régions de la partition initiale. Cette caractéristique rend difficile l'utilisation d'une telle méthode dans le cadre de très grandes bases de données ou de l'étude de séquences vidéo nécessitant un traitement en temps réel.

La sélection de la partition finale : Le résultat fourni par l'étape de groupement est un empilement de partitions. Il est difficile de proposer une méthode automatique de sélection de la meilleure partition. Le but étant de fournir une partition sémantique, il est préférable de laisser ce choix à l'utilisateur. Il serait bien sûr envisageable de poser une contrainte permettant de stopper le procédé de groupement telle que le nombre minimum de régions à atteindre, leur taille ou le coût des fusions .... Cependant, cette condition d'arrêt est difficilement maîtrisable. De plus, il est possible que l'empilement comporte plusieurs partitions intéressantes pour l'utilisateur. Dans le cas, par exemple où l'image comporte plusieurs objets, rien ne garantit le fait qu'ils soient tous reconstitués de manière correcte dans la même partition.

La robustesse : La qualité des groupements est fortement liée à la qualité des régions de la partition initiale $\mathcal{P}$. La diversité des régions à regrouper est telle que la méthode de groupement n'est pas forcément robuste d'une partition initiale à une autre.

Il serait intéressant de pouvoir évaluer la qualité de ce type de résultats cependant, elle reste relativement subjective et le fait de devoir choisir manuellement la partition parmi l'empilement hiérarchique rend difficile l'élaboration d'un test de qualité.

Remarque : Dans [56], nous avions présenté une méthode de groupement hiérarchique sur un plus grand nombre de critères visuels tels que la compacité, la convexité ou la différence de couleur. Dans [65] il est par exemple stipulé que la perception visuelle est attirée par des régions compactes. Il s'avère que plus la dimension de l'espace des critères augmente, plus leur contribution diminue dans la classification de l'espace. Il est alors de plus en plus difficile de contrôler les fusions. De plus nous avons remarqué que les critères de continuité et de symétrie suffisent à guider convenablement les groupements.

Afin de pallier certains inconvénients vus ci-dessus dus au manque d'information sémantique dans l'initialisation de la segmentation, nous allons maintenant explorer une autre méthode utilisant l'information du mouvement dans les séquences vidéo.

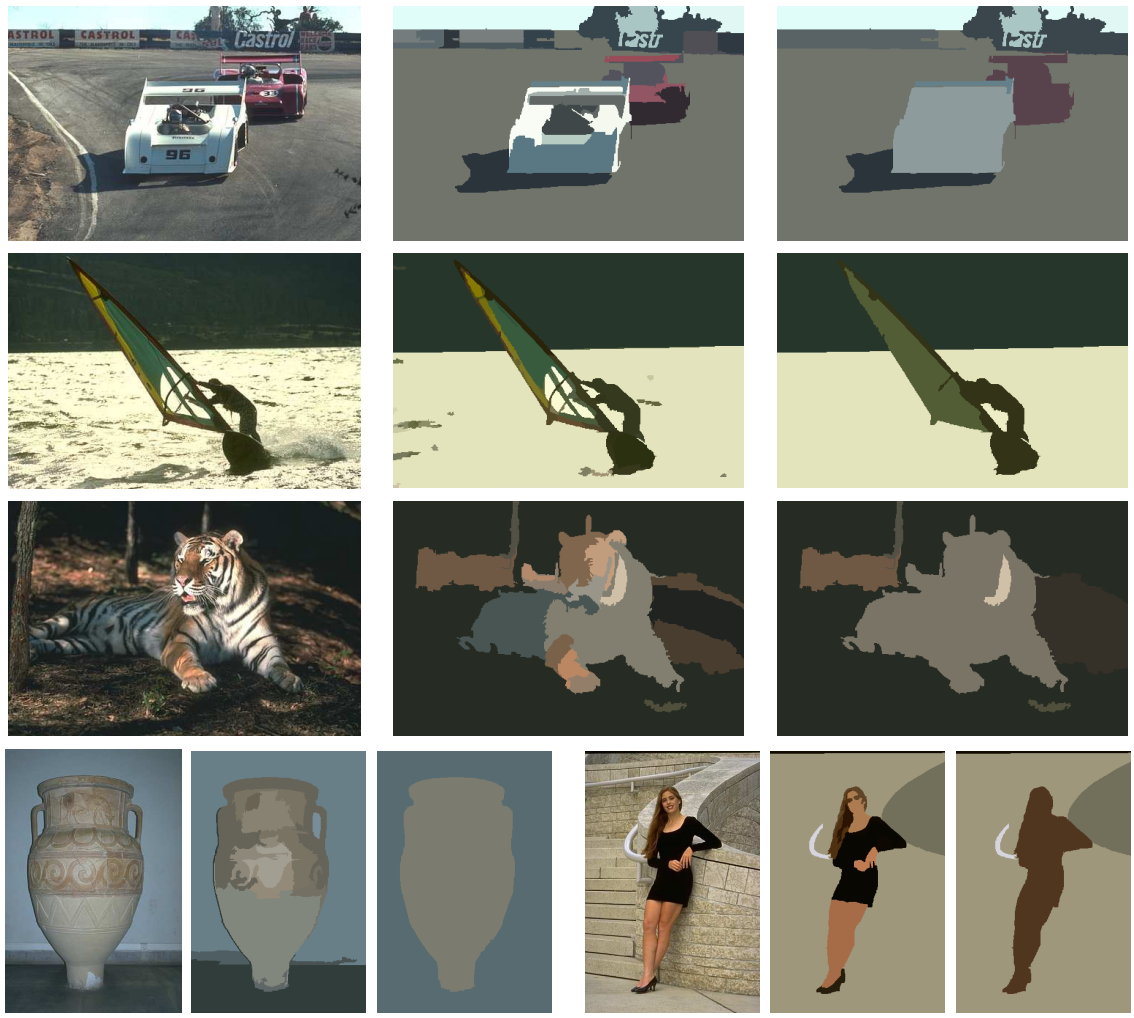


FIG. 3.29: Résultats de l'étape de groupement hiérarchique. De gauche à droite : l'image originale, la partition obtenue par segmentation locale automatique et la partition correspondant aux meilleurs groupements

3.4 Initialisation temporelle automatique - critère inter images

概念

Cette section présente l'utilisation de la segmentation locale dans un contexte vidéo. La segmentation est focalisée sur les régions du premier plan animées d'un mouvement différent du mouvement global de la caméra.

Le principe de cette initialisation est d'exploiter l'information temporelle disponible dans les séquences vidéo pour localiser les zones de segmentation automatiquement dans les zones d'intérêt de l'image. L'objectif est de fournir une méthode totalement automatique présentant une partition unique à l'utilisateur. L'information temporelle est considérée comme un fort indicateur sémantique. L'extraction est focalisée sur les régions animées d'un mouvement différent du mouvement global de la caméra entre deux images successives. Ces régions sont appelées régions du premier plan par opposition au fond animé du mouvement global. Elles constituent notre modèle d'objet extractible car elles sont supposées issues d'objets d'intérêt de la séquence.

Le procédé de segmentation locale intervient pour extraire à partir de ces régions une région d'intérêt (ROI : *Region Of Interest*) approchant au mieux le VOP de l'objet d'intérêt. Voici les 4 étapes permettant de résumer le procédé d'extraction :

1. Estimation du champ de vecteurs du mouvement apparent $\overrightarrow{V_a}$ entre 2 images. Les images utilisées peuvent être successives ou éloignées temporellement. Ceci permet de s'adapter à la rapidité du mouvement dans les séquences. L'estimation de mouvement est réalisée à l'aide de l'algorithme du *block-matching* rapide appelé *Blocks Sum Pyramid Algorithm* (BSPA). Ce dernier est décrit dans la suite de cette section.
2. Estimation de champs de vecteurs du mouvement global $\overrightarrow{V_g}$ induit par la caméra à partir de $\overrightarrow{V_a}$.
3. Détermination des régions du premier plan par compensation du mouvement global.
4. Extraction des ROIs par segmentation localisée dans les régions du premier plan selon le principe de l'initialisation par ruban. La zone de segmentation et les racines sont construites automatiquement en prenant en compte la forme des régions du premier plan.

3.4.1 Estimation du champ de vecteurs mouvement par *block-matching*

Pour cette initialisation de la segmentation spatio-temporelle, nous avions besoin d'une méthode rapide et efficace d'estimation de champ de vecteurs mouvement entre deux images. Nous avons choisi le *Block Sum Pyramid Algorithm* (BSPA) dont nous connaissions déjà l'efficacité au niveau de la méthode de suivi d'objets développée par Guillaume Foret [40].

Le *block-matching* est une méthode grandement utilisée dans la compression vidéo, le suivi d'objet et de nombreux autres types de traitements de séquences vidéo.

Nous allons ici nous intéresser au BSPA [67], mis en œuvre par Chang-Hsing Lee et Ling-Hwei Chen. Il est fondé sur une estimation rapide de mouvement appelée l'algorithme d'élimination successive soit en anglais le SEA (*Successive Elimination Algorithm*). Ce dernier permet d'obtenir la même estimation que le classique *Full Search Algorithm* (FSA) tout en diminuant le temps de calcul.

Avant d'expliquer plus en détails le BSPA faisons, tout d'abord, un bref rappel de la technique de base du *block-Matching*.

Technique du *block-matching*

Le *block-matching* a été conçu pour estimer le mouvement entre deux images d'une séquence vidéo. Usuellement, l'image de référence est divisée en blocs contigüs. Tous les pixels appartenant à un même bloc sont alors traités comme ayant un seul et même vecteur mouvement. Le but est de trouver les pairs de blocs appartenant respectivement à l'image de référence et à l'image courante qui minimisent l'erreur de mise en correspondance. Cette erreur entre le bloc à la position $(x,\ y)$ dans l'image de référence I_{ref} et le bloc candidat à la position $(x+u,\ y+v)$ dans l'image courante I_t est généralement définie comme la somme des différences absolues (SAD) :

$$SAD_{(x,y)}(u,v) = \sum_{j=0}^{B-1}\sum_{i=0}^{B-1} |I_{ref}(x+i, v+j) - I_t(x+u+i, y+v+j)| \tag{3.29}$$

Où les blocs sont de la taille $B \times B$ pixels.

La meilleure estimation du vecteur mouvement $(\widehat{u},\ \widehat{v})$ est obtenue pour le couple $(u,\ v)$ minimisant la $SAD_{(x,y)}(u,v)$. Cette estimation peut-être effectuée par l'algorithme du *full-search* (FS) qui calcule la SAD pour toutes les positions possibles $(x+u,\ y+v)$ dans une fenêtre de recherche centrée sur la position du bloc dans l'image courante I_t. Ce qui peut aussi s'exprimer de la manière suivante :

$$(\widehat{u},\ \widehat{v}) = arg \min_{(u,v)\in S(x,y)} \left\{SAD_{(x,y)}(u,v)\right\} \tag{3.30}$$

Où S est l'ensemble de recherche déterminé par un entier R conformément à l'expression suivante :

$$S(x,\ y) = \{(u,\ v)| -R \leqslant u,v \leqslant R \text{ et } (x+u,\ y+v) \text{ une position valide de } I_t\} \tag{3.31}$$

Le principal inconvénient de cette méthode est qu'étant exhaustive elle est très coûteuse en calcul ce qui la rend difficile à intégrer dans des procédés temps réel. De cet algorithme sont nés plusieurs autres versions appelées algorithmes rapides auxquels nous nous intéressons dans l'annexe C. Parmi ces techniques nous avons choisi le BSPA [67] (*Block Sum Pyramid Algorithm*) pour notre méthode. Son fonctionnement est détaillé dans la section C.2 de l'annexe C.

3.4.2 Estimation du mouvement global

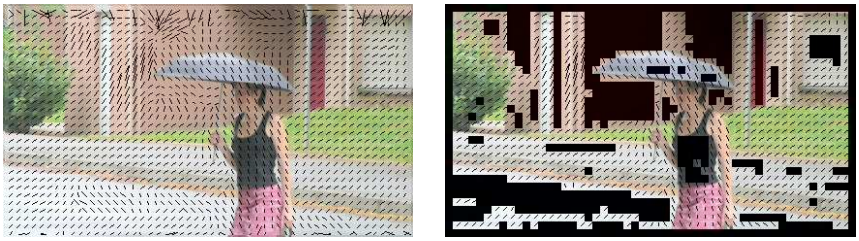
Le champ de vecteurs $\overrightarrow{V_a}$ qui associe un vecteur mouvement à chaque bloc de l'image est désormais connu. L'objectif est désormais d'estimer le mouvement global dans l'image courante à partir de $\overrightarrow{V_a}$. Cependant, les vecteurs de ce dernier ne sont pas tous d'une fiabilité équivalente. Ainsi, seul un sous-ensemble de vecteurs de $\overrightarrow{V_a}$ est utilisé pour l'estimation du mouvement global. Ce sous-ensemble doit être constitué des vecteurs de confiance.

Vecteurs mouvement de confiance

Un inconvénient intrinsèque aux méthodes de *block-matching* est la fiabilité de la correspondance entre blocs homogènes soumis au problème bien connu d'ouverture [89]. Les vecteurs mouvement obtenus dans les zones homogènes sont aléatoires comme le montre les distortions du champ de vecteur au niveau de telles zones, dans la figure 3.30.a. De plus, les vecteurs issus des blocs situés au bord de l'image sont généralement erronés à cause du mouvement de la caméra qui induit un recouvrement du fond. Ainsi, les paramètres du mouvement global sont estimés uniquement à partir des vecteurs de confiance issus des blocs hétérogènes, représentés

par les blocs d'écarts-types élevés, et qui ne bordent pas l'image (cf. figure 3.30.b). On notera B_c, l'ensemble des blocs de confiance et $\overrightarrow{V_c}$ l'ensemble de vecteurs associés.

L'ensemble B_c est déterminé par une valeur d'écart-type minimum paramétrable par l'utilisateur. Au cours de nos tests, un écart-type d'une valeur de 10 suffit à obtenir une estimation correcte sur l'ensemble des séquences vidéo utilisées. La détermination automatique de ce seuil mériterait une étude statistique sur plusieurs vidéos. Cependant, ceci nécessite une vérité terrain sur les champs de vecteurs dont nous ne disposions pas. Il aurait été intéressant d'effectuer des travaux de recherche sur ce sujet ; le manque de temps nous a décidé à utiliser un seuil empirique permettant toutefois d'améliorer grandement les résultats. Les blocs homogènes sont considérés comme incertains est ne seront donc déterminés comme appartenant au premier plan ou au fond, uniquement lors de l'extraction des ROIs.



(a) Estimation de mouvement (b) Blocs non fiables (en noir)

FIG. 3.30: Détermination des vecteurs de confiance (mouvement global dirigé vers le bas à gauche)

Pour estimer le mouvement global à partir de $\overrightarrow{V_c}$, il est nécessaire de choisir le modèle de mouvement qui retranscrit au mieux les mouvements 2-D dits « apparents » que nous désirons détecter et qui aura le meilleur rapport *efficacité/temps de calcul* pour notre application. Le mouvement apparent, observé dans le plan de l'image est une projection du mouvement réel 3D de la scène filmée.

Il existe un grand nombre de modèles différents dont la complexité augmente avec le nombre de paramètres d'estimation qu'ils utilisent. Plus le modèle est complexe et plus le système détectera des mouvements variés. Cependant cette efficacité s'accompagne d'un coût élevé en calcul et la sensibilité aux erreurs numériques s'en trouve de même accrue. Le choix du modèle de mouvement détermine également le modèle des objets d'intérêt extractibles par l'algorithme. Nous allons maintenant faire une brève présentation de ces différents modèles, puis nous verrons la méthode que nous avons utilisée conformément à [53] pour estimer le mouvement global et détecter les objets du premier plan.

Choix du modèle

- **Modèle de mouvement constant**

  Les déplacements sont modélisés de la manière suivante :

$$\begin{pmatrix} \mathrm{dx} \\ \mathrm{dy} \end{pmatrix} = \begin{pmatrix} \mathrm{tx} \\ \mathrm{ty} \end{pmatrix} \tag{3.32}$$

  tx et *ty* représentent les composantes des translations suivant respectivement l'axe horizontal et l'axe vertical de l'image. Ce modèle est celui qui a été adopté dans les normes de codage MPEG1 et MPEG2.

- **Modèle de mouvement linéaire simplifié ou affine**

  Les déplacements sont modélisés de la manière suivante [50] :

$$\begin{pmatrix} \text{dx} \\ \text{dy} \end{pmatrix} = \begin{pmatrix} C_F - 1 & -C_F\varphi_z \\ C_F\varphi_z & C_F - 1 \end{pmatrix} \cdot \begin{pmatrix} \text{x} \\ \text{y} \end{pmatrix} + \begin{pmatrix} \text{tx} \\ \text{ty} \end{pmatrix} \tag{3.33}$$

  Le mouvement est ici défini par 4 paramètres qui sont :
  - C_F représentant le facteur de « zoom » avant ou arrière dans la séquence. Ce paramètre est très dépendant de la distance focale de la caméra.
  - φ_z traduisant la rotation autour de l'axe optique de la caméra.
  - tx et ty représentant les composantes des translations suivant respectivement l'axe horizontal et l'axe vertical de l'image.

  Dans ce modèle, on considère que la scène bénéficie d'une profondeur de champ assez restreinte et que les rotations autour de l'axe optique observées d'une image à l'autre sont faibles.

- **Modèle linéaire de mouvement**

  Les composantes du déplacement dx et dy sont modélisées de la façon suivante :

$$\begin{pmatrix} \text{dx} \\ \text{dy} \end{pmatrix} = \begin{pmatrix} C_F - 1 + Hyp1 & Hyp2 - C_F\varphi_z \\ Hyp2 + C_F\varphi_z & C_F - 1 - Hyp1 \end{pmatrix} \cdot \begin{pmatrix} \text{x} \\ \text{y} \end{pmatrix} + \begin{pmatrix} \text{tx} \\ \text{ty} \end{pmatrix} \tag{3.34}$$

  Ce modèle rajoute au modèle linéaire simplifié les paramètres $Hyp1$ et $Hyp2$ simulant la contraction du champ de mouvement respectivement en x et en y.

- **Modèle quadratique de mouvement**

  Le modèle quadratique de mouvement s'intéresse aux mouvements non-linéaires. Il est donc principalement utilisé pour la modélisation de mouvements d'objets déformables. Jusqu'à présent, les modèles ne prenaient en compte que les premiers termes du développement de Taylor du déplacement. Il s'agit maintenant d'effectuer un développement jusqu'au deuxième ordre. On introduit donc six paramètres supplémentaires. Ce modèle se formalise par le système suivant :

$$\begin{pmatrix} \text{dx} \\ \text{dy} \end{pmatrix} = \begin{pmatrix} a1 & a2 \\ b1 & b2 \end{pmatrix} \cdot \begin{pmatrix} \text{x} \\ \text{y} \end{pmatrix} + \begin{pmatrix} a3 & a4 \\ b3 & b4 \end{pmatrix} \cdot \begin{pmatrix} x^2 \\ y^2 \end{pmatrix} + \begin{pmatrix} \text{a5} \\ \text{b5} \end{pmatrix} .xy \tag{3.35}$$

L'objectif de notre application est de modéliser le mouvement global de la séquence afin de pouvoir localiser par la suite avec plus ou moins de précision, les objets qui ne vérifient pas ce mouvement. Pour le mouvement global, nous nous limiterons comme dans [117] aux mouvements de translation, de rotation et aux zooms en sachant, de plus, que la plupart des vidéos vérifient les hypothèses d'une rotation faible et d'une profondeur de champ restreinte.

Nous choisirons donc le modèle affine simplifié appelé également modèle de « transformation rigide ». Ce modèle est celui dont le rapport *efficacité/temps de calcul* correspond le mieux à notre application. Passons maintenant à la présentation de la méthode de résolution permettant de déterminer les paramètres du modèle de mouvement.

Résolution du système de paramètres

Récrivons de manière simplifiée la transformation rigide de l'équation 3.33. Les déplacements dx et dy deviennent des coordonnées : x, y, x'' et y'' où le couple $(x_i'',\ y_i'')$ représente la position prédite par le modèle de mouvement global du bloc centré en $(x_i,\ y_i)$ dans l'image

de référence. Les expressions des paramètres sont également simplifiées en a_1, a_2, a_3 et a_4. L'équation 3.33 devient :

$$\begin{pmatrix} x''_i \\ y''_i \end{pmatrix} = \begin{pmatrix} a_1 & a_2 \\ -a_2 & a_1 \end{pmatrix} . \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} a_3 \\ a_4 \end{pmatrix} \tag{3.36}$$

Ce modèle est associé à une fonction de coût Φ qui traduit l'erreur quadratique entre les positions $(x'_i,\ y'_i)$ estimées par le *block-matching* et les positions $(a_1x_i - a_2y_i + a_3 - x'_i,\ a_2x_i + a_1y_i + a_4 - y_i)$ prédites par le modèle. Cette fonction de coût est définie par :

$$\Phi = \sum_{i=1}^{N} [(a_1x_i - a_2y_i + a_3 - x'_i)^2 + (a_2x_i + a_1y_i + a_4 - y'_i)^2] \tag{3.37}$$

Où N est le nombre de blocs déterminés par le découpage de l'image dans le *bloc-matching*.

Le but est de trouver l'ensemble de paramètres a_1, a_2, a_3 et a_4 qui minimise cette erreur sur l'ensemble des blocs de B_c. Pour cela on résout un système matriciel de la forme $Y = a.A$ à l'aide de la méthode des moindres carrés. Ici, Y représente la position estimée des blocs dans l'image courante et a représente la matrice à estimer contenant les paramètres. Pour nous affranchir de la non inversibilité de la matrice A, nous procédons à sa décomposition en valeur singulière (SVD). Le code que nous avons utilisé pour mettre en œuvre ces calculs est inspiré de celui disponible dans les *Numerical Recipes* [36].

Blocs du premier plan

La détermination des paramètres du mouvement global et du champ de vecteurs V_g associé permet de prédire la position dans l'image courante de chaque bloc de B_c de l'image de référence. Soient $(x''_i,\ y''_i)$ les coordonnées prédites en fonction du mouvement global pour le bloc de coordonnées (x_i, y_i). La compensation de mouvement permet de déterminer l'ensemble des distances Euclidiennes d_i entre les coordonnées $(x''_i,\ y''_i)$ et $(x'_i,\ y'_i)$.

$$d_i = \sqrt{(x''_i - x'_i)^2 + (y''_i - y'_i)^2} \tag{3.38}$$

Le problème de la détermination des blocs du premier plan revient à trouver le seuil T_0 permettant de scinder B_c en deux groupes B_{P_0} et B_{F_0} tels que :

$$B_{P_0} = \{b_i \in B_c / d_i > T_0\} \tag{3.39}$$

$$B_{F_0} = \{b_i \in B_c / d_i \leq T_0\} \tag{3.40}$$

B_{P_0} et B_{F_0} correspondent respectivement au groupe des blocs du premier plan et des blocs du fond déterminé par T_0.

Pour résoudre ce problème, on considère que la surface des régions du premier plan est plus faible que celle du fond. Ainsi, la population des d_i comporte en majorité des valeurs proche de 0. Si l'on considère l'histogramme des distances : le premier mode qui est aussi le plus important symbolise les blocs du fond. L'objectif est de modéliser par une Gaussienne la population des d_i provenant de ces blocs.

Dans un premier temps, on émet l'hypothèse selon laquelle 70% de l'image est animée du mouvement global. C'est pourquoi seule les d_i les plus faibles représentant 70% de la population totale des distances est modélisée par une Gaussienne de moyenne μ_0 et d'écart-type σ_0. Les blocs du premier plan correspondent aux blocs dont la distance d_i dépasse le seuil $T_0 = \mu_0 + \sigma_0$.

Dans le cas où l'hypothèse n'est pas vérifiée, c'est-à-dire que les blocs du fond représentent une surface supérieure à 70%, les distances dues aux blocs du fond restent relativement compactes et proches de 0. Ainsi, la prise en compte de l'écart-type permet de classer correctement

les blocs du fond. Seuls quelques blocs isolés peuvent être mal classés mais ils sont aisément filtrables par un traitement morphologique simple. Par contre, le fait d'utiliser seulement un pourcentage de la population des distances pour la modélisation, permet d'écarter dès le départ des distances importantes dues aux blocs du premier plan qui auraient tendance à fausser la classification.

Itérations

Afin de raffiner les ensembles B_{F_0} et B_{P_0}, dans le cas où les régions du premier plan sont de tailles plus importantes par exemple, nous réitérons l'estimation du mouvement global mais cette fois uniquement sur les blocs qui ont été déclarés comme appartenant au fond. A la première itération c'est donc B_{F_0} qui permet d'estimer le nouveau mouvement global. Ceci permet d'écarter du procédé d'estimation les blocs ayant une forte probabilité d'appartenir au premier plan. La $i^{\text{ème}}$ itération détermine un nouveau seuil $T_i = \mu_i + \sigma_i$, où μ_i et σ_i sont respectivement la moyenne et l'écart-type de $B_{F_{i-1}}$. T_i permet donc de déterminer B_{F_i} et B_{P_i} à partir de $B_{F_{i-1}}$. Le procédé d'itération s'arrête dès que le nombre de blocs de B_{F_i} et de $B_{F_{i-1}}$ est identique.

3.4.3 Détermination du masque de segmentation

概念

La détermination des blocs du premier plan nous permet de focaliser la segmentation sur les objets du premier plan. J'explique dans cette section comment est construit le masque de segmentation.

Introduction

L'objectif est maintenant de générer à l'aide d'un filtrage morphologique des blocs du premier plan, des composantes connexes contenant un maximum de pixels appartenant aux objets en mouvement tout en limitant les régions parasites provenant du fond. Ces composantes connexes forment les régions du premier plan et permettent de focaliser correctement la segmentation locale en construisant la racine du masque de segmentation illustré à la figure 3.31.c. Le masque que nous cherchons à construire comporte trois zones : (1) la zone noire qui correspond à la zone de segmentation comportant le contour de l'objet à extraire, (2) la zone intérieure grisée considérée comme la racine de l'objet et (3) la zone extérieure en blanc considérée comme la racine du fond. Chaque racine possède sa propre étiquette qui peut être du type *objet* ou *fond*. Le principe de l'extraction consiste alors en la propagation des étiquettes du fond et de celles de l'objet dans la zone de segmentation selon le même principe que la segmentation locale initialisée par un ruban (paragraphe 3.2.2). Après la segmentation, seuls les pixels bénéficiant d'une étiquette *objet* font partie des ROIs extraites. Dans le cas où 2 ROIs portant des étiquettes objet différentes se retrouvent connexes elles sont regroupées et considérées comme une seule et même ROI. Les éventuels pixels non étiquetés ou portants l'étiquette du fond sont écartés du masque des ROIs.

Détermination des régions du premier plan

Afin d'obtenir des régions du premier plan (figure 3.32.d) robustes, il est primordial d'effectuer un filtrage morphologique des blocs du premier plan (figure 3.32.c). En effet, un certains nombre de blocs a été, à tort, détecté comme appartenant au premier plan. Généralement, nous sommes alors en présence de deux types de blocs : (1) des blocs *regroupés* contenus dans l'objet en mouvement, et (2) des blocs parasites plus *parsemés*. L'élément structurant utilisé

(a) Image originale (b) Blocs du premier plan (c) Masque de segmentation (d) ROI

FIG. 3.31: Principe global de l'extraction

pour les traitements morphologiques suivants est une croix dont l'envergure correspond à la largeur L_B d'un bloc utilisé lors de l'estimation de mouvement.

Voici la succession des traitements morphologiques :

1. une ouverture qui a pour effet de supprimer une partie des blocs isolés considérés comme parasites.
2. une fermeture afin de lisser les composantes connexes et de connecter des régions très proches.
3. une étape optionnelle permettant de limiter la taille des régions du premier plan qui génère une racine. Dans les cas où l'estimation d'un plan est très bruité, cette étape peut permettre d'obtenir des résultats plus satisfaisants.

Une fois que les régions du premier plan sont déterminées (figure 3.32.d), il est possible de construire les racines permettant d'initialiser la segmentation locale. De chaque région du premier plan résulte une racine et une zone de segmentation afin d'extraire l'objet d'intérêt qui l'a générée. L'objectif est de focaliser la segmentation sur les zones incertaines des contours et de définir une racine la plus fidèle possible à l'objet.

Construction de la racine

La racine (figure 3.32.f) est un élément important dans le procédé de segmentation locale car elle définit la surface minimum de la ROI après extraction et contient les différentes hétérogénéités qui composent l'objet. Elle doit répondre à deux caractéristiques importantes :

1. Contenir un maximum de pixels appartenant au réel VOP de l'objet d'intérêt afin de rendre compte des divers hétérogénéités de l'objet.
2. Exclure tous pixels appartenant au fond pour éviter les fuites lors de la segmentation.

Afin d'éviter d'inclure des blocs contenant des pixels du fond, le bord d'une épaisseur d'un bloc de la région du premier plan est systématiquement et définitivement enlevé de la racine. En effet, ces blocs contiennent généralement le contour en mouvement de l'objet intégrant des pixels du fond. Cependant, en plus de cette zone, la détection de blocs du premier plan fournit parfois des erreurs en périphérie de la région. C'est pourquoi, nous définissons dans un premier temps la racine comme étant le cœur de la région du premier plan. Le cœur (cf. figure 3.32.e) est obtenu en érodant fortement la région du premier plan. Il est considéré comme robuste dans le sens où les pixels le composant ont une forte probabilité d'appartenir à l'objet.

Cependant, limiter la racine uniquement au cœur de la région du premier plan aurait pour effet de ne pas rendre suffisamment compte des diverses hétérogénéités formant l'objet. Ainsi, il est nécessaire de consolider la racine avec des blocs de la région du premier en périphérie du cœur.

Consolidation de la racine

La périphérie du cœur est susceptible de comporter des pixels du fond et ainsi de provoquer des fuites non désirables lors de la segmentation s'ils sont intégrés directement dans la racine. Nous effectuons donc une sélection précise des blocs de cette zone qui pourront s'ajouter au cœur pour former et consolider la racine. Pour cela, il est utilisé une méthode introduite par D. Chen et J. Yang dans [25] qui permet, par un calcul de différence d'histogrammes couleur, de déterminer les blocs d'une zone, appelés *blocs de confiance*, qui sont les plus dissemblables des blocs d'un voisinage. En l'occurrence, ceci nous permet de trouver les blocs de la périphérie dissemblables des blocs du voisinage de la région du premier plan. La taille des blocs est de 64×64 pixels afin que l'histogramme construit à partir des pixels contenus dans le bloc soit significatif.

Une amélioration à été apportée par rapport à [25] puisqu'au lieu d'utiliser un voisinage rectangulaire, nous utilisons un voisinage de forme quelconque, plus adapté aux formes des régions du premier plan. La dissimilarité Dis (cf. éq. 3.41) d'un bloc candidat b de la périphérie avec le fond est donné par la somme pondérée des distances $D(b_i, b)$ entre l'histogramme couleur du bloc b et celui de chaque bloc b_i du voisinage.

$$Dis(b) = \frac{\sum_{b_i \in N} (\|b_i - b\| D(b_i, b))}{\sum_{b_i \in N} \|b_i - b\|} \tag{3.41}$$

Avec $\|b_i - b\|$ la distance séparant les blocs. $D(b_i, b)$ représente la différence de Battacharrya qui présente le meilleur compromis entre complexité et performances [57].

Afin de sélectionner les blocs de confiance, la distribution des valeurs de $Dis(b)$, b appartenant à la périphérie, est modélisée par une Gaussienne $P(Dis(b)) \sim G(\mu, \sigma)$. Les blocs dont la dissimilarité dépassent le seuil $T = \mu + \sigma$ sont alors considérés comme des blocs de confiance de la région du premier plan et appartiennent dès lors à la racine (cf. figure 3.32.f).

Les racines ainsi construites, se voient attribuer une étiquette *objet* en vue du procédé de segmentation locale.

Détermination de la zone indéfinie

Pour rappel, la zone indéfinie détermine les pixels non étiquetés dans le procédé de segmentation de la pyramide locale (cf. section 3.1.7). Cette zone doit contenir :

1. un minimum de singularités[7] de l'objet
2. le contour de l'objet
3. un minimum de singularités[7] du fond

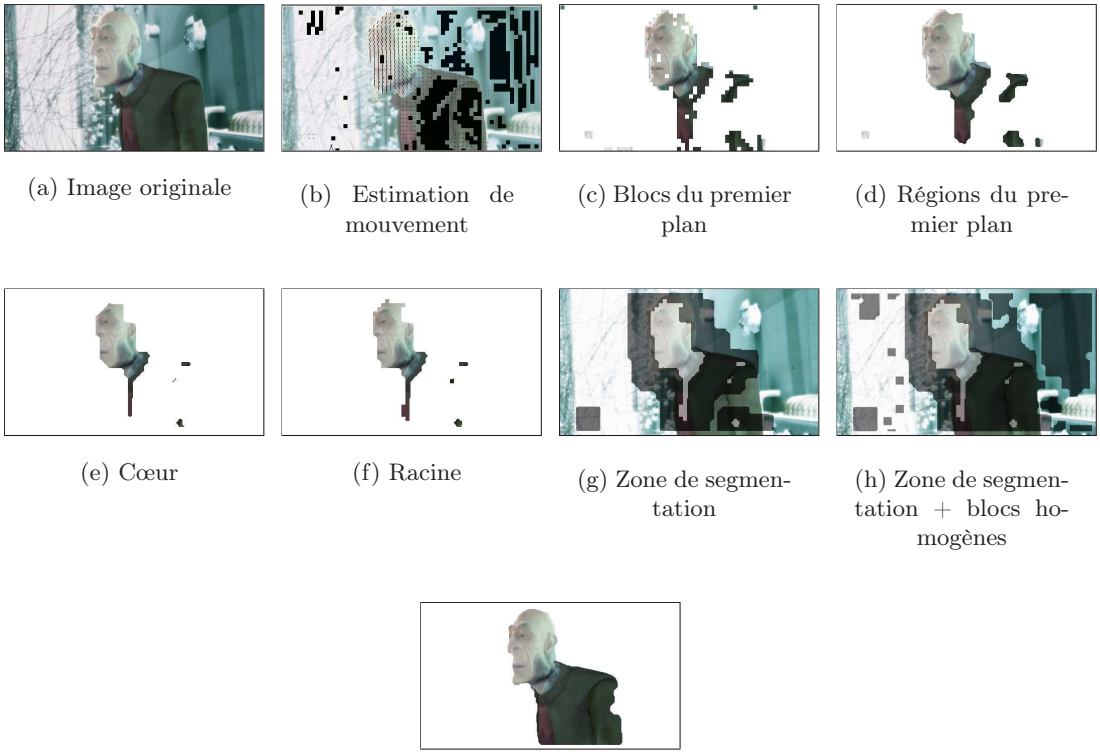
Le premier critère est dual avec la création de la racine elle-même. Par contre il est possible de répondre aux deux autres critères. Pour cela, il est intéressant d'adapter l'épaisseur de la zone de segmentation s'étendant au delà de la racine (figure 3.32.g), aux caractéristiques géométriques de la région du premier plan. Ces caractéristiques sont obtenues en modélisant la région par une ellipse d'inertie [58]. L'épaisseur de la zone de segmentation correspond à une fraction du petit axe de l'ellipse et permet d'adapter la zone de segmentation à la taille de la région du premier plan.

De plus, il nous est apparu intéressant d'inclure systématiquement dans la zone de segmentation, les blocs homogènes écartés du procédé d'estimation de mouvement (figure 3.32.h). En effet, ces blocs ne peuvent pas être classés comme une région du premier plan à cause de leur manque de fiabilité au niveau de l'estimation de mouvement. Pourtant, ils peuvent appartenir à l'objet. Ainsi, le fait de les ajouter dans la zone de segmentation peut permettre de

[7] Toute singularité contenue dans la zone indéfinie ne peut pas fusionner avec l'une ou l'autre des racines et reste non étiquetée.

reconstruire l'objet lors de la segmentation. Le fait d'étendre les zones de segmentation et de connecter éventuellement différentes racines par ce biais, permet une reconstruction de l'objet à partir de plusieurs parties en mouvement. C'est ce que nous voulions mettre en avant avec l'exemple de la figure 3.32. Cependant, une telle reconstruction n'est pas possible à chaque image, mais l'utilisation de la zone de segmentation dite *expansée*, augmente la probabilité d'obtenir des ROIs proches de VOPs sémantiques.

Une fois la segmentation effectuée, on considère qu'une ROI est une composante connexe pouvant regrouper plusieurs régions possédant une étiquette *objet*. Quelques post-traitements ont été imaginés pour améliorer la qualité visuelle et la manipulation des résultats. Tout d'abord, il est possible de limiter la taille minimale des ROIs présentées à l'utilisateur. De plus, les ROIs subissent une légère fermeture afin d'être lissées et plus agréables à visualiser.



(a) Image originale (b) Estimation de mouvement (c) Blocs du premier plan (d) Régions du premier plan

(e) Cœur (f) Racine (g) Zone de segmentation (h) Zone de segmentation + blocs homogènes

(i) ROI

FIG. 3.32: Construction du masque de segmentation et extraction d'une région d'intérêt

3.5 Résultats et discussion

Description

Les résultats de l'extraction automatique sont présentés dans l'annexe D. Le choix des ROIs présentées a été effectué afin de faire ressortir les différents cas observables et non pas de présenter uniquement les meilleures ROIs qui ont pu être extraites au cours des séquences traitées. En effet, l'extraction automatique parfaite d'un objet dans une image est utopique et encore plus pour chaque image d'un plan. Les résultats mettent l'accent sur les variations de la qualité des ROIs au cours d'un plan, qui dépendent du mouvement apparent de l'objet, de son contraste avec le fond, de ses hétérogénéités *etc.*

Il est rare qu'un objet puisse se résumer à une région totalement homogène. Un objet dans une image est souvent composé de nombreuses hétérogénéités aux abords de son contour. Ainsi, il est possible qu'après la segmentation locale, il persiste de petites régions non étiquetées car trop différentes des racines des objets ou du fond. Un étiquetage forcé des ces régions risque de détériorer le contour. En toute rigueur, ces régions ne peuvent pas être rattachées à l'objet ou au fond car elles peuvent tout aussi bien appartenir à l'un ou l'autre. C'est pourquoi, nous proposons deux types de résultats. Le premier présente les ROIs brutes constituées uniquement des pixels ayant obtenu une étiquette *objet* au cours de la segmentation locale. Le deuxième propose de présenter les ROIs brutes augmentées des régions non étiquetées. Afin de marquer leur caractère incertain, il est appliqué un flou Gaussien sur ces régions supplémentaires (ou dans les cas positifs complémentaires). L'insertion de ces régions a uniquement pour but d'améliorer l'aspect visuel et de ce fait la compréhension du masque par un utilisateur.

Les résultats concernent trois séquences (*Foreman*, *Vectra*, *Elephant's Dream*[8]) et se composent de trois parties principales :

1. Les images originales
2. Les ROI brutes
3. Les ROI floues

Avantages

Nous pouvons constater plusieurs avantages dans l'utilisation de notre méthode :

1. Limitation des fuites : les fuites de l'objet vers le fond sont limitées grâce à l'utilisation d'une zone de segmentation restreinte. Le phénomène de fuite intervient lorsque l'objet est trop peu contrasté localement avec le fond et que la différence de contraste se trouve inférieure au seuil défini dans la segmentation. Il peut arriver que la zone de segmentation devienne plus importante lorsque l'image contient de trop nombreux blocs considérés comme homogènes. Dans ce cas, il peut être plus intéressant de supprimer l'intégration de ces blocs dans la zone de segmentation et de segmenter uniquement les zones générées par les régions du premier plan.
2. Précision de la localisation des contours : les frontières des ROIs correspondent souvent avec précision à une majeure partie des contours de l'objet d'intérêt en mouvement.
3. Reconstruction de l'objet : l'utilisation des zones de segmentation expansées permet de reconstruire une grande partie de l'objet à partir uniquement des zones en mouvement.

Inconvénients

Le principal inconvénient de notre méthode d'extraction spatio-temporelle est l'instabilité des ROIs. En considérant un même jeu de paramètres au cours d'une séquence vidéo, la qualité des ROIs concernant un même objet d'intérêt peut varier grandement d'une image à l'autre à cause de la variabilité de son mouvement apparent. Le problème est que l'optimisation du résultat, à chaque image, passe par l'optimisation de chaque étape du traitement qui sont les suivantes :

1. Détection et création des régions du premier plan (automatique)
2. Filtrage optionnel (empirique)
3. Seuillage des blocs homogène (empirique)
4. Création de la racine (automatique)

[8]Court métrage d'animation *open source* réalisé par le studio Orange et distribué sous licence « *Creative Commons* »

5. Création de la zone de segmentation (automatique)
6. Segmentation (définition d'un seuil)

Bien que chaque étape ait une influence, la qualité du résultat dépend fortement de la première. Au cours d'un plan, nous avons pu constater que des jeux de paramètres différents concernant la segmentation (étapes : 4, 5 et 6), permettent d'obtenir approximativement autant d'extractions de qualité d'un objet d'intérêt donné mais ne concernant pas forcément les mêmes occurrences de cet objet.

La succession de ces étapes peut également constituer un avantage dans le sens ou elles sont indépendantes et donc tout à fait remplaçables par des méthodes qui s'avèreraient plus efficaces. Une des perspectives principales est de remplacer la méthode de détection des régions du premier plan par une méthode plus robuste et précise en remplaçant l'estimation de mouvement par bloc par une estimation au niveau du pixel.

Évaluation

La question de l'évaluation des résultats se pose inévitablement. Cependant, il est très difficile d'élaborer un test fiable de ce type d'extraction de part le caractère très subjectif des résultats. Nous évoquerons une nouvelle fois cet aspect dans nos perspectives.

A défaut de fournir une évaluation, le caractère jugé instable des ROIs par rapport aux objets qu'ils représentent nous pousse à définir une notion nouvelle : celle du *sous-objet vidéo* ou S-VOP (*Sub-Video Object*). En effet, les ROIs extraites ne peuvent pas être considérées comme des VOPs sémantiques, tout au moins pas à chaque image. Ainsi, nous définissons le terme de S-VOP qui permet d'insister sur le fait que le mouvement apparent ne permet pas d'extraire à chaque instant l'intégralité de l'objet mais uniquement la partie en mouvement (lorsque la reconstruction de l'objet ne fonctionne pas).

S-VOP : Sous objet vidéo résultant de l'extraction d'une partie d'un objet en mouvement.

3.6 Conclusion

Il apparaît clairement qu'il est difficile de prédire la qualité de l'extraction des S-VOPs. Cette méthode ne peut pas être considérée comme une méthode d'extraction d'objet pour chaque image d'un plan. L'information bas-niveau du mouvement apparent n'est pas significative à chaque image. Cependant, lorsque l'objet se trouve dans les hypothèses d'extraction la qualité des masques est plutôt convaincante. S'il est difficile d'imaginer que l'objet vérifie les hypothèses sur chaque image, en revanche, il n'est pas insensé de penser qu'il peut les vérifier sur plusieurs images du plan et ainsi présenter une meilleure configuration pour son extraction. L'objectif des chapitres suivants est de tirer partie de la qualité des S-VOPs, tout en gérant la variabilité des résultats due à la variation du mouvement apparent.

Chapitre 4

Création de résumés de vidéos

Sommaire

4.1 Introduction

L'évolution rapide des technologies d'imagerie numérique et des télécommunications à ouvert de nombreux axes de recherche s'intéressant à la représentation des vidéos. En effet, les vidéos numériques représentent un volume important de données, mais grâce à la progression des débits des réseaux de télécommunication et à la miniaturisation des moyens de stockage, cette forme de média est de plus en plus utilisée. Il est maintenant possible de filmer ou de visualiser des vidéos dans n'importe quel endroit puisque ces techniques sont de plus en plus communément intégrées dans les systèmes portables tels que les assistants personnels numériques ou les téléphones portables. Les vidéos font partie intégrante de notre quotidien. De plus, les enregistrements vidéo sont très présents dans le milieu professionnel que ce soit pour des applications médicales, de météorologie ou encore de vidéo surveillance, *etc.* Les moyens de stockage grandissant, il est possible de stocker toujours plus de données. L'exemple le plus impressionnant est celui de l'Institut National de l'Audiovisuel (INA) qui est le gardien de pas moins de 60 ans de programmes de télévision ($1,5$ millions d'heures) et propose actuellement près de 10 000 heures de programmes en téléchargement.

Ainsi les utilisateurs se retrouvent face à une masse toujours plus importante de données et qui plus est, difficile à gérer et à manipuler. Il est donc nécessaire de trouver des techniques efficaces pour archiver, cataloguer et indexer ces vidéos toujours plus nombreuses. C'est pourquoi une multitude d'axes de recherches tentent de répondre à ces problèmes.

Afin de mieux cerner les difficultés de la modélisation et de la recherche de documents audio-visuels dans de larges bases de données, nous allons tout d'abord présenter les différentes représentations d'une vidéo. Nous verrons à quel point la structure d'une vidéo peut se révéler complexe. Nous évoquerons ensuite une catégorie de représentation permettant de proposer à un utilisateur, un aperçu synthétique mais fidèle des vidéos : les résumés. Puis nous verrons les différentes méthodes actuelles capables de fournir cette représentation s'appuyant sur une sélection judicieuse des images représentatives du contenu de la vidéo. Nous finirons ce chapitre en évoquant une approche prometteuse de création de résumés de vidéos orientée sur une analyse des objets contenus dans les images.

4.2 Représentation de vidéos

La représentation de données vidéo constitue un problème complexe. En effet, le volume considérable de données et sa richesse rendent difficiles l'extraction et la caractérisation du contenu. De plus, son interprétation par un spectateur tient souvent du domaine du subjectif. C'est pourquoi la majeure partie des outils disponibles actuellement nécessitent une forte interaction avec l'opérateur ou l'utilisateur. Cependant, si l'extraction manuelle d'informations est envisageable au cours de quelques minutes de séquence, elle devient bien plus problématique lorsqu'il faut traiter la vidéo dans son intégralité.

Il est donc nécessaire d'élaborer un procédé, sinon totalement automatique du moins limitant les interactions avec l'opérateur. Ceci constitue une tâche difficile du fait que le contenu d'une vidéo ne répond à aucun schéma standard, sauf pour des cas particuliers d'émissions structurées tels que les journaux télévisés ou les évènements sportifs ... Le problème de l'automatisation induit implicitement 3 niveaux de représentation d'une séquence vidéo [92] :

1. La représentation ***bas-niveau*** vise à décrire les caractéristiques du contenu d'une vidéo à l'aide de critères tels que la couleur, la texture, les formes ou le mouvement... L'objectif des méthodes bas-niveau est généralement de fournir une partition en régions des images de la séquence. Il est alors possible d'extraire de celles-ci des informations permettant de caractériser la vidéo.

2. La représentation ***structurelle*** met en évidence une organisation hiérarchique de la vidéo en images, plans, scènes et séquences (cf. fig. 4.1). Cette structure est directement issue de la production cinématographique. Un *plan* est défini comme une séquence d'images prises en continu par la caméra. Les *scènes* sont définies comme des suites de plans contigus qui sont sémantiquement reliés. Elles situent l'action d'un groupe de personnages dans un même environnement et dans une période de temps continue. Une *séquence* est un ensemble de scènes appartenant au même élan de narration et d'émotion cinématographique.
3. La représentation ***haut-niveau*** vise à fournir une description sémantique du contenu de la vidéo. Le but est ici, de modéliser « l'histoire » véhiculée dans la vidéo. On s'attache à extraire non seulement les objets et les personnages mis en scène mais également leurs interactions.

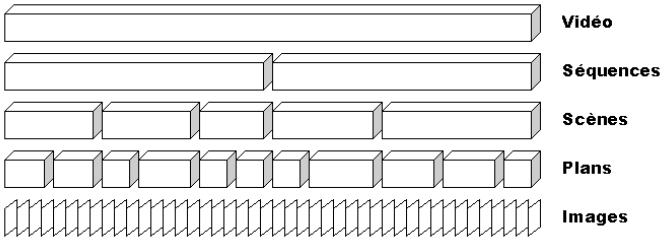


FIG. 4.1: Structure cinématographique d'une vidéo

L'extraction des caractéristiques bas-niveau, de la structure et de la sémantique à des fins de modélisation constitue le travail d'*indexation* de la vidéo [92].

La représentation bas-niveau reste facilement automatisable. Les systèmes d'interrogation de bases d'images par le contenu (CBIR : *Content-based Image Retrieval*) sont généralement fondés sur ce principe. La mesure de similarité entre les caractéristiques bas-niveau extraites dans les images sert à effectuer la recherche. C'est par exemple l'approche des procédés de recherche d'images dans de larges bases de données tels que QBIC développée par P. Flickner *et al.* [38] ou VisualSEEK [109] présentée par J. Smith. Les auteurs de [54] proposent de solliciter l'utilisateur pour construire une région d'intérêt à partir d'une première partition de l'image obtenue sur des critères bas-niveau. La région d'intérêt ainsi choisie, apporte l'aspect sémantique à la requête sur laquelle se base la recherche. Ainsi, la représentation bas-niveau est relativement bien adaptée aux systèmes d'interrogation de bases d'images. Cependant, il est assez problématique de l'étendre à la vidéo. Traiter chaque image serait trop coûteux en temps. De plus, le phénomène de redondance de l'information est très présent dans une vidéo : le contenu de deux images successives peut être excessivement proche. Dans le système *VideoQ* [24], Chang propose de décrire la vidéo et de définir les requêtes de recherche à partir de la trajectoire des objets.

De nombreux travaux ont été effectués sur la représentation structurelle de la vidéo. Les techniques de détection de plan par exemple, semblent assez bien maîtrisées. La difficulté réside dans la diversité des transitions observables entre les plans : fondu-enchaîné[1], volet naturel[2], ou encore le raccord analogie[3] *etc.* La détection de scènes, quant à elle, utilise souvent

[1]Fondu-enchainé : les dernières images d'un plan sont mixées et progressivement remplacées par les premières images du plan suivant

[2]Volet naturel : la transition utilise une obstruction totale de l'objectif lors d'un premier plan (par un personnage par exemple) pour ensuite enchaîner sur un second plan

[3]Raccord analogie : il utilise une similitude en couleur ou de forme entre les deux plans

l'information audio pour arriver à ses fins [69]. Cette décomposition structurelle de la vidéo permet de focaliser l'extraction de caractéristiques et les requêtes sur les différents éléments de la vidéo.

La représentation haut-niveau, quant à elle, comporte encore beaucoup d'obstacles, rendant non envisageable l'absence d'un opérateur. En effet, elle repose sur la notion *d'annotation* qui permet une description symbolique de la vidéo. Ces annotations sont généralement effectuées manuellement à l'aide de logiciels d'indexation. La norme MPEG-7 définit l'utilisation de schémas de description englobant des descripteurs et définissant les relations entre eux. Elle s'intéresse davantage à formaliser le protocole d'annotation pour l'indexation et ceci quel que soit le type de documents audio-visuels. La figure 4.2 présente les différentes étapes d'une application utilisant la norme MPEG-7. Il faut noter que l'extraction de l'information du signal vidéo et les applications ne sont en rien définies dans la norme. C'est pourquoi, de nombreux travaux de recherche tentent de déterminer les informations les plus appropriées pour la construction de ces descripteurs.

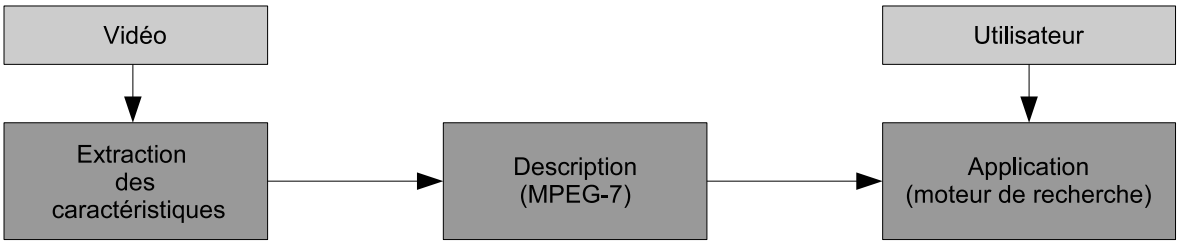


Fig. 4.2: Différentes phases de manipulation d'une vidéo

Une fois la représentation de la vidéo réalisée, il est possible de générer un résumé de cette vidéo. L'objectif est de créer à partir du volume de données considérable contenues dans une vidéo, un condensé d'informations fidèle à la vidéo. La difficulté est bien sûr de conserver l'essentiel de la vidéo tout en la synthétisant un maximum. Nous allons maintenant nous intéresser aux différentes méthodes de création de résumés.

4.3 Résumés de vidéos

A partir des différentes représentations vues dans le paragraphe précédent, il est possible de générer deux types de résumés de vidéo. Ceux générés à base de segments d'images respectant chacun une intégrité sémantique et rassemblant les séquences importantes de la vidéo originale appelés *condensés de vidéo* (ou *video skimming*) puis ceux conçus à base d'images fixes appelés *résumés de vidéo* (ou *video summary*). L'objectif est de présenter à l'utilisateur un contenu rapidement consultable donnant le meilleur aperçu de la vidéo entière.

4.3.1 Condensé de vidéos

Le condensé de vidéos permet d'obtenir un condensé sonore cohérent avec la vidéo originale. Il nécessite donc une synchronisation audio-visuelle. Il existe deux approches pour produire un condensé vidéo. La première consiste à ne considérer que les scènes les plus intéressantes, ce qui se rapporte alors à un problème de détection de scène. Cette forme de condensé ainsi générée est ce qui s'apparente le plus à une bande-annonce. Le concept de la deuxième approche est de donner un aperçu de l'ensemble du contenu du film en un temps plus court et sans distortion. Ceci nécessite une modification de l'échelle temporelle par compression de la vidéo et augmentation de la vitesse du son, tout en conservant un timbre, une qualité et un ton de voix audible et compréhensible.

4.3.2 Résumé de vidéos

Le résumé de vidéos est une collection d'images considérées comme les plus représentatives du contenu des plans des vidéos. Ces images sont appelées *images-clés*. L'extraction d'images-clés peut également prendre en compte la structure cinématographique de la vidéo vue à la figure 4.1. Généralement, un résumé vidéo sélectionne une, voire quelques images-clés par plan. De plus, ces images peuvent ensuite servir de support pour l'indexation de la vidéo. C'est ce type de résumé auquel nous allons davantage nous intéresser par la suite. La paragraphe suivant traite de l'extraction de ces images particulières.

4.4 Extraction d'images-clés

Dans [48] et [72], A. Hanjalic et Y. Li proposent des états de l'art des méthodes d'extraction d'images-clés. La principale difficulté de ce type d'algorithmes réside dans l'évaluation de la pertinence des images choisies. L'extraction d'images-clés s'appuie sur différents critères que nous allons évoquer dans les paragraphes suivants.

4.4.1 Echantillonnage

La sélection d'images-clés par échantillonnage est la méthode la plus triviale. Les images-clés sont extraites parmi les images de la séquence originale par sélection aléatoire ou uniforme selon certains intervalles de temps. L'inconvénient d'une telle méthode est le fait que des plans courts en temps peuvent ne pas avoir d'image représentative ou très peu, tandis que des plans longs contiendront plusieurs images-clés avec, éventuellement, des contenus similaires. Ceci conduit donc à une mauvaise représentation du contenu de la vidéo originale.

4.4.2 Découpage en plans

Principe général

De nombreuses techniques d'extraction d'images-clés utilisent l'analyse structurelle de la vidéo et notamment le découpage en plans. Il est possible de définir deux genres de changements de plans :

- Les changements progressifs qui consistent en l'obtention d'une continuité visuelle lors du passage d'un plan à l'autre. Cette transition est réalisée soit par fondu enchaîné, soit par changement progressif de la couleur de la séquence jusqu'à atteindre une teinte uniforme (*fade in/fade out*).

- Les changements de plans brusques (ou *hard cut*) qui consistent à juxtaposer la fin d'un plan avec le début du plan suivant sans transition. Ces changements sont les plus utilisés dans les vidéos (avec un taux d'environ 95%).

Différentes techniques ont été développées pour le découpage en plans : (1) Les méthodes orientées pixels telles que la comparaison de deux images successives pixel à pixel. (2) Les méthodes orientées histogrammes telles que la différence d'histogrammes. (3) Les méthodes orientées mouvement telles que l'étude du mouvement global, la différence de mouvement ou l'énergie du mouvement calculée sur la totalité de l'image. (4), Enfin les méthodes de fusion de données avec des approches cette fois statistiques. Le lecteur pourra se reporter aux différents états de l'art disponibles dans la littérature [73, 68, 47] ainsi qu'aux travaux de Nagasaka et Tanaka [93] considérés comme initiateurs du domaine. Toutefois, nous tenons à faire référence à des travaux d'un domaine de recherche grandissant se trouvant à la frontière

des sciences cognitives et du traitement du signal : dans [14], G. Boccignone *et al.* présentent une technique de découpage en plans fondée sur la focalisation de l'attention issue du système visuel humain. Le terme d'*attention* est utilisé pour désigner les traitements cognitifs qui filtrent les informations pour ne présenter à la conscience uniquement ce qui est significatif pour l'observateur [19]. Les régions de focalisation de l'attention (FOA : *Focus Of Attention*) sont extraites à l'aide de cartes de saillance. Elles sont ensuite classées par importance spatiale, persistance temporelle et intérêt visuel évalué sur la couleur, la texture et la forme. L'étude de ces régions permet un découpage en plans orienté sur l'intérêt visuel que peut porter un observateur sur la scène.

Une fois le découpage en plans effectué, il est possible d'entreprendre la sélection des images-clés. Les premières méthodes consistaient à choisir la première image de chaque plan. Cependant, si cela convient relativement bien aux plans statiques, ceci ne fournit pas une représentation acceptable du contenu visuel dans les plans dynamiques. De plus, la probabilité que ces images appartiennent à un effet de transition est assez forte, ce qui diminue grandement leur qualité de représentation. Il est donc nécessaire d'adapter le nombre d'images-clés à chaque plan. Pour interpréter le contenu des critères visuels bas-niveau, tels que la couleur, la texture, ou les formes, plusieurs techniques plus élaborées ont été mises au point se fondant toujours dans un premier temps, sur le découpage en plans de la vidéo. Ces méthodes font l'objet des paragraphes suivants.

Histogrammes couleur

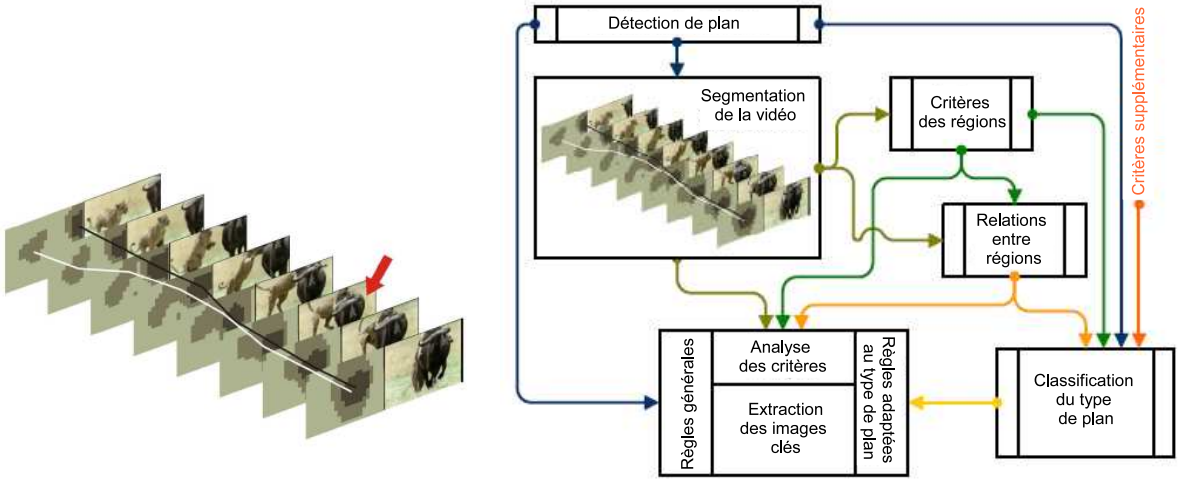
L'histogramme de couleurs est invariant aux orientations de l'image et robuste au bruit. C'est pourquoi, les algorithmes d'extraction d'image clés fondés sur la couleur ont été fortement exploités.

Les travaux de Y. Zhang [131] proposent d'utiliser la couleur pour détecter des images-clés. Une fois que le découpage en plans a été réalisé, la première image est systématiquement déclarée comme une image-clé et devient l'image de référence. L'histogramme de cette image de référence est ensuite comparé aux histogrammes des images suivantes. Le parcours du plan est effectué chronologiquement. Lorsque la distance entre l'histogramme de la référence et celui d'une image donnée excède un certain seuil, l'image courante est déclarée comme la nouvelle référence et devient par la même occasion une image-clé. Dans [132], Huang *et al.* proposent une méthode de classification (ou *clustering*) fondée sur la similarité des histogrammes couleur des images appartenant à un même plan. Les images-clés correspondent aux images les plus proches des centroïdes des différentes classes (ou *clusters*).

L'inconvénient de la plupart de ces travaux réside dans le fait qu'ils sont extrêmement dépendants d'un seuil puisqu'ils découlent de la comparaison d'histogrammes.

Régions clés

Dans [21], J. Calic *et al.* proposent une étude temporelle du comportement de *régions clés* obtenues par une segmentation spatiale à basse résolution. Cette segmentation est réalisée par classification des coefficients de la DCT directement issus d'un signal vidéo compressé. La figure 4.3.a présente une segmentation en régions d'une suite d'images où il est possible de visualiser les trajectoires des deux régions d'intérêt. Les images-clés sont sélectionnées suivant des règles fondées sur les disparitions, les apparitions et les interactions entre les régions. Certaines règles sont générales, tandis que d'autres sont inhérentes au type de plan détecté. La figure 4.3b présente les différentes étapes de la méthode.



(a) Images originales et leur segmentation avec visualisation des trajectoires des régions

(b) Synopsis de de la méthode [21]

FIG. 4.3: Extraction d'images-clés par suivi de régions-clés (d'après [21])

Mouvement

Les approches orientées mouvement ont l'avantage d'adapter le nombre d'images-clés à la dynamique temporelle de la scène. Les méthodes les plus utilisées sont les méthodes de différence d'images [66] ou de flux optique [126].

Dans [126], W. Wolf propose de calculer pour chaque image, une mesure simple du mouvement à partir du flux optique. Cette mesure M correspond à la somme des amplitudes du flux optique des N pixels p_i de l'image traitée ($n \in [\![1,\ N]\!]$). M est donnée par l'équation suivante :

$$M = \sum_{i=1}^{N} |o_x(p_i)| + |o_y(p_i)| \tag{4.1}$$

où $o_x(p_i)$ et $o_y(p_i)$ sont respectivement l'abscisse et l'ordonnée du flux optique calculés en le $i^{\text{ème}}$ pixel de l'image courante.

Par analyse des variations temporelles de cette mesure au cours du plan, les images correspondant aux minima locaux de mouvement sont sélectionnées pour devenir les images-clés. Selon l'auteur, l'analyse en mouvement permet de découper un plan en plusieurs évènements significatifs. Le mouvement est considéré comme un outil que le réalisateur peut utiliser pour mettre en valeur certaines parties du plan. En l'occurrence des mouvements lents sont considérés ici, comme synonymes d'un événement important.

Dans le cas d'applications plus spécialisées, Ju *et al.* réalisent la création d'un résumé dans le contexte d'une présentation d'un exposé [61]. L'extraction des images-clés est fondée sur une reconnaissance du mouvement de la caméra et des gestes des personnes.

Entropie

Dans [87], Markos Mentzelopoulos *et al.* proposent une méthode d'extraction d'images-clés fondée sur la différence d'entropie entre les images. Le principe de l'algorithme est d'isoler dans chaque image les zones saillantes qui présentent une entropie supérieure à 70% de l'entropie

totale de l'image. Théoriquement, ces zones sont supposées être de forte représentativité puisqu'elles contiennent 70% de l'information disponible dans l'image. L'extraction des images-clés est fondée sur l'étude de la différence entre deux images successives de l'entropie contenues dans les zones de forte représentativité. Lorsque la différence excède un certains seuil, une nouvelle image-clé est déclarée. Selon les auteurs, l'algorithme gère bien les images où les objets et les personnages se détachent correctement du fond. Cependant, les performances sont fortement dégradées lorsque la séquence contient des flashes, comme des explosions par exemple.

4.4.3 Découpage en segments

Un des inconvénients majeurs dans l'utilisation d'une ou plusieurs images-clés pour chaque plan est l'incompatibilité avec les longues vidéos comportant un nombre important de plans. C'est pourquoi de récentes recherches se focalisent sur l'aspect haut-niveau des vidéos et tentent d'extraire cette fois des segments de la vidéo qui peuvent être soit une scène soit un événement particulier voire la séquence entière. L'ensemble des images-clés sélectionné par ce type de méthodes est plus concis que celui issu de plans. Le système *video manga* [119] proposé par Uchihashi *et al.*, crée un résumé organisé comme une bande dessinée (fig. 4.4). Leur approche consiste à effectuer une classification de l'ensemble des images de la vidéo en utilisant la similarité de leur histogrammes couleur dans l'espace YUV (cf. annexe A). De cette classification résulte une segmentation de la vidéo indépendante d'un découpage en plans classique. En effet, chaque segment est déterminé par les images contiguës appartenant à la même classe. Ensuite une mesure de l'importance de chaque segment est évaluée par rapport à sa taille et sa rareté. Un segment est d'autant moins important qu'il est court et similaire aux autres segments. Cette mesure permet de supprimer des segments non significatifs et de hiérarchiser ceux restants. Les images-clés présentes dans le résumé sont les images situées au centre de chaque segment retenu. Leur taille est proportionnelle à l'importance des segments dont elles sont extraites.

Fig. 4.4: Exemple d'un résumé de style Bande Dessinée [119]

4.4.4 Autres

Les paragraphes précédents ne représentent qu'un aperçu des méthodes principales présentes dans la littérature ; ce domaine de recherche étant tellement actif. Ainsi, il existe de nombreuses méthodes d'extraction d'images-clés utilisant des outils différents et plus spécialisés telles que la détection de visage ou la transformée en ondelettes. Cependant, il existe une certaine catégorie d'algorithmes qui concerne davantage nos travaux. C'est la construction de résumés fondés sur les objets. Nous allons présenter dans le paragraphe suivant cette nouvelle approche séduisante.

4.5 Approche fondée sur les objets

Nous allons aborder les méthodes de création de résumé fondées sur les objets. Avant toute chose, remarquons que la notion d'objet apporte forcément une notion de sémantique. Cependant la plupart des méthodes d'extraction d'objets présentées par la suite utilisent des critères bas-niveau, donc non-sémantiques (couleur, texture, mouvement). En toute rigueur, il faudrait parler de régions d'intérêt, puisque la notion de sémantique intervient uniquement lors de la visualisation du résumé par l'utilisateur. Dans un souci de simplification, on se permettra par la suite de faire l'analogie entre une région d'intérêt et un objet.

Il existe en fait deux principales familles d'algorithmes fondés sur les objets. La première concerne les méthodes dont l'objectif est de fournir une décomposition fond/objet sur l'ensemble des images du plan. C'est l'objectif de la création de mosaïques. La deuxième famille regroupe les méthodes qui utilisent la notion d'objet pour orienter l'extraction d'images-clés et/ou pour construire une collection d'objets représentative du plan.

4.5.1 Création de mosaïque

Une limitation des techniques fondées sur l'étude du mouvement vues au paragraphe 4.4 est qu'il n'est pas toujours possible d'extraire les images représentant le contenu entier de la vidéo [72]. Par exemple, lors d'une séquence panoramique, même si plusieurs images-clés sont sélectionnées, les dynamiques sous-jacentes ne seront pas correctement capturées.

Dans ce cas, une approche fondée sur la création d'une mosaïque peut être utilisée pour générer une image panoramique qui représente, de manière indirecte, le contenu entier de la vidéo. De plus, elle permet d'extraire les objets tout au long du plan panoramique.

Le procédé s'effectue généralement en deux étapes [121] : (1) calcul du modèle du mouvement global entre deux images successives et (2) composition des images en une seule image panoramique selon les paramètres estimés de la caméra. Une fois que la mosaïque est construite, il est possible d'extraire les objets du premier plan. Dans [52], Javier Ruiz Hidalgo *et al.* proposent une extraction progressive des régions du premier plan, en prenant en compte des informations contour. L'objet (ou région clé) est ensuite caractérisé par sa forme, sa texture, et sa trajectoire

Bien que les mosaïques apportent plus d'informations que les images-clés, elles ont leurs propres limitations. En effet, elles peuvent être calculées uniquement dans les cas particuliers de mouvements panoramiques de la caméra. Cependant les vidéos réelles comportent des mouvements très complexes et changent fréquemment de fond et de premier plan. Une solution à ce problème a été proposée par Taniguchi *et al.* [116] qui consiste en l'utilisation soit d'images-clés soit d'une mosaïque suivant si un mouvement panoramique a été détecté ou non.

4.5.2 Extraction d'objet-clé

Une nouvelle notion est apparue récemment dans la littérature. Cette notion est celle de l'*objet-clé* qui vient compléter celle de l'image-clé. Le résumé est construit à partir d'une collection d'objets ou des images contenant ces objets. En effet, la détermination d'objets-clés peut éventuellement déboucher sur la sélection d'images-clés. Ces deux notions sont fortement liées.

La méthode décrite dans [96] tente de représenter un plan par un ensemble d'objets-clés dont l'extraction est ciblée sur les images supposées favoriser cette extraction. Le principe est d'utiliser un suivi de fond par comparaison de signatures, détaillé dans [95]. Ce suivi permet de déterminer des couples d'images consécutives dont le fond varie très peu. La différence pixel à pixel des deux images de chaque couple permet d'obtenir une carte de contours pertinente.

Cette carte est ensuite utilisée pour extraire les objets en utilisant une méthode fondée sur une estimation par blocs.

Dans [110, 76], Guoliang Fan *et al.* combinent une extraction d'images-clés et une segmentation d'objets vidéo. Une première extraction d'images-clés orientée couleur permet de définir un ensemble d'apprentissage restreint pour la segmentation d'objet fondée sur un modèle statistique. Ensuite, cet ensemble d'images est raffiné afin de maximiser la divergence entre les classes obtenues dans le modèle d'objets. Ceci a pour but d'améliorer la segmentation d'objets tout en sélectionnant les images-clés représentatives de ces derniers.

Dans [63], Kim et Hwang utilisent la segmentation en objets fondée sur une carte de contours en mouvement (*Moving Edge*) pour sélectionner les images-clés parmi l'ensemble des images d'un plan. La première image de chaque plan est automatiquement classée comme image-clé. Ensuite, la segmentation en objets est appliquée sur chaque image du plan. Ceci permet de contrôler le nombre d'objets contenus dans l'image courante par rapport à la dernière image-clé extraite. Une nouvelle image-clé est déclarée lorsque :

1. Le nombre d'objets varie
2. Les régions constituant les objets sont déclarées trop différentes (à nombre d'objets constant).

Cette méthode fonctionne correctement dans le cas où la vidéo ne contient que très peu d'objets telle qu'une vidéo de surveillance mais dans le cas de vidéos plus complexes, les performances diminuent rapidement.

4.5.3 Notre approche

Aux vues de ces méthodes fondées sur les objets, il apparaît intéressant d'utiliser l'extraction automatique de régions d'intérêt en mouvement présentée au chapitre 3 pour la construction de résumé vidéo. Nous avons montré qu'il est difficile de prévoir la qualité de l'extraction de cette méthode dans une vidéo naturelle : un mouvement apparent net de l'objet est synonyme d'une extraction de qualité et la région clé peut alors être apparentée à un VOP. Cependant, rien ne garantit un tel mouvement entre chaque image d'un plan. L'idée est donc de trouver le moyen de regrouper entre-elles les régions d'intérêt extraites au cours du plan concernant le même objet, puis de sélectionner habillement celle(s) qui le représente(nt) le plus fidèlement possible. Le principe de la méthode repose donc sur une première étape de classification de ces régions d'intérêt puis sur l'évaluation de la qualité de l'extraction de chacune d'elles. L'objectif final est bien-sûr d'orienter la sélection d'images-clé sur les images dont on sait qu'elles contiennent un objet d'intérêt tout en fournissant également un masque précis de ce dernier : l'objet-clé. Cette méthode fait l'objet du chapitre suivant.

Chapitre 5

Extraction d'objets-clés dans les vidéos

La perception d'un objet comme désirable ou indésirable ne réside pas dans l'objet lui-même, mais dans la façon dont on le perçoit.
Jean-François Revel, *Le Moine et le Philosophe.*

Sommaire

5.1 Introduction

La fin du chapitre 3 présente une méthode automatique d'extraction spatio-temporelle de régions hétérogènes animées d'un mouvement apparent différent de celui de la caméra. Nous allons utiliser cette méthode afin d'extraire des régions issues d'objets d'intérêt. Du point de vue de l'utilisateur-spectateur, nous appelons *objet d'intérêt* une entité dans un plan qui offre un intérêt sémantique particulier (personnage, visage, véhicule, objet manufacturé, ...). Cette notion est en partie subjective : dans la figure 5.1.a, l'objet d'intérêt est vraisemblablement la personne du premier plan.

Par la suite, on appelle S-VOP (Sub Video Object Plan) chaque instance de composante connexe en mouvement extraite au temps t par la méthode présentée au paragraphe 3.4, chapitre 3. Le suffixe "S" de "S-VOP" indique que la plupart du temps, seule une sous-partie de l'objet d'intérêt est détectée. Sur l'ensemble des plans vidéos que nous avons testés, les S-VOPS flous (cf. chapitre 3) présentent subjectivement une meilleure qualité que les S-VOPs bruts. Dans ce chapitre le terme de S-VOPs désigne donc les S-VOPs flous vus précédemment.

(a) Image originale

(b) Extraction « sémantique »

(c) Extraction sémantique précise : S-VOP pertinent

Fig. 5.1: Qualité d'un S-VOP

La qualité d'un S-VOP illustrée dans la figure 5.1, prend en compte deux aspects liés mais intéressants à distinguer, qui sont :

La pertinence sémantique : le S-VOP doit permettre à l'utilisateur de « reconnaître » l'objet d'intérêt (cf. figure 5.1.b).

La précision de l'extraction : le masque du S-VOP doit :

1. Recouvrir au maximum les pixels de l'objet.
2. Limiter les pixels appartenant au fond
3. Adhérer au maximum aux contours extérieurs de l'objet d'intérêt.

La figure 5.1.c[1] présente un S-VOP sémantique précis. La qualité sémantique des S-VOPs est liée directement au modèle fondé sur le mouvement des régions du premier plan. La qualité de l'extraction est, quant à elle, induite par la segmentation locale et son initialisation (cf. paragraphe 3.4, chapitre 3). Un S-VOP est dit *pertinent* ou *représentatif de l'objet d'intérêt* s'il réunit ces 2 qualités.

A partir d'une vidéo brute préalablement découpée en plans, nous proposons d'extraire de façon générique et automatique n occurrences (S-VOP) pour chaque objet d'intérêt ayant un mouvement apparent différent de celui de la caméra qui filme la scène. L'occurrence la plus représentative est appelée *objet-clé*. Les $n-1$ autres sont les vues-clés et doivent être représentatives de façon complémentaire à l'objet-clé (typiquement l'objet d'intérêt vu sous un autre aspect). Ces différentes vues pourraient permettre par exemple de constituer un

[1] S-VOP obtenu automatiquement par notre méthode d'extraction

modèle 2D multi-vues de l'objet d'intérêt pour prendre en compte sa variabilité et faciliter la recherche par le contenu comme le proposent les auteurs de [136].

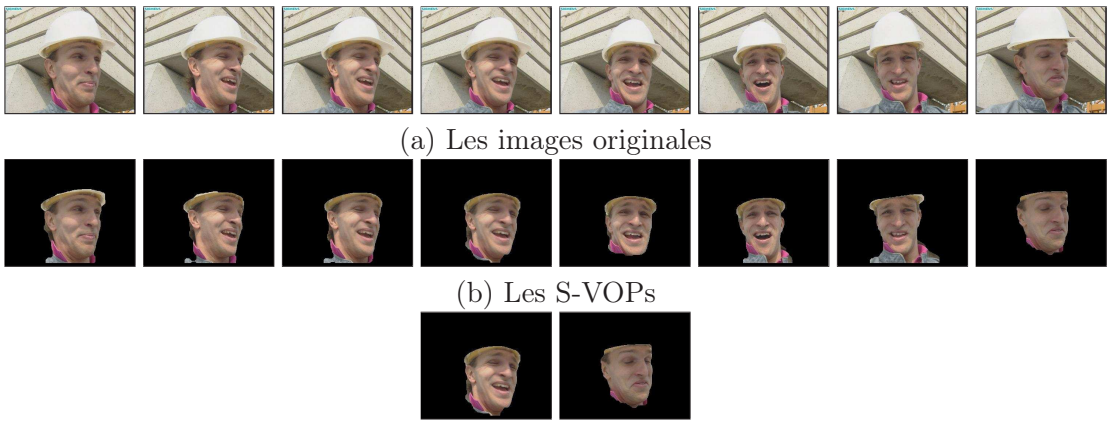
Réaliser ce traitement de représentation des objets d'intérêt de façon automatisée est complexe. Pour cela, nous proposons la chaîne générique suivante effectuée en chaque plan :

---

1. Extraction des S-VOPs en mouvement (cf. paragraphe 3.4, chapitre 3)
2. Rejet des S-VOPs (non pertinents) pas assez compacts ou dont le masque est de mauvaise qualité (algorithme 2, p.116)
3. Classification couleur des S-VOPs, une classe par S-VOP (S-VOP générateur) cf. algorithme 3, p.123
4. Suppression dans chaque classe des S-VOPs non cohérents spatio-temporellement avec le S-VOP générateur (algorithme 4, p.126)
5. Fusion des classes pour obtenir une classe par objet d'intérêt
6. Rejet des classes temporellement peu fiables (algorithme 5, p.132)
7. Sélection d'un objet-clé et des vues-clés associées, pour chaque classe

---

Chaque étape du traitement peut être vue comme une boîte noire pourvue d'un nombre limité d'entrées/sorties. De cette façon, il est envisageable que l'une de ces boîtes soit remplacée si besoin par une autre plus efficace ou dédiée à un type d'application particulier.



(a) Les images originales

(b) Les S-VOPs

(c) L'objet-clé et une vue-clé

Fig. 5.2: Illustration de la notion d'objet-clé

L'objectif de la méthode est de synthétiser un maximum le contenu sémantique de chaque plan en sélectionnant un objet-clé et éventuellement des vues-clés complémentaires décrivant chaque objet d'intérêt. Cette approche est illustrée dans la figure 5.2 par quelques images issues du plan-séquence *Foreman*. Les images associées aux objets-clés peuvent également servir en tant qu'*images-clés*. Le principe est de proposer à l'utilisateur un résumé de la vidéo compact et pertinent tout en proposant un masque le plus fidèle possible à l'apparence (VOP) de l'objet d'intérêt. Le paragraphe 5.6 propose de discuter sur la présentation et la mise en forme des résultats suivant l'application visée.

5.2 Rejets des S-VOPs non pertinents

概念

Parmi tous les objets extraits dans la première étape, certains ne sont pas pertinents (de mauvaise qualité en quelques sorte) et ne doivent pas être pris en compte par la suite. Dans cette section, j'explique comment est réalisé ce filtrage.

A partir de l'ensemble des S-VOPs extraits d'un plan, qui peuvent être de qualité et de pertinence diverses, on veut déterminer un représentant de chaque objet d'intérêt. Ce représentant est appelé *objet-clé*. Afin de sélectionner les objets-clés, il est nécessaire de regrouper les S-VOPs en classes appelées *classes-clés*. Chaque classe clé correspond idéalement à un objet d'intérêt. La classification a un double objectif :

1. La détermination du nombre d'objets d'intérêt qui n'est pas une information connue *a priori*.
2. Le regroupement dans chaque classe des S-VOPs pertinents et représentatifs d'un objet d'intérêt.

Selon le second point, une classe doit regrouper les S-VOPs les plus représentatifs de l'objet d'intérêt. Ainsi, parmi les S-VOPs extraits, seul un sous-ensemble sera sélectionné et conservé pour la suite du traitement.

Pour rendre compte de la validité d'un S-VOP, deux critères sont utilisés : le premier concerne la géométrie des S-VOPs et permet de rejeter des S-VOPs parasites issus d'une mauvaise extraction. Le deuxième exprime la qualité du masque binaire du S-VOP en évaluant l'adéquation des contours du masque binaire avec les contours réels dans l'image.

5.2.1 Compacité

Une méthode d'extraction générique non paramétrée pour chaque vidéo traitée induit automatiquement des erreurs ponctuelles dans la détection et l'extraction des S-VOPs. Un des principaux défauts gênant dans l'extraction provenant de l'estimation de mouvement est, ce qui peut être appelé : les *fuites* de l'objet vers le fond (cf. fig. 5.3.b). Une caractéristique intéressante de ce type de régions est une très faible compacité qui traduit des régions fines et très allongées. Ce critère est d'ailleurs souvent utilisé pour évaluer la pertinence visuelle d'une région au sens de la perception visuelle humaine [65]. C'est donc cette particularité qui est utilisée dans l'algorithme afin de discriminer les S-VOPs parasites des S-VOPs potentiellement pertinents.

(a) S-VOP extrait à l'image 55

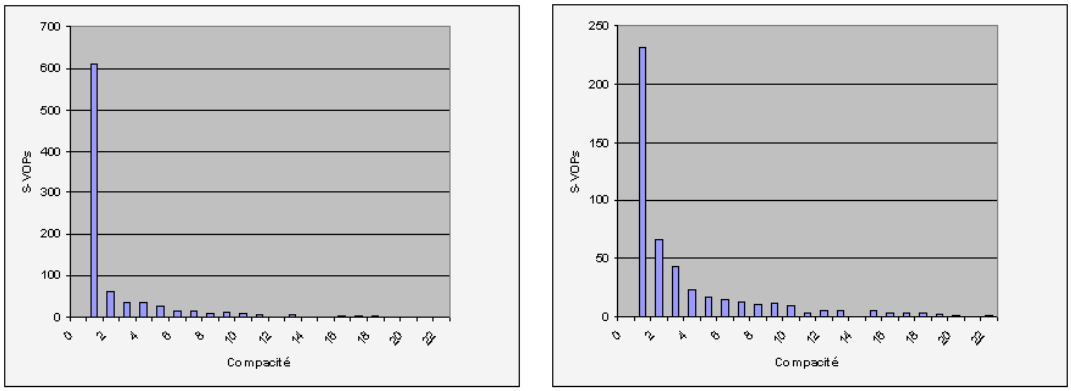
(b) S-VOP extrait à l'image 56

Fig. 5.3: La compacité : un critère discriminant

La compacité C_1 d'un S-VOP s est traduite par le *facteur de forme* dont l'expression est la suivante :

$$C_1(s) = \frac{\text{Périmètre}(s)^2}{4\Pi \times Aire(s)} \tag{5.1}$$

$C_1 \in [1,\ \infty]$. La figure 5.4 présente la distribution du facteur de forme des S-VOPs dans deux plans vidéo. On observe un mode prononcé pour un facteur de forme proche de 1 qui correspond a des régions compactes. Ce phénomène a été observé sur quelques dizaines de plans étudiés de vidéos diverses. Afin de ne pas être trop discriminant, un seuil empirique peu restrictif $S_1 = 2.5$ est appliqué afin de filtrer les S-VOPs dont le facteur de forme est trop élevé. Ainsi tout S-VOP dépassant cette valeur est considéré comme une région parasite et n'est pas pris en compte dans la suite du traitement.



(a) Plan-séquence *Foreman* (b) Plan-séquence *Vectra*

FIG. 5.4: Distribution du facteur de forme des S-VOPs au cours d'un plan vidéo

5.2.2 Évaluation de la qualité de l'extraction

L'objectif de cette étape est d'évaluer la pertinence du S-VOP selon l'adéquation du S-VOP compact avec un VOP théorique. Nous mesurons donc ici, la qualité d'un masque sous la forme d'une correspondance notée $C_2(s)$ entre la périphérie z du S-VOP s et les contours c dans l'image originale. z est obtenue par une dilatation morphologique du contour du masque binaire du S-VOP (cf. fig. 5.5) selon l'équation 5.2[2] :

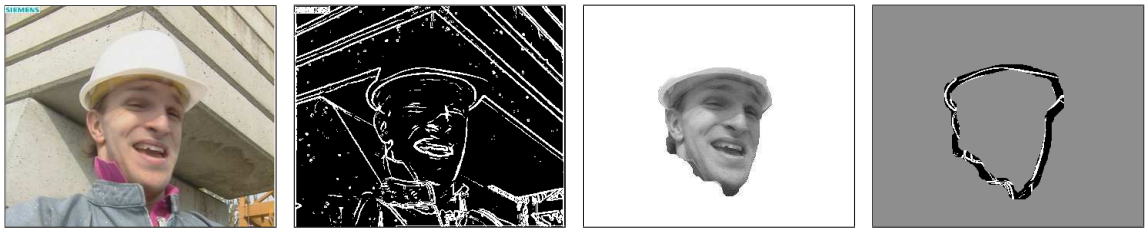
$$z(s) = Dilat_\epsilon(s) \setminus Erod_\epsilon(s) \tag{5.2}$$

$$C2(s) = \frac{card(c \in z)}{Aire(z)} \tag{5.3}$$

Les points de contour sont déterminés par un seuillage adaptatif des valeurs du gradient de l'image d'où est extrait le S-VOP. Cette image de la norme des gradients est obtenue par une méthode de Sobel. Afin de s'adapter au contours de l'image originale, l'image des gradients est modélisée par des points de contours auxquels s'ajoute du bruit blanc Gaussien (pixels des zones homogènes). En observant l'histogramme de l'image des gradients, il apparaît que les

[2] $A \setminus B$ désigne l'ensemble de tous les éléments de A qui n'appartiennent pas à B. Ici, c'est la "différence" entre le dilaté et l'érodé de s par l'élément structurant ϵ

pixels des zones homogènes sont représentés par le premier mode correspondant aux valeurs de gradients faibles. Pour estimer au mieux la distribution du bruit qui est théoriquement de moyenne nulle, il est intéressant d'écarter du procédé d'estimation les pixels qui ont une forte probabilité d'appartenir à des contours forts. Une hypothèse est faite, selon laquelle seulement un faible pourcentage p de l'image est constitué de points de contour (ici $p = 30\%$) ayant une valeur de gradient élevée. Ceci étant, seules les valeurs de gradient faible restantes $(100 - p\%)$ dans l'histogramme sont modélisées par une Gaussienne $\mathcal{N}(\sigma, \mu)$. Les pixels dont la norme du gradient dépasse 3σ sont alors considérés comme des points de contours.



(a) Image originale (b) Points de contours (c) S-VOP (d) Superposition des contours (blancs) et de la périphérie z du S-VOP (noir)

FIG. 5.5: Obtention du coefficient C_2

Une faible valeur de C_2 indique que peu de points de contours ont été détectés autour de la frontière du S-VOP. Un tel S-VOP est considéré comme étant de mauvaise qualité et a peu de chance d'être pertinent. L'algorithme supprime du procédé les S-VOPs dont la valeur de C_2 est inférieure à un certain seuil S_2 propre à chaque plan. La valeur de C_2 peut varier d'un S-VOP à un autre qu'ils soient issus du même objet d'intérêt ou non. Ceci est dû au fait que C_2 est directement lié au mouvement apparent (variable au cours du temps) de l'objet d'intérêt dans l'image. S_2 est donc calculé sur l'ensemble des coefficients C_2 de la population des S-VOPs du plan. La figure 5.6 présente l'allure de la distribution de la pertinence des S-VOPs au cours de 3 plans vidéo. Il apparaît dans chaque exemple, un mode principal représentant la majeure partie de la population des S-VOPs, que ce soit pour des plans contenant un seul (*Foreman* et *Vectra* fig. 5.6.a et 5.6.b) ou plusieurs objets d'intérêt (*Children* fig. 5.6.c). Afin de déterminer les S-VOPs de faible coefficient de pertinence du plan, l'ensemble des coefficients de la population est modélisé par une Gaussienne de moyenne μ et d'écart-type σ. Un S-VOP est considéré comme non pertinent si le coefficient C_2 est inférieur au seuil S_2 donné par l'expression suivante :

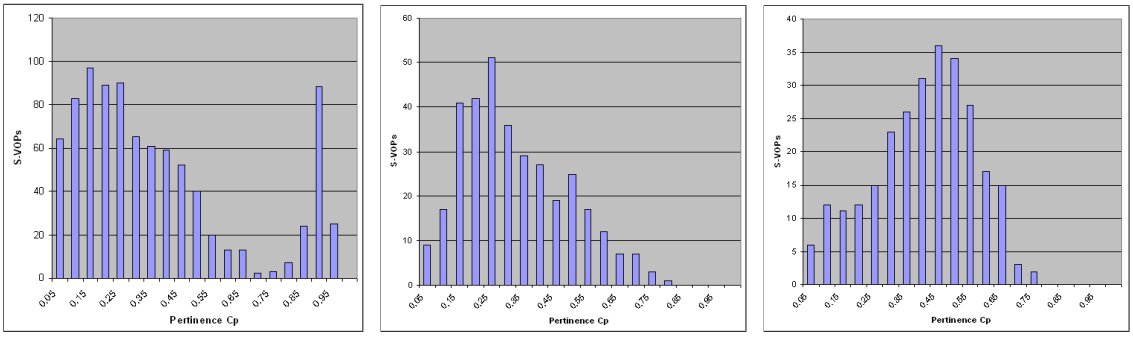
$$S_2 = \mu(C_2) - \sigma(C_2) \tag{5.4}$$

Le seuil S_2 étant peu restrictif, permet de conserver les S-VOPs pertinents sans diminuer de manière trop drastique la population des S-VOPs.

L'algorithme 2 fait la synthèse de l'utilisation des deux critères vus dans cette section.

Algorithme 2 : Critère de pré-selection des S-VOPs

si $C_1(s) > S_1$ *ou* $C_2(s) < S_2$ **alors**
| s est rejeté ;
fin



(a) Plan-séquence *Foreman* (b) Plan-séquence *Vectra* (c) Plan-séquence *Children*

FIG. 5.6: Distribution de la pertinence des S-VOPs au cours d'un plan

Cette étape de pré-sélection permet d'écarter les S-VOPs non pertinents du procédé de recherche des objets-clés. L'objectif est maintenant de trouver une modélisation judicieuse des S-VOPs afin de pouvoir les apparier entre eux par rapport aux objets d'intérêt dont ils sont issus.

5.3 Classification des S-VOPs

概念

A ce stade, de nombreux S-VOPs correspondent au même objet d'intérêt. Cette section a pour but de montrer comment sont construites les n classes représentant les n objets d'intérêt du plan.

5.3.1 Problématique

La modélisation des S-VOPs choisie détermine la méthode de classification utilisée pour apparier les S-VOPs aux objets d'intérêt correspondants. Afin d'éviter les problèmes inhérents aux méthodes de suivi (occultations, disparition, apparition...), la méthode de classification des S-VOPs est réalisée sur un critère d'*état ponctuel* permettant de modéliser les S-VOPs et mettant de côté dans un premier temps, toute information spatio-temporelle.

Le choix de la classification s'est porté vers une méthode *hors ligne*, en opposition à une classification *en ligne* orientée *suivi* : d'une part, à cause de la nature sporadique des S-VOPs instables temporellement[3] et d'autre part, à cause du rapport *complexité/efficacité* des méthodes de suivi dans un contexte de vidéos réelles.

Afin de repousser au maximum les contraintes sans restreindre les domaines d'applications, la classification est donc réalisée par une méthode utilisant tout d'abord un critère d'état prenant en compte l'ensemble des S-VOPs du plan. Ainsi, en théorie un objet d'intérêt peut être « perdu » ou « raté » pendant l'étape d'extraction sans que ceci ne constitue un problème pour l'étape de classification. Le but étant de trouver une représentation de chaque objet d'intérêt du plan, il n'est pas nécessaire de *suivre* l'objet tout au long du plan mais simplement d'être capable de connaître le nombre d'objets d'intérêt et de savoir quand ils sont extraits de manière correcte. La classification a donc pour but de limiter les fausses détections d'une part, en limitant le nombre de classes et d'autre part, en regroupant *uniquement* les S-VOPs les

[3]Seuls les objets animés d'un mouvement différent du mouvement global de la caméra peuvent être détectés. Un objet suivant le mouvement de la caméra entre deux images n'est pas extrait

plus pertinents appartenant au même objet d'intérêt. La méthode de classification recherchée privilégie donc la *précision* dans le rapport *rappel/précision*, souvent évoqué dans l'évaluation de tels algorithmes. Le paragraphe suivant s'intéresse aux descripteurs usuels permettant de modéliser les S-VOPs en vue de leur classification.

5.3.2 Les descripteurs usuels

Le choix du descripteur est très important dans la classification. Les méthodes d'appariement d'objets et d'images issus de l'indexation utilisent principalement trois types de descripteurs fondés sur la forme, la texture ou la couleur. Il est possible de combiner plusieurs descripteurs dans une seule méthode d'appariement. Cependant, plus la dimension de l'espace des descripteurs est élevée plus la constitution des classes est délicate. Il est donc nécessaire de n'utiliser que les critères les plus discriminants par rapport à la population à étudier.

La forme ne peut pas être utilisée ici comme critère de classification et ceci pour deux raisons. Premièrement, la nature même des S-VOPs discrédite le critère de forme. En effet, les S-VOPs représentent uniquement les parties de l'objet qui sont en mouvement. Ainsi, la forme des S-VOPs est instable temporellement ce qui la rend inutilisable. Deuxièmement, durant un plan la forme des objets non rigides peut changer radicalement. Afin que la méthode soit la plus invariante possible aux transformations des objets intra-plan, ce type de descripteur est donc écarté.

L'utilisation de descripteurs de texture, quant à eux, est davantage justifiée dans le cas de traitement sur les images en niveau de gris bien qu'ils constituent une bonne amélioration des descripteurs couleur.

La couleur est la caractéristique la plus utilisée par les systèmes d'indexation et de recherche d'images par le contenu. De nombreux descripteurs fondés sur la couleur ont été étudiés.

Swain *et al.* proposent dans [114] d'utiliser un histogramme afin de décrire la distribution globale des couleurs d'une image. Cependant, il est souvent nécessaire de quantifier les couleurs lorsque l'on est en présence d'images contenant un grand nombre de couleurs. Ceci permet de réduire le nombre de cellules (*bins*) contenues dans l'histogramme. La méthode de quantification la plus simple consiste à fusionner les couleurs contenues dans un espace donné. Une autre méthode consiste à appliquer un procédé de classification de type agrégation autour de centres mobiles afin de déterminer les classes couleur. Le choix du nombre de cellules reste un problème ouvert. Cependant, les expérimentations montrent que des histogrammes de dimensions réduites fournissent des résultats satisfaisants.

Les histogrammes couleur sont considérés comme une représentation de l'apparence des objets extrêmement robuste et sont donc largement utilisés dans les analyses de séquences. Une méthode de suivi d'objet non rigide grandement utilisée, est le *mean-shift*. Elle a été introduite par Comaniciu *et al.* dans [27] puis par la suite améliorée [28, 134]. La forme de l'objet est tout d'abord modélisée par une ellipse. Puis, les pixels de l'objet contenus dans l'ellipse sont utilisés pour générer un histogramme couleur. L'objectif est de déterminer dans l'image suivante, une région ellipsoïdale similaire à celle de l'objet de référence.

Les principaux atouts inhérents à l'utilisation d'histogrammes sont le faible coût en temps de calcul et l'invariance à certaines modification de l'objet ou de l'image telles que les rotations $2D$, les changements d'échelle et la position relative dans l'image. Afin de rendre l'histogramme invariant aux changements d'éclairement, il est utile de choisir l'espace couleur à partir duquel l'histogramme est construit. Le fait d'utiliser un espace couleur normalisé tel que *rgb* permet de s'affranchir du problème de la variation d'éclairement [37, 112]. Il existe également d'autres espaces couleurs permettant de séparer l'intensité de l'information chromatique tels que l'espace HSV qui est également très utilisé dans les systèmes d'indexation ou

encore l'espace $L^*a^*b^*$. Il existe de nombreuses mesures de distances entre histogrammes [23]. L'une des plus utilisées est la distance de Bhattacharrya qui présente le meilleur compromis entre complexité et performances.

Plusieurs améliorations ont été apportées aux histogrammes afin qu'ils puissent incorporer une information spatiale [99, 113, 100]. La plupart des méthodes utilisées sont fondées sur le partitionnement de l'image en régions. Une méthode nommée CCV (Color Coherent Vector) [99] utilise une autre approche (raffinement de l'histogramme) : les cellules de l'histogramme sont partitionnées en fonction de la cohérence spatiale des pixels ; un pixel est cohérent s'il appartient à une région d'assez grande taille et uniformément colorée, sinon il est incohérent. Cette méthode donne de bien meilleurs résultats que l'histogramme cependant il est sensible aux changements d'éclairement et aux occultations partielles du fait qu'elle intègre des informations spatiales.

5.3.3 Choix du critère

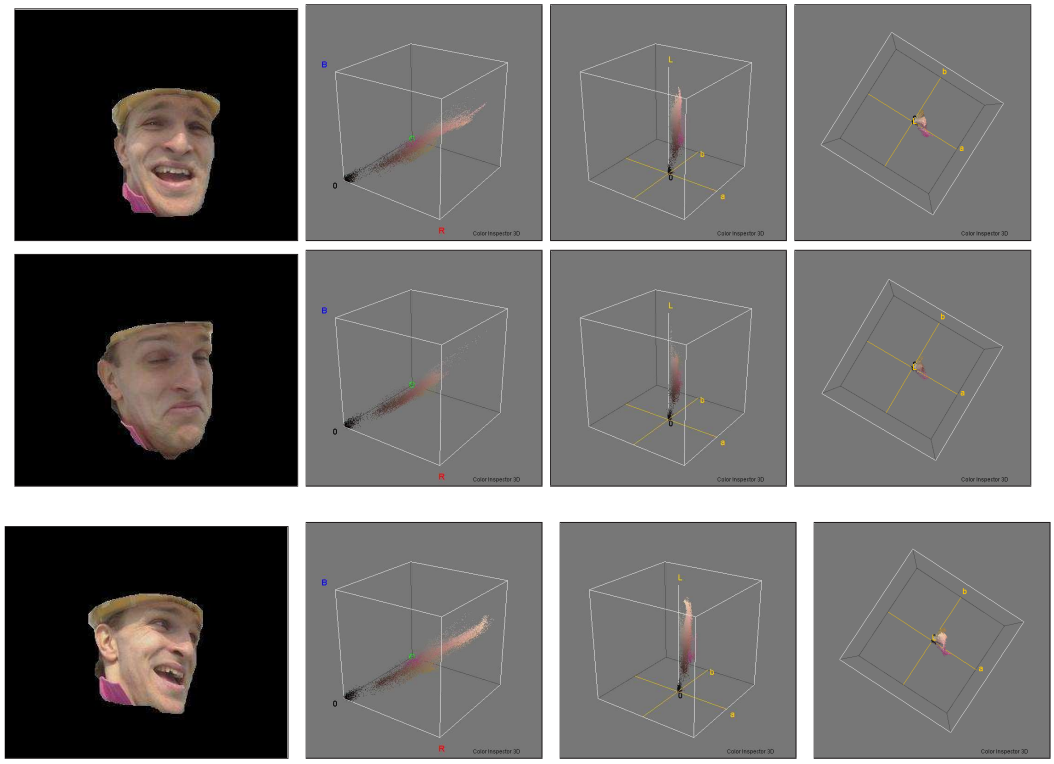
La couleur reste un des critères les plus représentatifs d'un objet au cours d'un plan, en émettant l'hypothèse que pendant la durée du plan qui est classiquement d'une durée courte, les variations de couleur des objets d'intérêt sont relativement faibles. L'étude de R. Hammoud [46] sur les variations intra-plan d'un objet vidéo, montre que ces dernières concernent essentiellement les variations d'éclairement. Ainsi, la couleur constitue un critère relativement robuste quant à la représentation des S-VOPs. C'est pourquoi, le choix du critère s'est porté vers celui de la couleur. La figure 5.7 montre l'effet du changement d'éclairement sur un objet durant un plan. Il apparaît que l'espace RGB n'est pas invariant à ces changements. L'étalement des données dans l'espace le montre. Par contre, il est possible de voir que la projection de l'objet sur les composantes chromatiques de l'espace $L^*a^*b^*$ permet d'être beaucoup moins sujet à ces variations puisque les données ont des distributions relativement similaires pour des éclairements différents. L'espace couleur qui a été retenu pour cette étude est l'espace $L^*a^*b^*$ où seules les composantes chromatiques a^* et b^* sont utilisées afin de s'affranchir des variations de l'éclairement.

5.3.4 Principe de la classification 2 temps utilisée

Dans cette partie, je discute et justifie notre choix de méthode de classification

Plusieurs questions se posent ici. D'une part, le nombre d'objets d'intérêt présents dans le plan n'est pas connu *a priori* (tout comme leur temps d'apparition). Or, pour la plupart des méthodes de classification hors ligne cela reste un paramètre sensible qui constitue un sujet de recherche encore ouvert et très actif. D'autre part, le problème est de savoir comment intégrer l'information spatio-temporelle dans le procédé d'appariement des S-VOPs en fonction des objets d'intérêt dont ils sont issus. En effet, se limiter au simple critère couleur ne permet pas de discriminer deux objets d'intérêt de couleurs similaires distants dans un même plan vidéo. Une classification hors-ligne en *2 temps* a donc été mise au point :

1. Tout d'abord, chaque S-VOP est considéré comme un objet-clé potentiel. Ainsi chaque S-VOP génère sa propre classe sur un critère couleur. Puis chaque classe appelée Classe Couleur ou 2XC, est ensuite filtrée selon un critère de cohérence spatio-temporelle pour donner un ensemble de Classes Couleur Cohérentes ou 3XC.
2. Ensuite les 3XC sont fusionnées pour obtenir idéalement une bijection entre objets d'intérêt et classes : chaque classe correspond à un objet d'intérêt et inversement. Ces classes sont appelées les *classes-clés*.



(a) Images originales de haut en bas : #116, #172 et #234

(b) Distribution dans l'espace *RGB*

(c) Distribution dans l'espace $L^*a^*b^*$

(d) Projection sur les composantes chromatiques a^* et b^*

FIG. 5.7: Variation de la luminosité sur un objet intra plan

Le fait que le nombre d'objets d'intérêt et donc de classes-clés soit inconnu, motive le choix d'une telle méthode. La méthode de classification couleur est orientée *un contre tous* ce qui permet d'obtenir une bonne précision en ce qui concerne la constitution des *2XC*. Cette méthode de classification a été choisie pour sa simplicité de mise en oeuvre et la qualité satisfaisante des résultats obtenus (cf. § 5.8) pour la sélection d'objets-clés.

5.3.5 Classification couleur

概念

Dans cette partie, j'explique comment le critère couleur est utilisé pour caractériser chaque S-VOP et comment on calcule la similarité couleur entre deux S-VOPs, afin de décider s'ils appartiennent à la même classe.

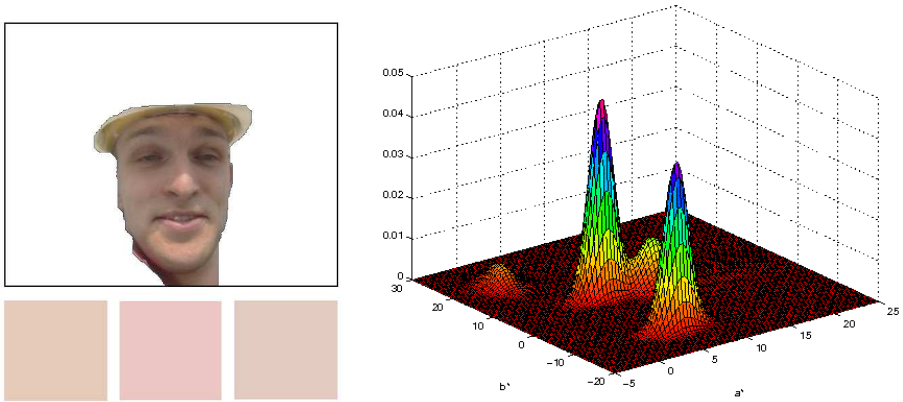
Ici, chaque S-VOP est modélisé par un mélange de Gaussiennes, puis comparé à tous les autres S-VOPs à l'aide de cette modélisation. Les S-VOPs possédant des mélanges similaires sont classés dans la même 2XC. Bien entendu, les classes s'intersectent de façon importante.

Modélisation des S-VOPs

Ce paragraphe présente comment le critère de la couleur est utilisé pour modéliser les S-VOPS. Afin d'être le plus robuste possible et d'avoir un modèle couleur fidèle à l'objet d'intérêt

au cours d'un plan, seuls les pixels des blocs de confiance[4] sont considérés dans la modélisation. Ces pixels sont utilisés pour constituer un histogramme. Ce dernier est ensuite modélisé par un mélange de k Gaussiennes à 2 dimensions (cf. figure 5.8). Les dimensions correspondent aux deux composantes chromatiques de l'espace couleur $L^*a^*b^*$. Chaque Gaussienne représente un *groupe* de pixels. Le risque d'utiliser uniquement les pixels des blocs de confiance est d'avoir un nombre de données insuffisant pour obtenir une modélisation couleur significative. Lorsque cette situation se présente, le S-VOP est exclu du traitement de recherche des objets-clés. Cependant, les objets considérés étant de taille relativement importante, on émet l'hypothèse selon laquelle le nombre de blocs de confiance est suffisant pour fournir dans la plupart des cas, assez de données pour effectuer une classification significative et robuste par rapport au contenu du S-VOP.

Le principe de la classification couleur est d'utiliser les mélanges de Gaussiennes comme critère de correspondance entre S-VOPs. Les S-VOPs possédant des mélanges similaires seront classés dans la même 2XC.



(a) Un S-VOP et ses 3 couleurs dominantes ($L^* = c^{ste} = 85$)

(b) Mélange de 5 Gaussiennes dont seulement 3 ont un poids significatif

FIG. 5.8: Modélisation couleur d'un S-VOP

Modélisation par mélanges de Gaussiennes

Soit X la variable aléatoire représentant la position du pixel dans le plan a^*b^*. Un mélange de Gaussiennes s'exprime par la fonction de densité de probabilité P suivante :

$$P(X) = \sum_{i=1}^{k} w_i \times G(\mu_i, \Sigma_i, X) \tag{5.5}$$

Où w_i est la proportion de données représentée par la $i^{\text{ème}}$ Gaussienne du mélange et est tel que $0 < w_i < 1 \ \forall i \in [\![1, \ldots, k]\!]$ et $\sum_{i=1}^{k} w_i = 1$. μ représente le vecteur des moyennes et Σ la matrice de covariance. $G(\mu_i, \Sigma_i, X)$ est la fonction de densité de probabilité de la $i^{\text{ème}}$ Gaussienne à 2 dimensions donnée par l'expression suivante :

[4]Les blocs de confiance utilisés ici sont les mêmes qui permettent de déterminer la racine dans l'étape d'extraction des S-VOPs. cf. paragraphe 3.4.3 page 95

$$G(\mu_i, \Sigma_i, X) = \frac{1}{(2\pi)|\Sigma|^{1/2}} \; exp\left(-\frac{1}{2}\;(x-\mu)^T \Sigma^{-1}\;(x-\mu)\right) \tag{5.6}$$

Obtention des Gaussiennes

Les Gaussiennes sont obtenues par l'agrégation autour de centres mobiles (*k-means*) qui est une variante des algorithmes EM (*Expectation-Maximisation*) dont l'objectif est de découper les données en k groupes appartenant à des distributions de type Gaussien. Le principe est de minimiser la variance intra groupe ce qui revient à minimiser la fonction suivante :

$$V = \sum_{i=1}^{k} \sum_{j \in G_i} |x_j - \mu_i|^2 \tag{5.7}$$

Où k est le nombre de groupes ; G_i, $i \in [\![1, ..., k]\!]$ et μ_i est le centroïde ou le point moyen de tous les points $x_j \in G_i$.

L'algorithme commence par partitionner les points en k groupes initiaux, choisis de manière soit stochastique soit heuristique (ici, l'initialisation est heuristique). Ensuite, les points moyens, appelés également centroïdes sont calculés pour chaque groupe. Une nouvelle partition est construite en associant chaque point avec le centroïde le plus proche. Ensuite les centroïdes sont re-calculés pour les nouveaux groupes. L'algorithme réitère ces deux étapes jusqu'à convergence qui est obtenue lorsque les points ne peuvent plus changer de groupes.

La convergence de cet algorithme est donc assurée. Elle ne dépend que du nombre d'itérations qui n'est pas connu *a priori*. Cependant, la convergence peut déboucher sur un minimum local et la qualité du résultat dépend grandement de l'initialisation.

Influence du nombre de Gaussiennes par mélange

L'inconvénient majeur de cet algorithme réside dans le fait qu'il est nécessaire de fixer au préalable le nombre de groupes utilisés pour modéliser les données. Le problème du nombre de groupes se pose donc de nouveau. La maximisation du nombre de groupes reste un problème ouvert sur lequel un grand nombre de travaux sont effectués. Cependant, cette caractéristique étant gênante pour des applications de type segmentation, l'est un peu moins pour des applications de classification utilisant les groupes calculés. En effet, le but de la modélisation par mélange de Gaussiennes est de réussir à capturer des éléments représentatifs de l'objet mais pas de représenter dans le détail les objets. Deux objets similaires peuvent être représentés par 1, 2 ou 3 Gaussiennes, la classification utilisant ces groupes sera bien sûr toujours correcte. Par contre, lorsque deux objets sont différents deux cas se présentent :

1. le nombre de Gaussiennes est trop important par rapport au nombre de couleurs dominantes constituant l'objet. Ainsi, des scissions non pertinentes apparaissent entre les groupes. Le mélange dépend alors fortement de l'initialisation de l'algorithme et des Gaussiennes peu représentatives apparaissent alors dans le mélange.
2. le nombre de Gaussiennes est trop faible ce qui tend à moyenner les divergences entre les objets.

Pour parer au problème du cas 1, seules les Gaussiennes de poids forts sont prises en compte pour la classification. Ce qui a pour but d'éviter les différences possibles apportées par des Gaussiennes ne représentant qu'une faible partie des données et dépendant fortement de l'initialisation.

En revanche il est nécessaire de trouver un compromis pour le cas numéro 2 qui constitue le cas le plus délicat pour la classification. Le nombre de Gaussiennes est fixé expérimentalement à 5 afin de pouvoir rendre compte d'un nombre suffisant de particularités au niveau des couleurs des objets.

Comparaison des Gaussiennes

La modélisation des couleurs par mélange de Gaussiennes permet une comparaison pratique et efficace des couleurs deux à deux : pour quantifier le recouvrement (et donc la similarité) de deux Gaussiennes, on utilise le critère de [31] : deux Gaussiennes $\mathcal{N}(\mu_1, \Sigma_1)$ et $\mathcal{N}(\mu_2, \Sigma_2)$ sont *c-séparées* si :

$$\|\mu_1 - \mu_2\| \geqslant c\sqrt{2 \cdot max(\lambda_{max}(\Sigma_1), \lambda_{max}(\Sigma_2))} \tag{5.8}$$

Avec $\lambda_{max}(\Sigma_1)$ et $\lambda_{max}(\Sigma_2)$ les plus grandes valeurs propres des matrices de covariance respectives Σ_1 et Σ_2.

Deux Gaussiennes 2-séparées sont considérées comme complètement séparées. Deux Gaussiennes 1- ou $^1/_2$-séparées se recouvrent significativement. Ces valeurs permettent d'établir les *2XC* à partir des mélanges de Gaussiennes de chaque S-VOP en quantifiant leur séparation : Soient m_1 et m_2 deux mélanges de Gaussiennes modélisant deux S-VOPs s_1 et s_2. m_1 est inclus dans m_2 si et seulement si chaque Gaussienne de m_1 est au plus 1-séparées avec l'une des Gaussiennes de m_2. L'inclusion d'un mélange dans un autre permet de regrouper les S-VOPs correspondants dans la même 2XC (cf. algorithme 3). L'utilisation de l'inclusion et non pas de l'égalité entre les mélanges permet de regrouper les S-VOPs représentant des sous parties d'un même objet d'intérêt.

Algorithme 3 : Critère de fusion des S-VOPs

si $m_1 \subset m_2$ *ou* $m_2 \subset m_1$ **alors**
| s_1 et $s_2 \in$ même 2XC ;
fin

5.3.6 Contrôle de trajectoire dans une classe couleur

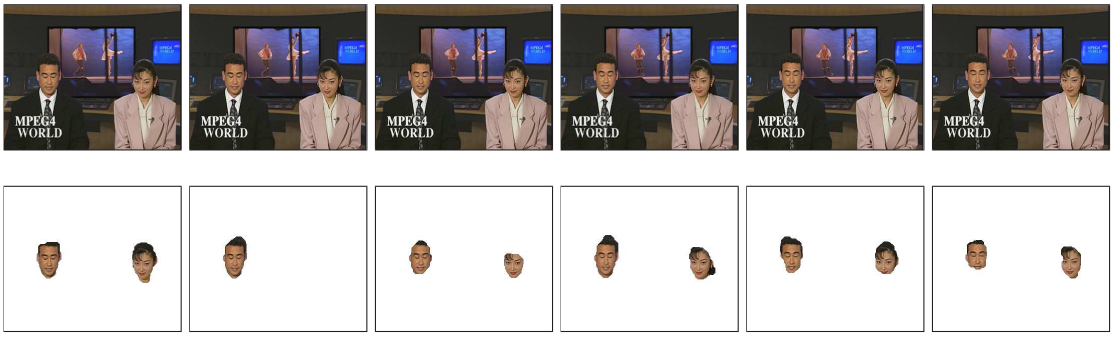
概念

Cette partie montre comment dans une classe, on élimine les S-VOPs qui ne sont pas conformes à la trajectoire estimée de l'objet d'intérêt représenté.

Problématique

Le modèle de classification couleur vu dans le paragraphe 5.3.5 ne prend pas en compte la répartition spatiale des couleurs et les informations spatio-temporelles des S-VOPs. Ce qui signifie que 2 S-VOPs issus d'objets d'intérêt différents mais pourvus de couleurs similaires, peuvent cohabiter dans la même 2XC. Afin de comprendre les problèmes inhérents aux classes couleur et la nécessité d'inclure de l'information spatiale et temporelle dans notre procédé, voici quelques illustrations utilisant des plans vidéo classiques.

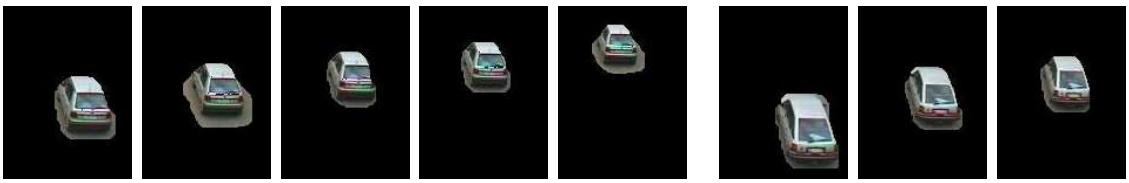
Le plan-séquence *News* permet d'illustrer les contraintes spatiales prises en compte dans le contrôle. Dans ce plan-séquence, notre algorithme extrait deux visages entre les images #178 à #183 (cf. fig. 5.9). Ces deux visages de personnes différentes, constituant donc des objets d'intérêt distincts, sont pourtant regroupés dans la même *2XC* puisque leur distribution couleur est proche. Cependant, la position des visages indique une incohérence spatiale entre les 2 S-VOPs. Nous avons donc intégré dans le contrôle, un critère spatial permettant de discriminer 2 S-VOPs de même distribution couleur mais dont l'écart spatial laisse présager leur appartenance à des objets d'intérêt différents.



(a) (b) (c) (d) (e) (f)

FIG. 5.9: Extraction de visages dans les images n^o178 à 183 du plan-séquence *News*

Le plan-séquence *Highway* permet d'illustrer l'aspect temporel du contrôle. Deux voitures relativement similaires et de même couleur se succèdent entre les images #180 et #220. Ainsi ces deux voitures sont regroupées au sein de la même *2XC* de par leurs caractéristiques couleur (cf. fig. 5.10). Cependant, leur apparition et leur dynamique sont bien distinctes. Ce qui rend possible la séparation des S-VOPs sur un critère temporel. Nous avons donc intégré dans le contrôle, des informations rendant compte de la dynamique de l'objet d'intérêt comme la direction du mouvement et son amplitude.



(a) (b)

Véhicule 1 Véhicule 2

FIG. 5.10: Composition d'une seule et même *2XC* sur le seul critère couleur (ordre chronologique)

Nous avons donc élaboré un contrôle antérieur et postérieur de la trajectoire du SVOP générateur de la classe. Ce contrôle permet de supprimer, par défaut, les S-VOPs non cohérents avec une trajectoire traduisant une dynamique réaliste d'un seul et même objet d'intérêt. Nous montrons ici une version simple de traitement qui améliore de façon certaine les résultats. Les différents paramètres (position, vitesse et taille des SVOPs) étant bruités et les mesures étant incomplètes (SVOPs manquants), il nous paraît indispensable à court terme d'améliorer cette première version par un filtre de Kalman.

Le contrôle consiste, connaissant la position et la vitesse (après compensation du mouvement dominant) du centre de gravité G_{ref} d'un S-VOP de référence S_{ref}, à rechercher itérativement dans les images voisines, les S-VOPs correspondants. La toute première référence est le S-VOP générateur S_{gen}. La recherche se fait en deux étapes : postérieurement puis antérieurement à S_{gen}. Elle est itérative à deux titres (algorithme 4) : d'une part pour parcourir les images en s'éloignant de S_{gen} et ainsi contrôler l'ensemble de la trajectoire ;

d'autre part pour parcourir tous les S-VOPs d'une image pour savoir quels sont ceux qui sont cohérents ou non à la trajectoire du S-VOP de référence courant.

La recherche se fait dans une fenêtre circulaire centrée sur la projection de $G_{ref} = (x, y)$ dont on connaît la vitesse compensée $\vec{V} = (dx, dy)$:

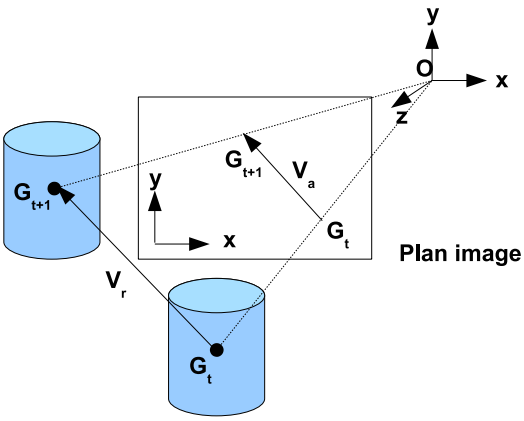
$$proj(G_{ref}) = (x + dx, y + dy) \tag{5.9}$$

Nous avons adopté cette méthode de recherche à cause de la nature sporadique des S-VOPs qui induit le fait que les $2XC$ ne peuvent pas forcément contenir un S-VOP en chaque image. Il existe 3 cas différents expliquant l'absence de S-VOPs dans une $2XC$ pendant n images.

cas 1 Immobilité apparente : Le mouvement de l'objet d'intérêt est identique au mouvement global pendant n images. L'objet est indétectable par l'algorithme et l'extraction ne peut avoir lieu.

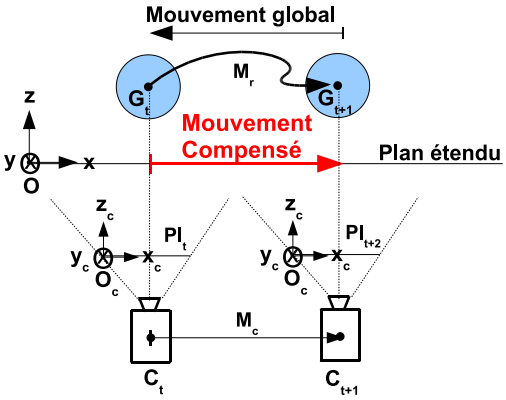
cas 2 Occultation : L'objet est totalement ou partiellement occulté pendant les n images, ce qui empêche une extraction et donc une classification correcte. L'occultation peut être due à un élément du fond (occultation passive) ou à un autre objet d'intérêt présent dans le premier plan (occultation active).

cas 3 Erreur de classification : Les modèles couleur des S-VOPs extraits dans les n images ont été déclarés différents de ceux des S-VOPs de la $2XC$.



G_t : position de l'objet à t
G_{t+1} : position de l'objet à $t+1$
V_r : mouvement réel
V_a : mouvement apparent
O : centre de projection

(a) Mouvement apparent : projection du mouvement réel dans le plan image



G_t : position de l'objet à t
G_{t+1} : position de l'objet à $t+1$
C_t : position de la caméra à t
C_{t+1} : position de la caméra à $t+1$
M_r : mouvement réel de l'objet
M_c : mouvement de la caméra

(b) Mouvement compensé de l'objet

FIG. 5.11: Illustrations des mouvements existants lors d'une prise de vue

Dans le cas de l'immobilité apparente, la position de l'objet vidéo entre le temps t et $t+n$ dépend à chaque instant du mouvement global. Il est donc possible de suivre l'objet d'intérêt en chaque image faisant évoluer la fenêtre de recherche circulaire selon le mouvement global.

Dans les *cas 2 et 3*, l'objet d'intérêt est animé d'un mouvement apparent différent du mouvement global entre les temps t et $t+n$. Ce mouvement est malheureusement inconnu.

Afin de gérer ces cas, il est donc nécessaire d'émettre des hypothèses concernant le mouvement de l'objet d'intérêt entre deux extractions. Pour cela, il faut distinguer :

1. le *mouvement réel* de l'objet d'intérêt
2. le *mouvement apparent* (mouvement observé) de l'objet vidéo
3. le *mouvement compensé* de l'objet vidéo.

Mouvement apparent : projection dans le plan image du mouvement réel $3D$ de l'objet auquel se rajoute le mouvement de la caméra (figure 5.11).

Mouvement compensé : Mouvement apparent auquel est retranché la composante due au mouvement de la caméra.

La figure 5.11 (a) illustre la différence entre le mouvement réel et le mouvement apparent d'un objet d'intérêt. La figure 5.11 (b) illustre le mouvement apparent compensé dans le cas le plus simple à schématiser qui est le cas du *travelling*.

La gestion de la perte momentanée de l'objet d'intérêt implique d'elle-même une hypothèse sur le mouvement de l'objet d'intérêt. Il semble plus raisonnable d'émettre une hypothèse sur le mouvement compensé plutôt que sur le mouvement apparent qui dépend de l'objet *et* de la caméra. Ainsi, une hypothèse est émise selon laquelle *le mouvement compensé de l'objet d'intérêt est considéré comme uniforme entre deux extractions réussies*. En prenant en compte le mouvement compensé avant la perte de l'objet et le mouvement global mis à jour à chaque image, il est possible d'estimer ses positions dans les images voisines.

Algorithme 4 : Nettoyage d'une classe couleur par contrôle de trajectoire

Données :
$2XC$, une Classe Couleur de S-VOPs
$SVOP_{gen}$, l'élément générateur de $2XC$
t_{gen}, le numéro de l'image contenant $SVOP_{gen}$
t_{fin}, le numéro de l'image contenant le dernier SVOP de $2XC$
r, le rayon de la fenêtre de recherche

Sorties :
$3XC$, la Classe Couleur correspondant à $2XC$ sans les SVOPs non cohérents

$3XC = \emptyset$;
$(x, y, dx, dy) = (G, \vec{V})_{SVOP_{gen}}$;
pour chaque *t variant de* $t_{gen} + 1$ *à* t_{fin} **faire**
  $x = x + dx$;
  $y = y + dy$;
  $\vec{V}_{ref} = (dx, dy)$;
  pour chaque $SVOP_i(t) \in 2XC$ **faire**
    si $G_i \subset Cercle(x, y, r)$ *et* $\left\langle \vec{V}_{ref} \cdot \vec{V}_{SVOP_i} \right\rangle \geq 0$ **alors**
      $3XC = 3XC \cup SVOP_i$; `/* un SVOP cohérent de plus */`
      $(x, y, dx, dy) = (G, \vec{V})_{SVOP_i}$;
      `/* il devient la nouvelle référence */`
    fin
  fin
  $t = t + 1$;
fin

FIG. 5.12: Exemple de variation du centre de gravité due à des S-VOPs incomplets et à une déformation de l'objet d'intérêt

L'algorithme 4 présente la construction des 3XC. Il est élaboré autour de l'hypothèse précédente et prend en compte la variabilité de la position des centres de gravité des S-VOPs d'un même objet d'intérêt (figure 5.12).

Le rayon r de la fenêtre de recherche est calculé relativement à l'élément générateur de la *2XC* :

$$r = \max_{(x,y)} \|G_g en - p(x,y)\| \tag{5.10}$$

Avec G_{gen} : le centre de gravité de l'élément générateur et $p(x,y)$ un pixel appartenant à l'élément. Un élément candidat $S_i(t)$ est cohérent temporellement avec la trajectoire de l'élément de référence S_{ref} si et seulement si :

$$\|G_{S_{ref}} - G_{S_i(t)}\| \leqslant r \text{ et } \left\langle \vec{V}_{S_{ref}} \cdot \vec{V}_{S_i(t)} \right\rangle \geq 0 \tag{5.11}$$

La première condition assure la cohérence spatiale, tandis que la seconde assure la conformité des directions du mouvement des S-VOPs successifs appartenant à la 3XC ($\langle \cdot \rangle$ représente le produit scalaire).

Pour une image donnée $I(t)$, la recherche s'effectue en respectant les règles suivantes :

1. Si aucun (centre de gravité de) SVOP n'est inclus dans la fenêtre de recherche, la recherche recommence dans l'image suivante ou précédente (en fonction d'une recherche postérieure ou antérieure à l'élément générateur), la position de la fenêtre de recherche étant alors incrémentée du vecteur vitesse du SVOP de référence.

2. Si un seul SVOP est inclus dans la fenêtre de recherche et qu'il vérifie la contrainte de direction, il est conservé dans la classe et c'est lui qui devient la nouvelle référence (position et vitesse).

3. Si plusieurs SVOP sont inclus dans la fenêtre de recherche et qu'ils vérifient la contrainte de direction, ils sont conservés dans la classe et c'est leur centre de gravité qui devient la nouvelle référence.

4. Tous les S-VOPs dont le centre de gravité est extérieur à la fenêtre de recherche sont exclus de la classe.

Après cette étape, chaque classe (3XC) est censée contenir uniquement des S-VOPs se rapportant à un seul objet d'intérêt. Cependant, à chaque objet d'intérêt est associé plusieurs 3XC. L'objectif de l'étape suivante est de fusionner les 3XC se rapportant au même objet d'intérêt afin de générer les classes-clés.

5.3.7 Fusion hiérarchique des classes couleur

概念

A ce stade, on dispose toujours d'autant de classes que de S-VOPs. Dans cette partie, je montre comment il faut fusionner ces classes afin d'avoir d'une part des classes disjointes et d'autre part une seule classe par objet d'intérêt.

Pour réaliser cette étape, une classification hiérarchique ascendante a été choisie. L'indice de dissimilarité et le critère d'agrégation sont présentés dans ce paragraphe. La méthode de classification et le contrôle de trajectoire des 2XC induit le fait qu'un grand nombre des 3XC comportent des éléments communs et se rapportent au même objet d'intérêt. Grâce à la présélection, le nombre d'éléments par classe est relativement restreint et se limite aux éléments pertinents. Or, la classification de ces éléments est relativement robuste. Ainsi, il est émis une hypothèse selon laquelle les 3XC concernant le même objet d'intérêt ont un contenu très proche et peuvent être fusionnées sur l'étude de la similarité de leur contenu. Ceci revient à répondre à la question : une classe est-elle globalement incluse dans une autre ? Si oui, les deux classes n'en font plus qu'une.

Indice de dissimilarité

La théorie des ensembles permet de modéliser simplement le problème. En effet, les 3XC peuvent être assimilées à des ensembles et leur similarité est alors évaluée à partir de l'étude de leur intersection au sens de la théorie des ensembles (cf. figure 5.13). La fusion des 3XC est effectuée selon une méthode de classification hiérarchique ascendante (« agglomérative »).

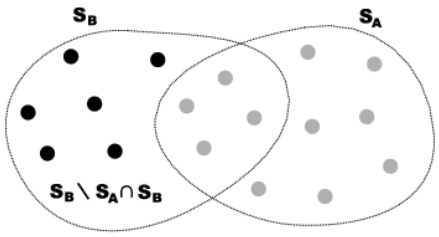


Fig. 5.13: Exemple d'intersection de deux classes

Soit Ω l'ensemble des 3XC. Initialement chaque individu (ici les 3XC) de Ω forme une classe. Il existe une dissimilarité d entre chaque couple de Ω (dissimilarité : distance sans l'inégalité triangulaire). Cette dissimilarité entre deux 3XC : S_A et S_B qui vérifient $|S_B| \leq |S_A|$[5] est donnée selon une notation ensembliste par l'expression suivante :

$$d = \frac{|S_B \setminus S_A \cap S_B|_6}{|S_B|} \tag{5.12}$$

$d = 1$ lorsque l'intersection entre les ensembles est vide et $d = 0$ lorsque $S_B \subset S_A$.
d vérifie :

1. la non-négativité : $d(S_A, S_B) \geq 0$

[5] $|S_A|$ représente le cardinal de S_A

[6] $S_B \setminus S_A \cap S_B$ désigne l'ensemble de tous les éléments de S_B qui n'appartiennent pas à $S_A \cap S_B$, autrement dit, ceux qui n'appartiennent qu'à S_B (figure 5.13)

2. la symétrie : $d(S_A, S_B) = d(S_A, S_B)$
3. la normalisation : $d(S_A, S_B) = 0$ si et seulement si $S_A = S_B$

d n'est pas une distance puisqu'elle ne vérifie pas l'inégalité triangulaire :

$$d(S_A, S_B) \leq d(S_A, S_C) + d(S_C, S_B)$$

Agrégation des classes

L'indice de dissimilarité permet d'agréger itérativement et deux à deux, les classes les plus similaires jusqu'à ce qu'il n'en reste plus qu'une. Cependant, à chaque agrégation, il est nécessaire de mettre à jour l'indice de dissimilarité entre la classe nouvellement formée et toutes les autres. Cette mise à jour est régit par le critère d'agrégation. Soit $c_3 = c_1 \cup c_2$. Il existe plusieurs critères d'agrégation entre les deux classes c_3 et c de Ω, définis à partir d'une dissimilarité d sur Ω.

- le critère du saut minimum (*single linkage*) :
  Les distances entre classes sont déterminées par la plus petite distance existant entre deux éléments de classes différentes (c'est à dire les « 3XC les plus proches »). C'est cet indice qui est utilisé par la suite. Les dissimilarités sont recalculées à chaque étape à l'aide de la règle suivante :

$$d(c_3, c) = min[d(c_1, c), d(c_2, c)]$$

- le critère du saut maximum (*complete linkage*) :
  Les distances entre deux classes sont déterminées par la plus grande distance existant entre deux éléments de chaque classe (c'est à dire les « 3XC les plus éloignées »). Les dissimilarités sont recalculées à chaque étape à l'aide de la règle suivante :

$$d(c_3, c) = max[d(c_1, c), d(c_2, c)]$$

- le critère de Ward :
  Cette méthode se distingue de toutes les autres en ce sens qu'elle utilise une analyse de la variance approchée afin d'évaluer les distances entre classes. Elle consiste à limiter à chaque fusion la perte d'inertie intra-classe. Les dissimilarités sont recalculées à chaque étape à l'aide de la règle suivante :

$$d(c_3, c) = \frac{(n_{c_1} + n_c)d(c_1, c) + (n_{c_2} + n_c)d(c_2, c) - n_c d(c_1, c_2)}{n_c + n_{c_1} + n_{c_2}}$$

  Avec n_c, n_{c_1} et n_{c_2} les poids respectifs des classes c, c_1 et c_2.
  La méthode de Ward se justifie bien lorsque la dissimilarité entre les individus est le carré de la distance euclidienne. Choisir de regrouper les deux individus les plus proches revient alors à choisir la paire de points dont l'agrégation entraîne la diminution minimale de l'inertie du nuage. Le calcul des nouveaux indices entre la paire regroupée et les points restants revient alors à remplacer les deux points formant la paire par leur point moyen, affecté du poids 2.

L'indice d'agrégation qui correspond le mieux à la nature des 3XC et à la recherche d'une inclusion entre ces 3XC est l'indice du saut minimum. Cette mise à jour particulière permet d'avantager des fusions centrées sur les classes les plus fédératrices. En effet, une classe peut contenir plusieurs 3XC totalement distinctes qui s'incluent dans des 3XC contenant plus d'éléments. Ces dernières peuvent être considérées de manière imagée comme des classes « liantes ».

Représentation sous forme de dendrogramme

Le résultat de la classification ascendante hiérarchique est plus facilement visualisable sous forme de dendrogramme (cf. fig. 5.14). Un dendrogramme nous donne la composition des différentes classes ainsi que l'ordre dans lequel elles ont été formées. L'axe vertical renseigne sur la valeur de l'indice d'agrégation pour un groupement donné. Cette représentation permet également de visualiser les sauts de l'indice afin de définir une éventuelle partition.

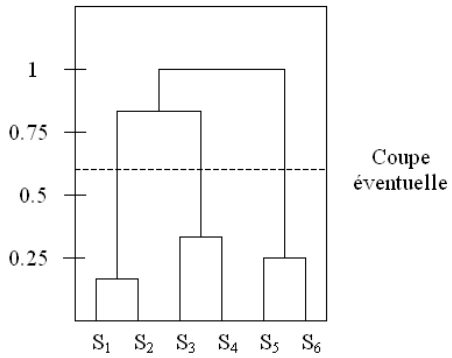


FIG. 5.14: Exemple d'un dendrogramme de 6 *3XC* - La coupe induit 3 classes

Détermination des classes-clés

La détermination du nombre de classes-clés est effectuée par classification des dissimilarités. Il est généralement pertinent de couper le dendrogramme de classification à l'endroit où est observé un saut dans les valeurs d'agrégation. Il est alors possible d'obtenir une partition de bonne qualité car les individus regroupés en dessous de la coupure sont proches tandis que les individus situés au dessus sont éloignés.

On calcule donc un seuil qui va maximiser l'inertie I_i entre deux sous-ensembles E_i et F_i. Soit E_i l'ensemble des dissimilarités ≥ 0 et $< i$. Soit F_i l'ensemble des dissimilarités $\geq i$ et < 1 (on exclut les dissimilarités égales à 1 qui indiquent que deux *3XC* sont disjointes). Soit $D = E_i \cup F_i$. À D, E_i, F_i on associe leur moyenne respective m_D, m_{E_i}, m_{F_i}. L'inertie est donnée par :

$$I_i = w_e d(m_{E_i}, m_D)^2 + w_f d(m_{F_i}, m_D)^2 \tag{5.13}$$

Où $w_e = |E_i|, w_f = |F_i|$ et d est la distance Euclidienne.

Le meilleur partitionnement est atteint pour une valeur de i qui maximise l'inertie. Toutefois, si ce seuil est trop faible, le risque est d'obtenir trop de classes. Pour cette raison, on définit empiriquement un seuil minimum m. Le seuil d'agrégation recherché est donc :

$$S_a = min(m, \underset{i}{\text{argmax}}(I_i)) \tag{5.14}$$

Le minimum m est placé à 0.5 afin de fusionner les classes ayant au moins en commun la moitié de leurs éléments. Comme le montre la figure 5.14, le calcul de S_a permet de fixer directement le nombre et la constitution des différentes classes-clés.

L'expérience montre que cette étape de fusion suffit à rendre compte avec efficacité des objets prédominants dans le plan puisque le nombre final de classes et leur contenu correspond bien aux objets d'intérêt.

5.4 Suppression des classes temporellement non significatives

概念

Cette étape permet de contrôler la validité temporelle des classes obtenues. Dans cette section, on définit deux critères simples permettant de supprimer les classes temporellement non significatives : la *durée* et la *persistance*.

A cause d'erreurs de l'estimation de mouvement dues à des images bruitées, il n'est pas rare d'observer parmi les S-VOPs extraits des régions très localisées issues du fond et ce sur quelques images (cf. figure 5.15). L'extraction de ces régions peut s'avérer robuste ce qui génère automatiquement une classe-clé. Il est donc nécessaire de mettre au point un critère permettant de supprimer ce type de classes parasites. Pour cela, il est important de prendre en

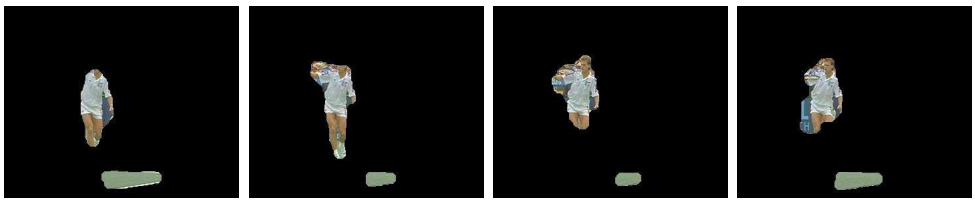


FIG. 5.15: Extractions parasites des régions du fond (images 32 à 35 du plan-séquence Stefan Edberg)

compte ce qu'une classe-clé représente. En effet, elle représente un objet d'intérêt qui est sensé « jouer un rôle » dans le plan. Il est raisonnable de penser qu'un objet d'intérêt reste à l'image au moins quelques secondes durant un plan. Ainsi, un objet est considéré comme un objet d'intérêt du plan si sa durée d'apparition est significative. De plus, d'après les hypothèses, les classes-clés ne contiennent pas forcément des éléments successifs à cause d'éventuelles détections ou extractions erronées voire impossibles. Cependant, plus la classe-clé contient d'éléments plus celle-ci sera considérée comme pertinente.

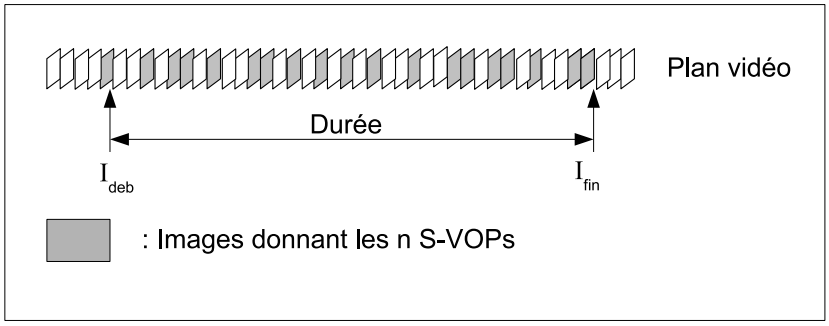


FIG. 5.16: Structure d'une classe

Pour une classe donnée, nous savons qu'elle contient n S-VOPs (cf. figure 5.16) dont les première et dernière apparitions chronologiques sont I_{deb} et I_{fin}. Nous en déduisons la durée (en nombre d'images) et la persistance (ou taux d'apparition) calculée sur la durée. Nous faisons l'hypothèse qu'un objet d'intérêt reste à l'image pendant une durée significative d et que pendant cette durée, il est extrait $p\%$ du temps. d et p sont fixés expérimentalement ($d = 50$ i.e. 2 secondes et $p = 20\%$ par exemple) et permettent de valider ou de supprimer chacune des classes (algorithme 5).

Algorithme 5 : Critère temporel

Données :
C une classe-clé
I_{deb} le plus petit numéro d'image où apparaît un S-VOP $\in C$
I_{fin} le plus grand numéro d'image où apparaît un S-VOP $\in C$
n le nombre de S-VOP $\in C$

$duree = I_{fin} - I_{deb} + 1$;
$persistance = n/duree$;
si $duree < 50$ *ou* $persistance < 0.2$ **alors**
| C est supprimée ;
fin

5.5 Sélection de l'objet-clé et des vues-clés

概念

Nous disposons maintenant des classes-clés définitives. Selon nos hypothèses, une classe correspond à un objet d'intérêt. Dans cette section, j'explique comment nous sélectionnons les représentants d'une classe : l'objet-clé et plusieurs vues-clés.

5.5.1 Objet-clé

On peut maintenant sélectionner un unique objet-clé dans chaque classe C. L'équation 5.3 page 115 présente le critère qui a permis d'estimer la qualité d'un masque de S-VOP en terme de segmentation. Ce critère est à nouveau utilisé ici, et c'est le S-VOP qui maximise le critère dans un sous-ensemble $\widehat{C}$ de C qui est l'objet-clé de la classe.

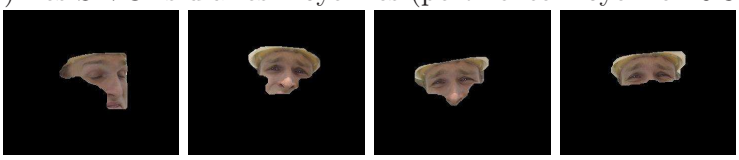
Voici comment est composé $\widehat{C}$: comme le critère utilisé s'exprime en pourcentage, les petits S-VOPs sont avantagés au détriment des plus grands. Privilégier les régions d'aire faible peut donc s'avérer problématique. En effet, le risque est de sélectionner les S-VOPs ne représentant qu'une sous-partie de l'objet d'intérêt et de passer à côté des régions plus représentatives de la totalité de l'objet d'intérêt. Les grandes régions sont également à manipuler avec précaution car elles peuvent être le résultat d'une fuite de l'objet vers le fond, ce qui explique l'augmentation de l'aire du S-VOP. Pour contourner ce problème de maximum local, on estime l'intervalle le plus représentatif des aires de C : C est découpée en 3 sous-ensembles disjoints selon les aires de ses S-VOPs : faibles, moyennes et élevées (figure 5.17). Cette classification est à nouveau réalisée par agglomération autour de centres mobiles. $\widehat{C}$ est le sous-ensemble dont la qualité moyenne des masques est la plus élevée. C'est celui-ci qui fournit l'objet-clé.

5.5.2 Vues-clés

Afin de résumer un maximum l'information contenue dans chaque classe, nous proposons ici d'extraire automatiquement des S-VOPs supplémentaires constituant des vues-clés et représentant des aspects différents de l'objet d'intérêt. Notons qu'un objet-clé constitue une référence précise et de qualité de l'objet d'intérêt correspondant. L'objet-clé va donc nous servir, ici, de modèle pour étudier la population de la classe-clé et ainsi orienter la sélection des vues-clés qui regroupent : une *vue complémentaire*, un aperçu des zooms et une visualisation de la composition des sous-parties mobiles de l'objet d'intérêt.

(a) Les S-VOPs d'aires élevées (pertinence moyenne : 0.42)

(b) Les S-VOPs d'aires moyennes (pertinence moyenne : 0.37)



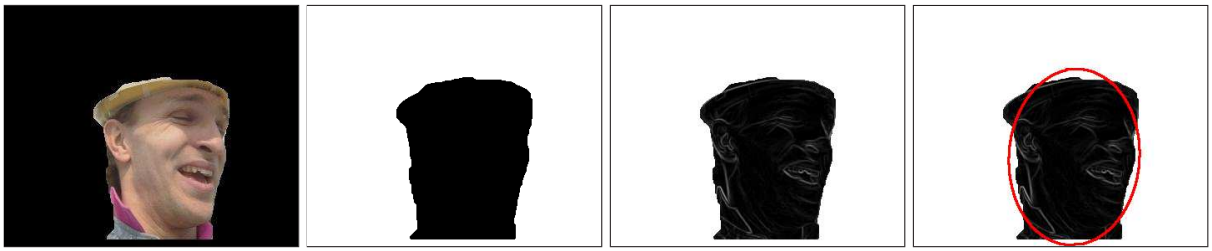
(c) Les S-VOPs d'aires faibles (pertinence moyenne : 0.41)

(d) L'objet-clé

FIG. 5.17: Découpage d'une classe en 3 sous-ensembles : le sous ensemble (a) qui fournit les masques de meilleure qualité fournit également l'objet-clé (d)

Vue complémentaire

L'objectif de cette étape est de sélectionner une vue complémentaire de l'objet-clé parmi les S-VOPs de la classe-clé. Le principe est de rechercher dans l'ensemble $\widehat{C}$, le S-VOP dont la répartition de l'information contour contenue dans le masque (cf. fig. 5.18.b et 5.18.c) est la plus dissemblable de celle de l'objet-clé. Pour cela, chaque S-VOP d'une classe-clé est modélisé par une ellipse [58] (cf. fig. 5.18.d) permettant de modéliser la répartition spatiale de l'information contour contenue dans le S-VOP : chaque quartier i du S-VOP fournit une valeur moyenne de la norme du gradient μG_i, calculée sur les pixels qu'elle contient.



(a) S-VOP extrait de l'image #111 de Foreman)

(b) Masque du S-VOP

(c) Norme du gradient

(d) Modélisation ellipsoïdale

FIG. 5.18: Extraction des données contour

Afin de comparer au mieux les ellipses des différents S-VOPs et celle de l'objet-clé, il est

nécessaire de les orienter de manière identique. Les deux axes principaux de l'ellipse la découpe en 4 quartiers numérotés dans le sens horaire comme le montre la figure 5.19).

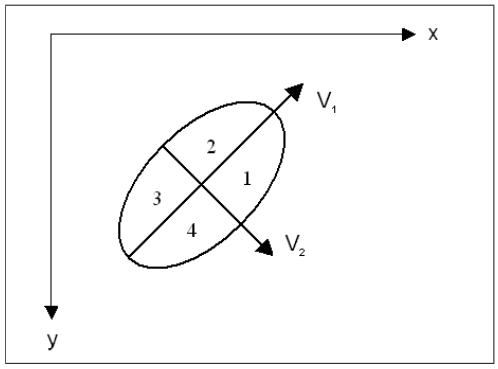


FIG. 5.19: Découpage de l'ellipse

Les μG_i permettent d'établir une distance d (cf. eq. 5.15) entre deux ellipses e_1 et e_2 pour chacune des positions possibles correspondant à une permutation circulaire p des quartiers entre e_1 et e_2.

$$d(e_1, e_2, p) = \sqrt{\sum_{i=1}^{4} (\mu G(e_1)_i - \mu G(e_2)_{(p+i) \bmod 4})^2)} \text{ avec } p \in [\![0, 3]\!] \tag{5.15}$$

La meilleure correspondance entre l'ellipse de l'objet-clé et celle du S-VOP candidat est celle qui minimise la différence d et c'est celle-ci qui est utilisée par la suite.

Soient e_{oc} l'ellipse de l'objet-clé et e_c l'ellipse du S-VOP candidat. La vue-clé d'une classe est le S-VOP qui maximise la meilleure correspondance :

$$I_c = \max_c(\min_p(d(e_{oc}, e_c, p))) \tag{5.16}$$

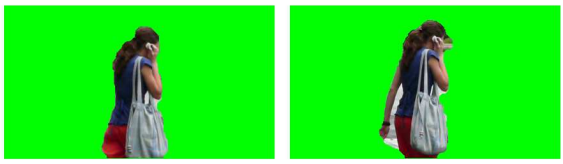
Comme le montre l'équation 5.16, la vue-clé est le S-VOP dont l'ellipse correctement comparée (la correspondance la plus probable parmi 4) est la plus dissemblable en terme de contours à l'ellipse de l'objet-clé. Les figures 5.20 et 5.21 présentent la sélection d'un objet-clé et d'une vue-clé complémentaire accompagnés des vue supplémentaires de $\widehat{C}$ pour deux plans-séquences.

Afin de sélectionner n vues-clés ($n > 1$), il faut calculer I_c pour chaque paire de S-VOPs de $\widehat{C}$. Le principe est alors de trouver l'ensemble des n S-VOPs maximisant les différences entre eux. Ce travail n'a pas été réalisé par manque de temps mais constitue une fonctionnalité intéressante pour l'utilisateur.

Étude du zoom

Dans le cas particulier où un zoom positif et/ou négatif est effectué sur l'objet durant le plan, chacun des sous-ensembles déterminés par les aires des S-VOPs en plus de $\widehat{C}$, représente un état pertinent de l'objet d'intérêt. La figure E.1 en annexe E présente une classification des S-VOPs selon leur aire. Ils sont rangés dans l'ordre décroissant de pertinence au sein d'un même sous-ensemble. Dans le cas présenté dans cette figure, l'objet-clé est choisi dans la classe d'aire faible. Dans ce cas, l'objet-clé et la vue supplémentaire ne peuvent pas rendre compte de cette transformation. Les S-VOPs présents dans l'ensemble $C \setminus \widehat{C}$ peuvent apporter cette information manquante sur l'objet (cf. figure E.1).

(a) Aperçu du plan-séquence



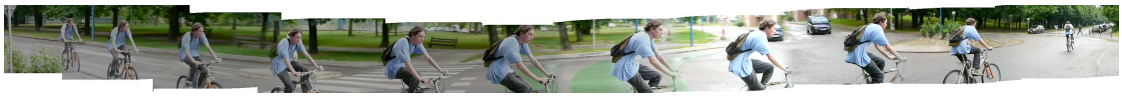
(b) L'objet-clé et sa vue complémentaire

(c) Vues supplémentaires de la classe $\widehat{C}$

FIG. 5.20: Sélection de l'objet-clé et de sa vue complémentaire

Pour cela nous proposons de modéliser chaque S-VOP de la classe $C \setminus \widehat{C}$ par un mélange de Gaussiennes m_{svop} calculé cette fois, sur l'ensemble des pixels du S-VOP (cf. paragraphe 5.3.5). L'objectif est de définir un ensemble C_{id} de S-VOPs qui vérifient l'identité entre leur mélange couleur et celui de l'objet-clé, noté m_{oc}. Les mélanges m_{svop} et m_{oc} sont identiques si chaque Gaussiennes de m_{svop} est au plus 1-séparée d'une Gaussienne de m_{oc} de poids similaire ($\bigwedge w \leq 0.05$). L'ensemble C_{id} est ensuite scindé en deux ensembles $C_{id}-$ et $C_{id}+$ regroupant les S-VOPs de mélange identique à l'objet-clé mais de taille respectivement inférieure et supérieure à celui-ci. Enfin, nous sélectionnons le S-VOP le plus pertinent au sens du coefficient donné par l'équation 5.3 page 115 dans chacun des deux ensembles $C_{id}-$ et $C_{id}+$ afin de rendre compte du zoom négatif et positif. Pour que l'information apportée soit significative, la taille des S-VOPs sélectionnés doit être supérieure ou inférieure d'au moins 10% de celle de l'objet-clé. La figure 5.21 présente le résultat de la sélection de la vue supplémentaire et des zooms obtenus pour un plan-séquence d'approximativement 130 images.

Notons que l'extraction de vues-clés relatives aux zooms est contingente dans le sens où elle dépend d'une part, de l'existence de zooms et d'autre part, de l'extraction effective d'un S-VOP pertinent pendant la durée des zooms.



(a) Aperçu du plan-séquence (∼ 130 images)

(b) L'objet-clé

(c) La vue complémentaire

(d) Zoom +

(e) Zoom -

(f) Vues supplémentaires de $\widehat{C}$ placées par ordre décroissant de pertinence

FIG. 5.21: Sélection des vues-clés caractérisant les zooms

Étude de la composition de l'objet-clé

La modélisation couleur par mélanges de Gaussiennes des S-VOPs de $C \setminus \widehat{C}$ vue au paragraphe précédent permet d'étudier les relations d'inclusion qui peuvent exister entre ces S-VOPs et l'objet clé. L'objectif est d'extraire des S-VOPs représentant des sous-éléments mobiles de l'OC ainsi qu'éventuellement un sur-élement. L'ensemble $C \setminus \widehat{C}$ est de nouveau scindé en deux groupes :

1. Ensemble des sous-éléments : $C_{/s} = \{\text{S-VOP} \mid m_{svop} \subset m_{oc}\}$
2. Ensemble des sur-éléments : $C_{s/} = \{\text{S-VOP} \mid m_{oc} \subset m_{svop}\}$

Enfin, le S-VOP le plus pertinent au sens du coefficient donné par l'équation 5.3 page 115 est sélectionné dans chacun des deux ensembles $C_{/s}$ et $C_{s/}$ (cf. figure 5.26). Par simplicité et manque de temps, nous nous sommes limités à un seul sous-élément de l'OC. Par la suite, il sera intéressant de classer les sous-éléments de l'OC afin de fournir à l'utilisateur les sous-éléments distincts du même OC.

5.6 Création de résumés de vidéos

Nous proposons dans ce paragraphe une mise en forme particulière des informations extraites automatiquement par notre méthode, en vue de la représentation d'un plan vidéo. Le résultat se compose donc de 3 types de visualisations (cf. figure 5.22) :

1. Images-clés
2. S-VOPS
3. Composition de l'objet-clé

L'objet-clé, sa vue supplémentaire et les éventuelles vues issues de zooms, fournissent les images-clés. Ces images peuvent éventuellement servir à créer un résumé du plan. A chaque image-clé est associé le masque du S-VOP extrait. La dernière partie propose les relations d'inclusion entre l'objet-clé et d'éventuels sur-élément et sous-élément détectés par notre algorithme. La première visualisation permet une meilleure compréhension de la scène par l'utilisateur/spectateur, en proposant l'objet d'intérêt dans son environnement. L'objet est mis en évidence et localisé dans l'image par une ellipse. La deuxième visualisation est, quant à elle, plus appropriée aux méthodes d'indexation fondées sur les objets. Elle comporte le masque du S-VOP flou (l'objet-clé, vue complémentaire et zooms) issu de l'objet d'intérêt. La troisième visualisation fournit des renseignements sur la composition de l'objet d'intérêt.

Les vues et les S-VOPS supplémentaires concernant les zooms et les relations d'inclusion étant contingents, ils sont remplacés par la mention "N/A" (Non Applicable) lorsqu'ils n'ont pas été détectés.

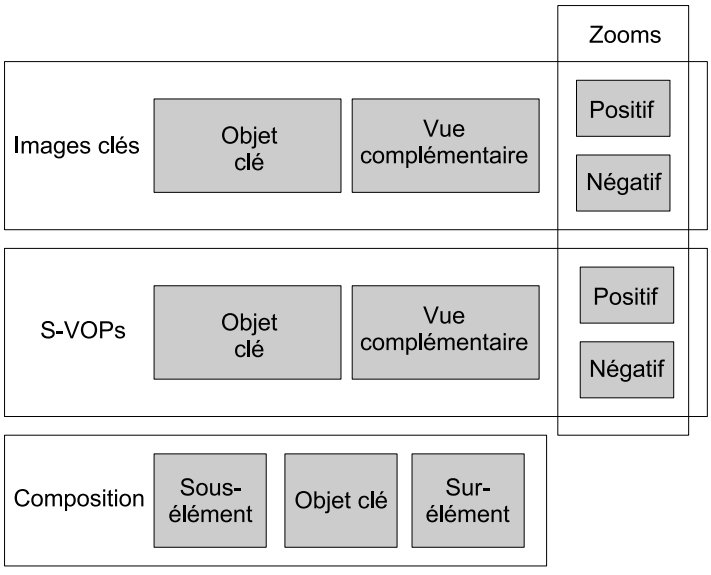


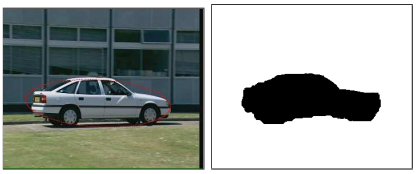
FIG. 5.22: Mise en forme des résultats pour le résumé d'un plan vidéo

5.7 Extension au suivi d'objet

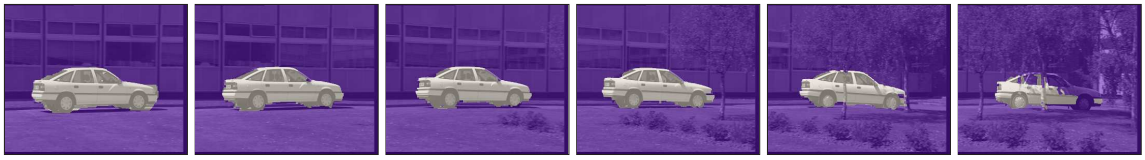
概念 L'objectif est d'initialiser une méthode de suivi d'objet précise afin d'extraire image par image l'objet d'intérêt représenté par l'OC.

5.7.1 Initialisation

Bien que notre méthode puisse être associée à une méthode de suivi dans les cas où l'objet est animé d'un mouvement apparent net, il n'est pas forcément possible d'extraire à chaque image l'objet d'intérêt. L'idée est donc d'initialiser une méthode de suivi précise par projection de partition pour obtenir une extraction à chaque image (cf. figure 5.23).



(a) Image #36 de *Vectra* (b) Masque de l'objet-clé



(c) Objet-clé #36 (d) Suivi à l'image #45 (e) Suivi à l'image #54 (f) Suivi à l'image #63 (g) Suivi à l'image #74 (h) Suivi à l'image #83

FIG. 5.23: Étude du comportement de l'objet par suivi spatio-temporel (plan-séquence *Vectra*)

L'étape de sélection de l'objet-clé permet d'obtenir un masque binaire relativement caractéristique de l'objet d'intérêt. Bien qu'il ne recouvre généralement pas totalement l'objet d'intérêt, la correspondance frontière/contour du masque avec l'objet est de qualité suffisante pour envisager un suivi plus précis tel que celui présenté par G. Foret dans [40]. Le masque de l'objet-clé fournit une initialisation automatique intéressante pour ce type d'application dont le principale défaut est l'initialisation manuelle. Ce suivi permet d'étudier en détails le comportement de l'objet d'intérêt dans le plan et de collecter de nombreuses informations [40].

Le principe est d'engager un suivi avant et arrière, de part et d'autre de l'OC. En annexe F, la figure F.1.a présente le suivi *avant* initialisé par l'objet-clé (encadré en vert). Quant à la figure F.1.b, elle présente le suivi *arrière*.

Il apparaît que pour le suivi arrière la qualité de l'extraction se dégrade au cours du procédé. Nous avons imaginé une méthode de contrôle utilisant les différents S-VOPs de la classe-clé de l'objet d'intérêt, extraits automatiquement par notre méthode.

5.7.2 Contrôle

L'extraction image par image par projection de partition est soumise à plusieurs inconvénients qui sont les fuites et les occultations. Pour un objet d'intérêt donné, le fait de disposer d'un ensemble de masques de qualité que sont les S-VOPs de la classe-clé, permet de contrôler ce type de suivi. Le principe est de remettre à jour le suivi grâce aux S-VOPs pertinents de la classe-clé, aux images correspondantes. Les résultats de contrôle présentés en annexe F à la figure F.1.c sont obtenus avec les S-VOPs issus de $\widehat{C}$. Les différents S-VOPs de contrôle sont repérés par un cadre bleu. Le contrôle permet également de pallier le problème d'occultation qui constitue une des principales limites des méthodes de suivi par projection de partition. Les résultats de la figure F.2 proposés en annexe F sont obtenus avec des S-VOPs choisis manuellement dans C (encadrés en bleu). L'occultation est signifiée par un rectangle rouge.

Par la suite, nous comptons automatiser la sélection des S-VOPs de contrôle. Le principe est de confronter le résultat obtenu par le suivi et le S-VOP de l'image correspondante. Si les divergences sont trop importantes, il est possible de prendre la décision de réinitialiser le procédé de suivi à l'aide de ce S-VOP de contrôle.

5.8 Résultats et discussion

Le premier résultat que nous présentons dans la figure 5.24 présente un aperçu des S-VOPs extraits lors d'un plan. Il dure 3 secondes et comporte 90 images. On y voit un vélo qui rentre dans le champ de la caméra, le traverse et en sort. La caméra est immobile mais est tenue à la main. L'objet cycliste a une surface qui varie d'un facteur 3 environ (figure 5.24.c). Le traitement extrait une classe-clé comportant 14 S-VOPs (figure 5.24.a). Cet exemple montre bien que la méthode peut être utilisée comme suivi à part entière d'objets en mouvement. La figure 5.24.b montre les images originales d'où sont extraits les S-VOPs. On remarque que le fond est complexe et que la cycliste n'est pas bien contrastée avec le fond. Néanmoins, sans être parfaits, les différents S-VOPs sont assez stables et descriptifs par rapport à l'objet d'intérêt.

La seconde vidéo, nommée *Chavant* montre la circulation en ville. La caméra, toujours tenue à la main, se comporte de diverses façons : elle est fixe, puis effectue quelques panoramiques dans le sens des véhicules puis dans le sens contraire. Ces mouvements sont accompagnés de zooms avant et arrière. Douze véhicules traversent le champ de la caméra de droite à gauche. Deux personnages passent au premier plan et se croisent. La vidéo dure 18 secondes c'est-à-dire 540 images. 12 objets-clés sont extraits (figure 5.25.a) au lieu de 14 espérés. Parmi eux, 6 voitures de couleur gris métallisé très similaires et deux piétons. La segmentation est de bonne qualité, les objets-clés ne débordent pas sur le fond et il est assez facile par exemple de reconnaître le modèle de chaque véhicule. Deux voitures (identiques) ne sont pas extraites. Elles se suivent et sont partiellement occultées par des poteaux qui les "découpent" en plusieurs morceaux (figure 5.25.b). On peut supposer qu'elles ont généré d'une part peu de S-VOPs (la caméra ne les suit pas) et d'autre part des S-VOPs trop petits. En conséquence de quoi, les classes correspondantes ont dû être supprimées.

La figure 5.26 présente les résumés vidéos de divers plans vidéos dont les objets sont relativement difficiles à extraire que ce soit à cause de leur mouvement faible (cf. figure 5.26.a[7] et 5.26.c) ou de leur forme complexe. Cependant, nous pouvons voir que les objets-clés sont généralement de bonne qualité bien qu'ils intègrent quelques petites régions du fond. La sélection de la vue supplémentaire est systématique. C'est pourquoi, si l'objet ne change pas d'apparence la vue supplémentaire reste assez similaire à l'objet-clé.

[7] Résultats obtenus sur une séquence vidéo partagée sur le site internet *blip.tv*, intitulée *Simple Do's and Don'ts* n^o 15. Distribution sous licence *Creative Commons* : Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique

(a) Les 14 S-VOPs extraits

(b) Les images originales correspondantes

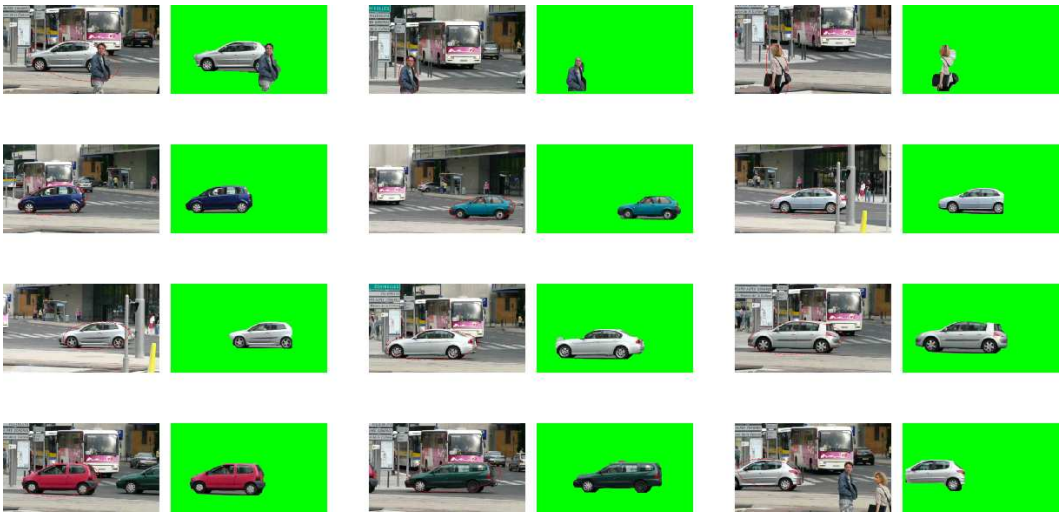
(c) Mixage des images 25, 39 et 63 montrant le léger bouger de la caméra et la variation de taille de l'objet d'intérêt

FIG. 5.24: Extraction d'un objet-clé dans la séquence *vélo* (les numéros des images sont indiqués)

Les taux de fausses et de vraies détections des zooms et des inclusions permettant de représenter la composition de l'objet n'ont pas été étudiés. La sélection de ces vues est présentée dans ce manuscrit afin de montrer les diverses possibilités de l'utilisation des classes-clés et des S-VOPs qu'elles contiennent : une fois que l'objet-clé a été déterminé, la population de la classe-clé associée peut être étudiée et caractérisée plus en détails. L'utilisation des S-VOPs pertinents d'une classe clé, bien qu'elle semble expérimentalement correcte, devra être complétée par la suite.

5.9 Conclusion

L'étape de sélection de l'objet-clé permet d'obtenir un masque binaire relativement caractéristique de l'objet d'intérêt. Dans le cas du résumé vidéo, il permet de focaliser l'attention du



(a) Ici, on présente un couple d'images par objet-clé : l'image originale où a été extrait l'objet-clé et le masque correspondant

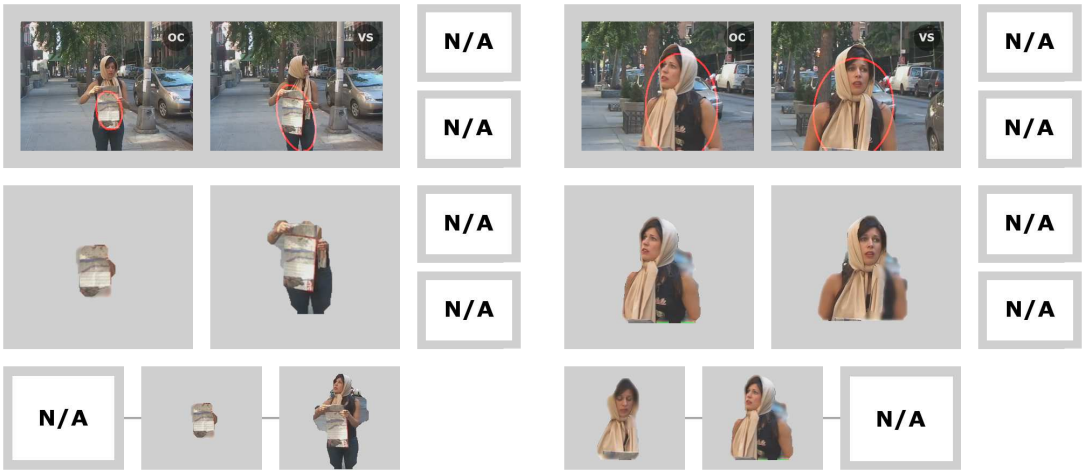
(b) Les 2 voitures non détectées par la technique

FIG. 5.25: Extraction de 12 objets-clés dans la séquence *Chavant*

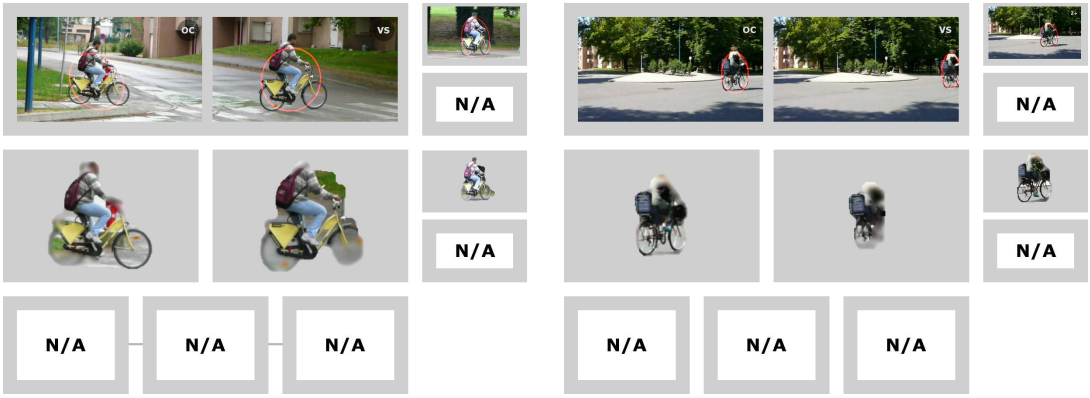
spectateur sur l'objet d'intérêt tout en fournissant un masque binaire utilisable pour des applications de type indexation. Bien que n'étant pas parfaite, la qualité sémantique des objets-clés permet dans la plupart des cas, une reconnaissance relativement aisée de l'objet d'intérêt.

En ce qui concerne la qualité de segmentation des objets-clés, bien qu'ils ne recouvrent généralement pas totalement l'objet d'intérêt, la correspondance frontière/contour du masque avec l'objet est de qualité suffisante pour envisager une initialisation et/ou un contrôle efficace de suivi comme nous l'avons vu au paragraphe 5.7. Le masque de l'objet-clé fournit une initialisation automatique intéressante pour ce type d'application dont le principal défaut est l'initialisation manuelle. Tandis que les vues-clés permettent de contrôler le suivi et de gérer les occultations de l'objet d'intérêt. Cette méthode de contrôle du suivi semble prometteuse mais nécessite encore quelques améliorations quant à l'automatisation de la sélection des S-VOPs de contrôle.

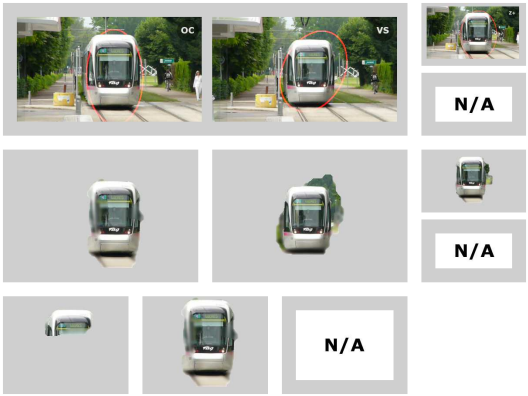
La phase d'évaluation de notre approche n'est pas encore réalisée. Comme nous avons orienté notre recherche dans une direction peu suivie, il est difficile d'envisager une procédure assez simple et systématique. Toutefois, il nous semble probable que les critères subjectifs sont plus représentatifs que les critères objectifs. Cette démarche ne peut être envisagée sérieuse-



(a) Objets peu mobiles



(b) Objets en translation



(c) Objet en translation vers la caméra

FIG. 5.26: Résumés de plans vidéo fondés sur les objets

ment qu'avec l'aide de spécialistes du domaine psycho-visuel avec lesquels il faudra entamer une collaboration. La seconde possibilité est plus pragmatique et consiste à voir si notre technique peut répondre à des besoins réels : à cet effet, nous sommes en relation avec une *startup* française qui désire améliorer ses outils de production de publicité interactive : IDside, Paris[8]. Avec ces partenaires, nous avons commencé à évaluer notre méthode d'extraction d'objets-clés combinée à la méthode de suivi [40], sur des vidéos choisies par eux. Après plusieurs rencontres et résultats qu'elle a jugés fructueux, nous sommes en train de mettre en place les clauses de notre collaboration. De plus, nous avons engagé une procédure de valorisation de nos recherches auprès du consortium GRAVIT[9] qui a débouché sur la création d'un projet d'un an dans lequel je serai chargé du développement d'une plate-forme logicielle utilisant notre approche spatio-temporelle. Cette plate-forme va nous permettre de favoriser les évolutions de notre technique et de fournir un support pour valoriser et évaluer les résultats.

[8] http ://www.idside.fr

[9] GRAVIT (Grenoble Alpes Valorisation et Innovation Technologique) est un consortium qui réunit le CNRS, le CEA, l'INP Grenoble, l'INRIA, l'UJF et l'UPMF dans le cadre d'un programme de mutualisation soutenu par l'Agence Nationale de la Recherche (ANR).

Conclusion

Soignez le commencement, pensez à la fin, la fin viendra sans fatigue. Si vous oubliez le but, vous succomberez avant la fin.
Chou King

Ce document a présenté un premier chapitre permettant d'une part, de définir la notion d'*objet* qui est délicate et très importante et d'autre part, d'introduire les applications susceptibles d'être concernées par une analyse du contenu d'une image ou d'une vidéo. Les chapitres suivants ont traité principalement deux problématiques pour lesquelles nous allons effectuer un bref bilan :

Bilan sur la segmentation fondée sur les objets

Le chapitre 2 présente un état de l'art des méthodes interactives puis automatiques de segmentation d'image fondée sur les objets. Le chapitre 3 présente nos travaux qui ont porté sur de nouvelles utilisations de la pyramide locale et qui ont donné naissance à trois types de traitements fondés sur les objets :

- Un traitement **spatial interactif** : ce traitement se compose de deux outils interactifs qui mettent en valeur les possibilités de la segmentation locale pour l'extraction d'objet dans les images. Ces deux outils permettent des extractions de qualité en demandant un minimum d'interaction avec l'utilisateur. La contribution de nos travaux a été d'adapter la pyramide locale à l'extraction interactive d'objet par segmentation de leurs contours. Notre méthode rivalise avec les méthodes interactives présentées dans l'état de l'art, de par sa simplicité d'utilisation et sa capacité à gérer le problème de camouflage puisqu'elle n'utilise que des informations locales et ne nécessite pas la modélisation globale du fond ou de l'objet. Ainsi un objet de même distribution couleur que le fond peut tout de même être extrait.

- Un traitement **spatial automatique** : l'objectif de nos travaux a été d'automatiser la pyramide locale pour la segmentation d'image en régions d'intérêt. La contribution de nos travaux réside dans la limitation du phénomène de sur-segmentation de la pyramide irrégulière classique en segmentation d'image à l'aide d'une initialisation novatrice de la pyramide locale adaptée à un traitement spatial automatique fondé sur les objets. Ceci a nécessité une méthode de localisation des contours des objets pour initialiser de manière novatrice le procédé de segmentation. Notre technique fournit des partitions de qualité contenant peu de régions et constituant un support intéressant pour les méthodes de recherche d'images dans des bases de données orientées sur le contenu. En effet, ce traitement s'avère fournir une représentation intéressante des objets d'intérêt présents

dans une image bien que la méthode ne puisse pas systématiquement fournir une seule région par objet d'intérêt.
De plus, notre étude sur la fusion de régions orientée sur des critères perceptifs constitue une extension intéressante de la pyramide locale. L'empilement des partitions issues des fusions hiérarchiques fournit une structure permettant à l'utilisateur de récupérer un résultat sémantique.

- Un traitement **spatio-temporel** : nos travaux ont permis d'élaborer une méthode de segmentation spatio-temporelle proposant une extraction *variable* d'objets d'intérêt intra-plan. Les objets d'intérêt sont définis comme appartenant au premier plan et donc animés d'un mouvement différent de celui de la caméra qui filme la scène. Le but de nos travaux a été de détecter les régions du premier plan et de focaliser de manière optimale la segmentation locale sur ces régions pour restituer un masque de l'objet le plus précis possible. Le procédé de focalisation de la segmentation gère les problèmes inhérents à l'estimation de mouvement. La qualité d'extraction, bien que variable, permet d'extraire des régions pertinentes fournissant une représentation de qualité de l'objet d'intérêt filmé.

Bilan sur la représentation de vidéo fondée sur les objets

Après un bref état de l'art des différentes approches concernant la représentation synthétique d'une vidéo (cf. chapitre 4), nous proposons dans le chapitre 5 notre méthode orientée sur la sélection d'*objets-clés*. Ces derniers représentent les occurrences intéressantes des objets d'intérêt filmés et correspondent aux extractions de qualité considérées comme les plus représentatives des objets d'intérêt. Les apports de nos travaux se situent dans :

- La mise en place d'une méthode automatique originale non orientée "suivi".

- La classification des régions extraites qui permet d'obtenir de manière robuste une classe clé par objet d'intérêt détecté. Cette classe clé regroupe toutes les régions extraites concernant le même objet.

- La sélection des objets-clés qui permet une représentation compacte et de qualité du contenu de la vidéo. Chaque objet-clé constitue une référence d'un objet d'intérêt réel.

- La sélection de vues-clés qui permet une description multi-vues des objets d'intérêt détectés dans le plan vidéo. Chaque vue-clé est un masque précis de l'objet et représente un état différent de l'objet d'intérêt : apparence, grossissement, ... L'étude de la population de la classe clé permet également de caractériser les sous-parties mobiles des objets d'intérêt.

- Le développement d'une approche nouvelle dont le but initial est de détecter les objets-clés mais dont les finalités peuvent être multiples dans le cadre de l'interprétation de vidéos : gestion de recouvrement/découvrement, création de résumé, indexation,...

Perspectives

L'état actuel de chaque méthode que nous avons présentée dans ce document laisse envisager un certain nombre d'applications. Cependant, il reste des améliorations à apporter :

- Une première perspective pour chacune de nos méthodes est l'optimisation du temps de traitement. En effet, la pyramide irrégulière est une structure puissante mais complexe d'un point de vue programmation. De ce fait, elle impose d'importants temps de traitement.

- En ce qui concerne la segmentation locale interactive, il serait intéressant de proposer une classification automatique des régions restant non étiquetées après le traitement qui serait par exemple, fondée sur une mesure de régularité du contour de l'objet extrait.
  De plus, l'obtention en différé du résultat constitue l'inconvénient majeur des outils interactifs proposés. Une perspective intéressante est de pouvoir proposer à l'utilisateur un traitement temps réel lui permettant de visualiser en chaque instant le résultat de ses actions. L'objectif est donc de modifier le mode d'interactivité et le fonctionnement de la segmentation locale afin qu'elle puisse fournir un contour optimal entre un point de départ et le point libre dirigé par l'utilisateur.

- Les possibilités de la méthode de segmentation d'image en régions d'intérêt pour l'indexation restent à démontrer. Une perspective est de développer un protocole de recherche d'image fondé sur une telle décomposition de l'image.

- En ce qui concerne l'extraction d'objets-clés, le premier objectif est d'ajouter une classification d'objets inter-plans afin de pouvoir proposer une organisation des objets-clés sur l'ensemble de la vidéo.
  De plus, nous avons vu que l'extraction et la sélection d'objets-clés nécessitent une succession d'étapes. Il nous paraît important d'améliorer en premier lieu ces différents points :

  – La détection des régions du premier plan : pour augmenter la qualité des régions extraites et obtenir une extraction plus robuste pour éventuellement tendre vers une méthode de suivi proprement dite.

  – L'étude de la trajectoire des régions extraites : pour améliorer la classification des régions et gérer avec plus de performance les croisements entre objets. L'estimation de la trajectoire d'un objet nécessite une étude plus approfondie car la nature instable des régions (forme, mouvement, temps d'apparition ...) complique considérablement cette étape.

- Dans l'étape de sélection des vues-clés, nous avons présenté dans la section 5.5.2 une méthode originale pour sélectionner une vue complémentaire de l'objet clé. Cette méthode devra être étendue à une sélection multi-vues dont le nombre pourra, par exemple, être fixé par l'utilisateur afin de compléter la représentation de l'objet d'intérêt.
  La section 5.5.2 présente également une représentation de la composition d'un objet d'intérêt par extraction du sous-élément et/ou du sur-élément de meilleure qualité. Il serait intéressant d'étendre cette technique afin d'extraire tous les sous-éléments et sur-éléments complémentaires entre-eux décrivant un même objet d'intérêt.

Un des points les plus sensibles de la méthode est l'évaluation de la qualité des extractions et des objets-clés sélectionnés. Pour pallier autant que possible ce manque, nous avons tenu à présenter dans ce document de nombreux résultats obtenus sur différentes séquences. Cependant, l'aspect subjectif de la qualité des résultats induit qu'il semble nécessaire d'entreprendre une campagne d'évaluation psycho-visuelle en effectuant une enquête avec plusieurs sujets. La

mise au point d'une telle expérimentation n'est pas aisée mais permettrait de valider le traitement. Cependant, nous nous sommes aperçus que notre technique peut répondre à des besoins réels : à cet effet, nous sommes en relation avec une *startup* française qui désire améliorer son outil de production de publicité interactive : IDside, Paris[10]. Avec ces partenaires, nous avons commencé à évaluer notre méthode d'extraction d'objets-clés combinée à la méthode de suivi [40], sur des vidéos choisies par eux. Les premiers contacts avec cette entreprise date de juillet 2006 et après plusieurs rencontres et résultats qu'elle a jugés fructueux, et devant notre intérêt à valoriser notre technologie, nous sommes en train de mettre en place les clauses de notre collaboration. De plus, nous avons engagé une procédure de valorisation de nos recherches avec GRAVIT[11] auprès de qui nous avons obtenu un financement pour un projet d'un an, prenant effet en 2007. En tant que chercheur contractuel, je serai chargé du développement d'une plate-forme logicielle utilisant notre approche spatio-temporelle. Cette plate-forme va nous permettre de favoriser les évolutions de notre technique et de fournir un support pour valoriser et évaluer les résultats.

L'état actuel de notre méthode d'extraction d'objet-clé laisse envisager un certains nombre d'applications qui nécessitent des masques précis des objets contenus dans une vidéo. En effet, nous avons vu que l'objet-clé fournit une référence de qualité de l'objet d'intérêt réel. Ceci a plusieurs avantages et peut permettre plusieurs applications dont les moins ambitieuses sont :

- **La caractérisation de l'objet d'intérêt** : la connaissance de l'objet-clé permet d'étudier la population d'une classe par comparaison entre chaque membre et la référence représentative de l'objet d'intérêt. Le contenu de la classe peut alors permettre de fournir des informations supplémentaires pour la caractérisation de l'objet d'intérêt : vues complémentaires pour une construction 3D de l'objet, trajectoire, décomposition hiérarchique en sous-parties mobiles, évolution de la composition couleur, temps d'apparition...

- **Le contrôle du suivi d'objet** : les objets-clés et les vues-clés associés à leur masque précis peuvent constituer une initialisation automatique et un moyen de contrôle performant pour une méthode de suivi d'objet par projection de partition tout au long d'un plan. Ils peuvent permettre ainsi d'étudier et de caractériser plus en détails le comportement des objets d'intérêt dans un plan vidéo et ce en chaque image.

- **La structuration de vidéo** : la sélection d'objets-clés intra-plan couplée à une méthode de classification d'objets inter-plan telle que celle proposée par R. Hammoud [46], par exemple, peut permettre la structuration d'une vidéo complète orientée sur les objets qu'elle contient.

- **Construction d'un dictionnaire d'objets-clés** : une classification inter-plans des objets-clés couplée à des modèles d'objets particuliers (personnages, véhicules,...) peut permettre d'organiser les objets-clés sous forme d'un dictionnaire représentatif du contenu de la vidéo.

- **Sélection d'images-clés** : l'étude des occurrences et des interactions entre objets-clés peut permettre une sélection d'images-clés performante et originale.

[10] http ://www.idside.fr

[11] GRAVIT (Grenoble Alpes Valorisation et Innovation Technologique) est un consortium qui réunit le CNRS, le CEA, l'INP Grenoble, l'INRIA, l'UJF et l'UPMF dans le cadre d'un programme de mutualisation soutenu par l'Agence Nationale de la Recherche (ANR).

Bibliographie

[1] Mpeg-7 overview. http ://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm.

[2] P. Meer A. Montanvert and A. Rosenfeld. Hierarchical image analysis using irregular tessellations. In *Technical Reports CS TR 2322 Computer Vision Laboratory, University of Maryland*, September 1989.

[3] E. Adelson and J. Wang. Representing moving images with layers, 1994.

[4] C.-D. Bei and R. M. Gray. An improvement of the mininmum distorsion encoding algorithm for vectors quantization. In *IEEE Transactions on Communications*, volume COM-33(10), pages 1132–1133, 1985.

[5] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18 :509–517, 1975.

[6] P. Bertolino. Contribution des pyramides irrégulières en segmentation d'images multirésolution. In *PhD thesis*, INPG, Grenoble, 30 Novembre 1995.

[7] P. Bertolino, G. Foret, and D. Pellerin. Detecting people in videos for their immersion in a virtual space. In *Proc. of the EURASIP Symposium on Image and Signal Processing and Analysis, ISPA'01*, Pula, Croatia, 2001.

[8] P. Bertolino, R. Mohr, C. Schmid, P. Bouthemy, M. Gelgon, F. Spindler, S. Benayoun, and H. Bernard. Building and using hypervideos. In *IEEE Workshop on Applications of Computer Vision (WACV'98)*, pages 276–277, 1998.

[9] P. Bertolino and A. Montanvert. Multirésolution segmentation using the irregular pyramid. In *IEEE International Conference on Image Processing, ICIP'96*, pages 257–260, Lausanne, Switzerland, 1996.

[10] S. Beucher and F. Meyer. The morphological approach to segmentation : the watersheds transformation in e.r. doughterty (ed.). In *Mathematical Morphology in Image Processing*, pages 433–481, New-York : Dekker, 1992.

[11] J.C. Bezdek. Pattern recognition with fuzzy objective function algorithms. *New York : Plenum Press*, 1981.

[12] G.-A. Bilodeau and R. Bergevin. Evaluation of object recognition algorithms with the image retrieval software plastique. In *ICISP'03*, pages 714–721, 2003.

[13] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. *Lecture Notes in Computer Science*, 3021 :428–441, Janvier 2004.

[14] G. Boccignone, A. Chianese, V. Moscato, and A. Picariello. Foveated shot detection for video segmentation. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 15 (3), pages 365–377, 2005.

[15] G. D. Borshukov, G. Bozdagi, Y. Altunbasak, and A. Murat Tekalp. Motion segmentation by multistage affine classification. *IEEE Transactions on Image Processing*, 6(11) :1591–1594, 1997.

[16] P. Bouthemy and E. François. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *Int. J. Comput. Vision*, 10(2) :157–182, 1993.

[17] Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. *IEEE International Conference on Computer Vision*, 2001.

[18] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *3rd. Intnl. Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, Septembre 2001.

[19] D. Broadbent. Perception and communication. *Pergamon, New York*, 1991.

[20] E. Brunswik and J. Kamiya. Ecological validity of proximity and other gestalt factors. In *American Journal of Psychology*, pages 20–32, 1953.

[21] J. Calic and B. Thomas. Spatial analysis in key-frame extraction using video segmentation. In *Workshop on Image Analysis for Multimedia Interactive Services*, April 2004.

[22] T. Carron and P. Lambert. Color edge detector using jointly hue, saturation and intensity. In *ICIP (3)*, pages 977–981, 1994.

[23] S. H. Cha and S. N. Srihari. On measuring the distance between histograms. *Pattern Recognition*, 35(6) :1355–1370, June 2002.

[24] H. Chang, S.-F. Chen, W. Horace, H.J. Sundaram, and H. Zhong. A fully automated content based video search engine supporting spatio-temporal queries. In *EEE Trans. CSVT*, volume 8 (5), pages 602–615, 1998.

[25] D. Chen and J. Yang. Online learning of region confidences for object tracking. In *The tenth IEEE international Conference on Computer Vision*, Beijing, China, October 15-21, 2005.

[26] Y. Chuang, B. Curless, D. Salesin, and R. Szeliski. A bayesian approach to digital matting. *IEEE Conference of Computer Vision and Pattern Recognition*, 2001.

[27] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 142–149, 2000.

[28] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5), 2003.

[29] J. L Crowley and A.C. Parker. A representation for shape based on peaks and ridges in the difference of low-pass transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2(6) :156–170, 1984.

[30] J. Cutrona and N. Bonnet. Two methods for semi-automatic segmentation based on fuzzy connectedness and watrersheds. In *IASTED International conference on visualization, imaging and image processing VIIP*, 2001.

[31] S. Dasgupta. Learning mixtures of gaussians. In *FOCS '99 : Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, page 634, Washington, DC, USA, 1999. IEEE Computer Society.

[32] Y. Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8) :800–810, 2001.

[33] A. Desolneux, L. Moisan, and J.-M. Morel. *Seeing, Thinking and Knowing*, chapter Gestalt Theory and Computer Vision, pages 71–101. A. Carsetti ed., Kluwer Academic Publishers, 2004.

[34] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1 :269–270, 1959.

[35] J.C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *J. Cybernetics*, pages 32–57, March 1973.

[36] W. H. Press et al. Numerical recipes in c : the art of scientific computing. Cambridge University Press, Cambridge, 1992.

[37] G.D. Finlayson and G.Y. Tian. Color normalization for color object recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 13 (8) :1271–1285, 1999.

[38] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content : The qbic system. *Computer*, 28(9) :23–32, 1995.

[39] F. Fontana, S.G Dellepiane, and G.L. Vernazza. Non linear image labeling for multivalued segmentation. In *IEEE Transactions on circuits and systems for video technology*, pages 429–446, 1996.

[40] G. Foret. *Segmentation spatio-temporelle d'objets vidéo en vue de leur caractérisation.* PhD thesis, EEATS, Octobre 2003.

[41] G. Foret and P. Bertolino. Label prediction and local segmentation for accurate video object tracking. In *SPIE Visual Communications and Image Processing 2003 (VCIP'03)*, Lugano, Switzerland, 8-11 July 2003.

[42] G. Friedland, K. Jantz, T. Lenz, and R. Rojas. Extending the siox algorithm : Alternative clustering methods, sub-pixel accurate object extraction from still images, and generic video segmentation. *Technical Report*, pages 253–259, Janvier 2006.

[43] G. Friedland, K. Jantz, and R. Rojas. Siox : Simple interactive object extraction in still images. *Proceedings of the IEEE International Symposium on Multimedia (ISM2005)*, pages 253–259, 2005.

[44] A. Gresho and R. M. Gray. Vector quantization and signal compression. In *Kluwer Academic Publishers, London, 1992*, 1992.

[45] S. R. Gunn and M. S. Nixon. A robust snake implementation ; a dual active contour. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1) :63–68, 1997.

[46] R. Hammoud. *Construction et présentation des Vidéos Interactives.* PhD thesis, GRAVIR-IMAG et INRIA rhône-Alpes, 2002.

[47] A. Hanjalic. Shot-boundary detection : unraveled and resolved ? *IEEE Transactions on Circuits Systems for Video Technology*, 12(2) :90–105, 2002.

[48] A. Hanjalic and HoongJiang Zhang. An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. *IEEE Transactions on Circuits Systems for Video Technology*, 9(8), 1999.

[49] H. Helmhotz. In *Handbuch der physiologishen optik*, Hambourg and Leipzig, 1866.

[50] J. Heuer, , and A. Kaup. Global motion estimation in image sequences using robust motion vector field segmentation. In *Proceedings ACM Multimedia 99*, pages 261–264, Orlando, Florida, 30 October - 5 November 1999,.

[51] J. Heuer and A. Kaup. Global motion estimation in image sequences using robust motion vector field segmentation. In *MULTIMEDIA '99 : Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 261–264, New York, NY, USA, 1999. ACM Press.

[52] J. Ruiz Hidalgo and P. Salembier. Robust segmentation and representation of foreground key-regions in video sequences. In *Proc. IEEE Internat. Conf. on Acoustic, Speech Signal Process. (ICASSP'01)*, 2001.

[53] C.-T. Hsu and Y.-C. Tsan. Mosaics of video sequences with moving objects. In *ICIP*, Hsinchu, Taïwan, 2001.

[54] K. Idrissi, G. Lavoué, J. Ricard, and A. Baskurt. Object of interest-based visual navigation, retrieval, and semantic content identification system. *Comput. Vis. Image Underst.*, 94(1-3) :271–294, 2004.

[55] B. Marcotegui J. Angulo. Sur l'influence des conditions d'éclairage dans la segmentation morphologique couleur par lpe. In *CORESA*, 2005.

[56] P. Bertolino J. Huart. Similarity-based and perception-based image segmentation. *IEEE Internat. Conf. on Image Processing*, 2005.

[57] A. Jacquot, P. Sturm, and O. Ruch. Adaptive tracking of non-rigid objects based on color histograms and automatic parameter selection. In *IEEE Workshop on Motion and Video Computing*, pages 103–109, Breckenridge, Colorado, January 2005.

[58] A.K. Jain. Fundamentals of digital image processing. Prentice-Hall, NJ, 1989.

[59] F. Jing, M. Li, H. Zhang, and B. Zhang. Unsupervised image segmentation using local homogeneity analysis. In *Proc. IEEE International Symposium on Circuits and Systems*, 2003.

[60] J. M. Jolion and A. Montanvert. The adapted pyramid : a framework for 2d image analysis. In *Computer Vision Graphics and Image Processing*, volume 55 (3), pages 339–348, May 1992.

[61] D. Ju, S. X., Black M. J., Minneman, and S. Kimber. Summarization of video-taped presentations : Automatic analysis of motion and gesture. *IEEE Trans. on Circuits and Systems for Video Technology*, 8(5) :686–696, 1998.

[62] G. Kanizsa. Grammatica del vedere / la grammaire du voir. In *Il Mulino, Bologna / Éditions Diderot, arts et sciences*, 1980 / 1997.

[63] C. Kim and J.-N. Hwang. An integrated scheme for object-based video abstraction. In *MULTIMEDIA '00 : Proceedings of the eighth ACM international conference on Multimedia*, pages 303–311, New York, NY, USA, 2000. ACM Press.

[64] C. Kim and J. N. Hwang. Fast and automatic video object segmentation and tracking for content-based applications. In *IEEE Transactions on Circuits Systems for Video Technology*, volume 12, pages 122–129, Feb. 2002.

[65] W. Kohler. *Gestalt Psychology*. Liveright Publishing Corporation, 1947.

[66] R.L. Lagendijk, A. Hanjalic, M.P. Ceccarelli, M. Soletic, and E.H. Persoon. Visual search in a smash system. In *Proceedings of ICIP'96*, Lausanne, Switzerland, 1996.

[67] C.-H. Lee and L.-H. Chen. A fast motion estimation algorithm based on the block sum pyramid. In *IEEE Transactions on Image Processing*, volume 6(11), November 1997.

[68] S. Lefèvre, J. Holler, and N. Vincent. A review of real-time segmentation of uncompressed video sequences for content-based search and retrieval. 2003.

[69] C.-C. Li, Y. Ming, and W. Jay Kuo. Semantic video content abstraction based on multiple cues. In *IEEE International Conference on Multimedia and Expo (ICME) 2001*, Tokyo, Japan, 2001.

[70] W. Li and E. Salari. Successive elimination algorithm for motion estimation. In *IEEE Transactions on Image Processing*, volume 4(1), pages 105–107, 1995.

[71] X. Li, S.-C. Chen, M.-L. Shyu, and B. Furht. Image retrieval by color, texture and spatial information. 2002.

[72] Y. Li, T. Zhang, and D. Tretter. An overview of video abstraction techniques. *HP*, July 31st 2001.

[73] R. Lienhart. Comparison of automatic shot boudary detection algorithms. *SPIE Conference on Storage and Retrieval for Image and Video Databases*, January 1999.

[74] C.W. Lin, Y.J. Chang, and Y.C. Chen. Hierarchical motion estimation algorithm based on pyramidal successive elimination. In *International Computer Symposium*, 1998.

[75] B. Liu and A. Zaccarin. New fast algorithms for the estimation of block motion vectors. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 3(2), pages 148–157, 1993.

[76] L. Liu and G. Fan. Combined key-frame extraction and object-based segmentation. In *IEEE Trans. Circuits and System for Video Technology*, 2005.

[77] J. Luo and C.-E. Guo. Perceptual grouping of segmented regions in color images. *Pattern Recognition*, 36(12) :2781–2792, 2003.

[78] D. MacAdam. In *Visual sensivities to color differences in daylight*, volume 32, pages 247–274, 1942.

[79] A. Mahboubi, J. Benois-Pineau, and D. Barba. Joint tracking of polygonal and triangulated meshes of objects in moving sequences with time varying content. In *IEEE International conference on image processing*, Thessaloniki, Greece, 2001.

[80] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[81] A. Maßmann, S. Posch, and G. Sagerer. Using markov random fields for perceptual grouping. *International Conference on Image Processing*, 2 :207–210, 1997.

[82] B. Maxwell and S. Shafer. Physics-based segmentation : Moving beyond color. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 742 – 749, June, 1996.

[83] J. Maxwell. On the theory of compound colours and the relations of the colours of the spectrum. In *In Proceedings of the Royal Society of London*, volume 10, pages 404–484, 1860.

[84] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. In *ICCV*, pages 840–845, 1995.

[85] J. McQueen. Some methods for classification and analysis of multivariate obervations. In *5th Berkeley Symposium on mathematics, Statistics and Probability*, volume 1, pages 281–298., 1967.

[86] P. Meer and S. Connelly. A fast parallel method for synthesis of random patterns. In *Pattern Recognition*, volume 22, pages 189–204, September 1989.

[87] M. Mentzelopoulos and A. Psarrou. Key-frame extraction algorithm using entropy difference. In *MIR '04 : Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 39–45, New York, NY, USA, 2004.

[88] F. Meyer and S. Beucher. Morphological segmentation. *Journal of Visual Communication and Image Representation*, 1(1) :21–46, september 1990.

[89] A. Mitiche and P. Bouthemy. Computation and analysis of image motion : a synopsis of current problems and methods. *Int. J. Comput. Vision*, 19(1) :29–55, 1996.

[90] A. Montanvert, P. Meer, and A. Rosenfeld. Hierarchical image analysis using irregular tessellations. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 13(4), pages 307–316, April 1991.

[91] E. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In *ACM SIGGRAPH 95 : Computer Graphics and Interactive Technologies*, pages 191–198, Los Angeles, August 1995.

[92] P. Mulhem, J. Gensel, and H. Martin. Modèles pour résumés adaptatifs de vidéos. In *Revue Ingénierie des Systèmes d'Information, Numéro Spécial Bases des Données et Multimédia*, volume 7 (5-6), pages 91–118, December 2003.

[93] Y. Nagasaka and A. Tanaka. Automatic scene-change detection method for video works. *2nd Working Conference on Visual Database Systems*, pages 119–133, 1991.

[94] S. N. Nayar and R. M. Bolle. Reflectance based object recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 17(3) :219–240, June 1996.

[95] J. Oh, K. A. Hua, and N. Liang. A content-based scene change detection and classification technique using background tracking. In *Proceedings of SPIE : Multimedia Computing and Networking 2000*, pages 254–265, San Jose, CA, Januar 2000.

[96] J.-H Oh, J. Lee, and E. Vemuri. An efficient technique for segmentation of key object(s) from video shots. In *ITCC '03 : Proceedings of the International Conference on Information Technology : Computers and Communications*, page 384, Washington, DC, USA, 2003. IEEE Computer Society.

[97] W. Osberger and A. J. Maeder. Automatic identification of perceptually important regions in an image using a model of the human visual system. In *International Conference on Pattern Recognition*, Brisbane, Australia, 1998.

[98] A. Pardo. Extraction of semantic objetcts from still images. In *International Conference on Image Processing 2002*, New York, USA, September 2002.

[99] G. Pass and R. Zabih. Histogram refienement for content-based image retrieval. In *IEEE Workshop on Applications of Computer Vision*, pages 96–102, 1996.

[100] R. Rickman and J. Stonham. Content-based image retrieval using color tuple histograms. In *SPIE*, pages 2–7, 1996.

[101] A. Rosenfeld. Fuzzy digital topology. *Inform. Control.*, pages 76–87, 1979.

[102] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cuts. *Proceedings of ACM Siggraph Conference*, 2004.

[103] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth moverŠs distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2) :99–121, 2000.

[104] P. Salembier, P. Brigger, J. Casas, and M. Pardàs. Morphological operators for image and video compression. *IEEE Transactions on Image Processing*, 5(6) :881–898, 1996.

[105] P. Salembier and L. Garrido. Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *IEEE Transactions on Image Processing*, 9(4) :561–576, April 2000.

[106] P. Salembier, A. Oliveras, and L. Garrido. Anti-extensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing*, 7(4) :555–570, 1998.

[107] H. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, volume 18, pages 814–830, Août 1996.

[108] Y. Q. Shi and X. Xia. A thresholding multiresolution block-matching algorithm. In *IEEE Transactions on Circuits and Systems for Video Technology*, volume 7(2), pages 344–351, 1995.

[109] J. Smith and S. Chang. Querying by color regions using the visualseek content-based visual query system. In *M.T. Maybury (Ed), Intelligent Multimedia Information Retrieval, AAAI Press*, 1997.

[110] X. Song and G. Fan. Key-frame extraction for object-based video segmentation. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2005)*, Philadelphia, PA, March 2005.

[111] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for realtime tracking. In *Computer Vision and Pattern Recognition*, volume 3, pages 246–252, Fort Collins, USA, 1999.

[112] H. Stokman and T. Gevers. Photometric invariant region detection in multispectral images. In *the 12th Conference on Vision Interface*, pages 90–96, Trois Rivières, Québec, 1999.

[113] M. Striker and A. Diamai. Color indexing with weak spatial constraints. In *SPIE*, pages 29–40, 1996.

[114] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7 (1) :11–32, 1991.

[115] Adobe Systems. Inc : Adobe photoshop user guide. 2002.

[116] Y. Taniguchi, A. Akutsu, and Y. Tonomura. Panorama excerpts : extracting and packing panoramas for video browsing. In *MULTIMEDIA '97 : Proceedings of the fifth ACM international conference on Multimedia*, pages 427–436, New York, NY, USA, 1997. ACM Press.

[117] S. Treetasanatavorn, U. Rauschenbach, J. Heuer, and A. Kaup. Automatics video structuring for multimedia messaging. In *EUSIPCO-2002, XI European Signal Processing Conference*, volume III, pages 467–470, Toulouse, France, September 3-6, 2002.

[118] E. Tuncel and L. Onural. Utilization of the recursive shortest spanning tree algorithm for video-object segmentation by 2-d affine motion modeling. *IEEE Trans. Circuits Syst. Video Techn.*, 10(5) :776–781, 2000.

[119] J. Uchihashi, S. Foote, J. Girgensohn, and A. Boreszk. Video manga : Generating semantically meaningful video summaries. *ACM Multimedia'99*, pages 383–392, 1999.

[120] J.K. Udupa and S. Samarasekera. Fuzzy connectedness and object definition : theory, algorithms and applications in image segmentation. In *GMIP*, pages 246–261, 1996.

[121] N. Vasconcelos and A. Lippman. A spatiotemporal motion model for video summarization. In *CVPR '98 : Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 361, Washington, DC, USA, 1998. IEEE Computer Society.

[122] P. Vasseur, C. Pégard, and M. Mouaddib. Perceptual organization approach based on dempster-shafer theory. *Pattern Recognition*, 32 :1449–1462, 1999.

[123] L. Vincent and P. Soille. Watershed in digital spaces : an efficient algorithm based on immersion simulations. *IEEE Transaction on Pattern Analysis and Machine On Intelligence*, 13(6) :583–598, June 1991.

[124] M. Wertheimer. Principles of perceptual organization. *Readings in perception*, pages 115–135, 1958.

[125] A. Witkin, M. Kass, and D. Terzopoulos. Snakes : Active contours models. *International journal of computer vision*, pages 321–331, 1988.

[126] W. Wolf. Key frame selection by motion analysis. In *Proc. IEEE Internat. Conf. on Acoustic, Speech Signal Process. (ICASSP'96)*, pages 1228–1231, 1996.

[127] K. Y. Wong and M. E. Spetsakis. Motion segmentation by em clustering of good features. In *CVPRW '04 : Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 11*, page 166, Washington, DC, USA, 2004. IEEE Computer Society.

[128] H. Xu, A. A. Younis, and M. R. Kabuka. Automatic moving object extraction for content-based applications. *IEEE Trans. Circuits Syst. Video Techn.*, 14(6) :796–812, 2004.

[129] N. Xu, R. Bansal, and N. Ahuja. Object segmentation using graph cuts based active contours. volume 2, pages 46–53, 2003.

[130] M. Yazdi and A. Zaccarin. Semantic object segmentation of 3d scenes using color and shape compatibility. In *Proc. 6th World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, FL, USA, July, 2002.

[131] H. J. Zhang, J. Wu, D. Zhong, and S. W. Smoliar. An integrated system for content-based video retrieval and browsing. *Pattern Recognition*, 30(4) :643–658, 1997.

[132] Y. Zhang, Y. Rui, T. S. Huang, and S. Metrotra. Adaptive key frame extraction using unsupervise clustering. In *Proceeding of IEEE Int. Conf. on Image Processing (ICIP' 98)*, pages 886–890, October 1998.

[133] D. Zhong and S.F. Chang. An integrated approach for content-based video object segmentation and retrieval. In *IEEE Transactions on Circuits and System for Video Technology*, volume 9 (8), pages 1259–1268, December 1999.

[134] Z. Zivkovic and B. Krose. An em-like algorithm for color-histogram-based object tracking. In *International Conference on Computer Vision and Pattern Recognition*, 2004.

[135] N. Zlatoff, B. Tellez, and A. Bazkurt. Image understanding and scene models : a generic framework integrating domain knowledge and gestalt theory. *International Conference on Image Processing (ICIP)*, pages 2355–2358, 2004.

[136] Nicolas Zlatoff, Guillaume Ryder, Bruno Tellez, and Atilla Baskurt. Content-Based Image Retrieval : on the Way to Object Features. In *International Conference on Pattern Recognition*, september 2006.

Annexes

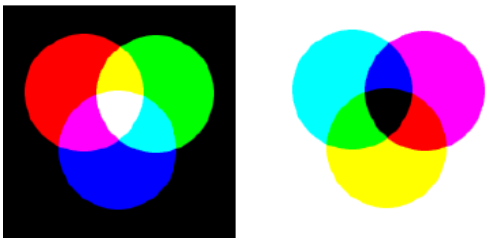
Annexe A

Espaces colorimétriques

Les espaces colorimétriques peuvent être classés selon 4 familles principales : les systèmes primaires, les systèmes perceptuellement uniformes, les systèmes perceptuels fondés sur des attributs caractéristiques de la perception visuelle humaine et enfin les systèmes d'axes indépendants. Chacun trouve son utilité dans les techniques de vision par ordinateur bien que les caractéristiques peuvent être très différentes. Nous verrons que le choix de l'espace couleur doit être fait en accord avec l'application visée.

A.1 Les systèmes primaires

Les systèmes primaires résultent d'une propriété de la perception visuelle humaine de la couleur : l'expérience montre que seules 3 couleurs appelées *primaires* suffisent à reproduire la quasi-totalité des couleurs. Ce phénomène se nomme la *trichromie* (cf. figure A.1). C'est Maxwell [83] qui a véritablement mis au point les principales lois de mélanges fondées sur la trichromie. Ainsi, il est possible de représenter les couleurs dans un espace à 3 dimensions, où chaque composante traduit une couleur primaire.



(a) Synthèse additive (b) Synthèse soustractive

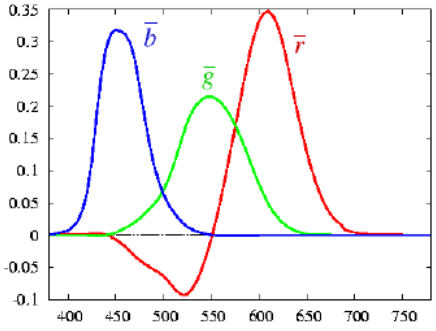
Fig. A.1: Phénomène de trichromie

Ces 3 couleurs peuvent être choisies arbitrairement, si toutefois aucune d'elles ne peut être obtenue par un mélange *ad hoc* des 2 autres. On distingue 2 types de mélanges introduits par Helmhotz [49] : le premier est obtenu par *synthèse additive* (cf. fig. A.1.a) qui utilise le rouge, le vert et le bleu. Le deuxième est obtenu par *synthèse soustractive* (cf. fig. A.1.b) et les couleurs communément utilisées sont le jaune, le cyan et le magenta.

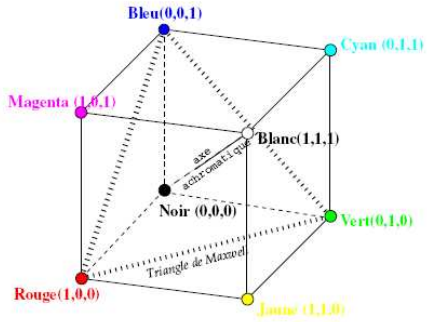
A.1.1 L'espace RVB

En 1931, la CIE[1] a mis en oeuvre un système fondé sur des résultats expérimentaux permettant d'interpréter les sensations colorées selon la synthèse additive. Elles sont fondées sur 3 couleurs primaires : le rouge, le vert et le bleu : C'est l'espace RVB ou RGB[2] en anglais.

Ces expériences ont permis de définir des composantes trichromatiques spectrales illustrées dans la figure A.2.a. Ces composantes $\bar{r}(\lambda)$, $\bar{g}(\lambda)$ et $\bar{b}(\lambda)$ indiquent les quantités de chacune des primaires qui est nécessaire à l'égalisation d'un stimulus monochromatique de longueur d'onde λ. Cette représentation comporte un inconvénient du fait que $\bar{r}(\lambda)$ comporte une partie négative, ce qui est contraire à la théorie de synthèse additive. En effet, il n'est pas possible d'ajouter une lumière négative. Il en résulte l'impossibilité de reconstruire toutes les couleurs à partir du système RGB. C'est ce qu'on appelle le problème de l'égalisation.



(a) Composantes trichromatiques spectrales $\bar{r}(\lambda)$, $\bar{g}(\lambda)$ et $\bar{b}(\lambda)$



(b) Représentation de la couleur dans l'espace RVB

FIG. A.2: Construction de l'espace RVB

L'espace RVB est sans doute le plus communément utilisé en imagerie numérique. Il est également très utilisé dans la vidéo et l'affichage sur les écrans. Cependant en plus du problème d'égalisation, il comporte d'autres inconvénients qui sont :

1. Une forte corrélation entre les 3 composantes
2. Une relation entre l'espace et la perception des couleurs peu intuitive.
3. Une dépendance entre la luminance et les 3 composantes chromatiques puisque elle est définie comme leur somme pondérée : $L = \alpha R + \beta G + \gamma B$. Pour limiter ce phénomène, on utilise généralement les composantes normalisées r, v, b données par les équations suivantes :

$$r = \frac{R}{R+G+B} \quad v = \frac{G}{R+G+B} \quad b = \frac{B}{R+G+B} \tag{A.1}$$

La figure A.2 présente la représentation de la couleur dans l'espace rgb normalisé par rapport à la luminance.

A.1.2 L'espace XYZ

L'espace XYZ correspond à une transformation linéaire de l'espace RGB. Il a été construit pour pallier le problème d'égalisation de l'espace RGB dû à la partie négative de $\bar{r}(\lambda)$. Pour

[1] Comission Internationale de l'Éclairage
[2] *Red Green Blue*

cela, il a été rajouté une quantité de rayonnement rouge au stimulus à égaliser (cf. figure A.3). Ainsi, les composantes n'ont plus de réalité perceptive mais uniquement mathématique. Cependant, grâce à cette transformation le système XYZ permet de reconstruire toutes les couleurs à l'aide de composantes trichromatiques positives. De plus, la luminance a été séparée des composantes chromatiques puisque Y porte seule cette information.

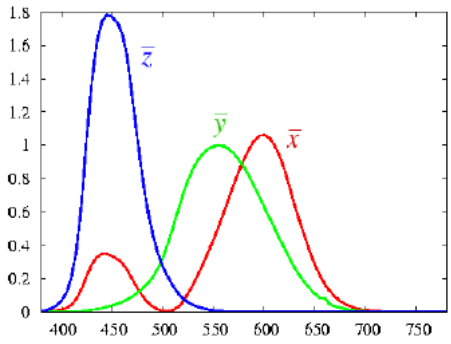


FIG. A.3: Composantes trichromatiques spectrales $\bar{x}(\lambda)$, $\bar{y}(\lambda)$ et $\bar{z}(\lambda)$

A.1.3 L'espace YUV

L'espace YUV est le standard Européen PAL[3], et SECAM[4] de transmission des images couleur pour la télévision. Il correspond à une transformation linéaire de l'espace RGB donnée par :

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ U &= -0.147R - 0.289G + 0.436B \\ V &= 0.615R - 0.515G - 0.100B \end{aligned} \tag{A.2}$$

La composante Y est identique à la composante Y de l'espace XYZ. Les deux composantes de chrominances U et V forment un plan perpendiculaire au plan $R = V = B$.

Tous ces espaces présentent un défaut d'uniformité dans le sens où il n'y pas de linéarité entre la distance existante entre deux couleurs dans le système et l'écart chromatique perçu réellement par un observateur.

A.2 Les espaces perceptuellement uniformes

Les espaces perceptuellement uniformes tentent de conserver une relation entre les distances mesurées quelle que soit leur localisation dans l'espace et les écarts perçus entre les couleurs par un observateur. Le phénomène de non-uniformité a été décrit par les expériences de MacAdam [78]. Pour cela, il a défini dans le diagramme chromatique de la CIE, des ellipses de tailles et d'orientations différentes présentant une uniformité perceptuelle des couleurs (cf. fig. A.4).

Il s'avère que notre oeil est plus sensible dans le vert et permet de distinguer dans cette gamme de couleur de très faibles variations. Au contraire, il se révèle bien moins sensible dans le bleu et le rouge.

[3]Phase Alternative Line
[4]SEquentiel Couleur Avec Mémoire

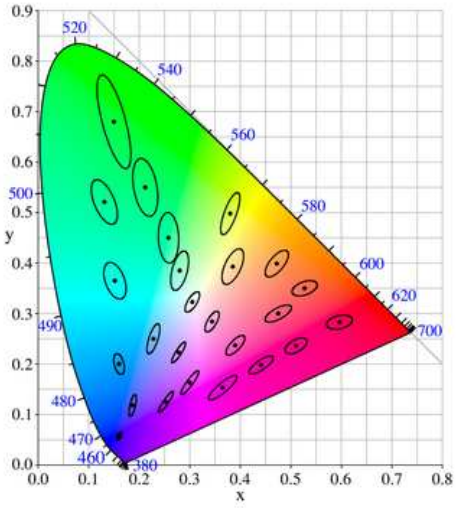


FIG. A.4: Ellipses de MacAdam dans le diagramme chromatique de la CIE

La définition d'un système parfaitement uniforme n'est pas chose aisée et n'a d'ailleurs pas été encore réalisée à ce jour. Cependant, la CIE a proposé deux espaces : le $L^*u^*v^*$ et le $L^*a^*b^*$. Ils sont tous deux issus d'une transformation non-linéaire des systèmes primaires et sont admis comme uniformes.

A.2.1 L'espace L^*u^*v

Le premier espace dit uniforme mis en place par la CIE est le $u^*v^*w^*$. Les composantes u^* et v^* portent l'information de chrominance. La troisième composante w^* représente la luminance. Ce système permet de définir un écart colorimétrique qui n'est autre que la distance Euclidienne. L'inconvénient de cet espace réside dans le fait que les écarts ne sont pas uniformes suivant la luminance. Ce système n'était pas jugé satisfaisant jusqu'en 1973 où son amélioration débouche sur la création du système L^*u^*v fournissant alors l'uniformité au niveau de la luminance.

A.2.2 L'espace $L^*a^*b^*$

Ce système, présenté dans la figure A.5, constitue le standard de référence. Ce qui vaut parfois l'appellation de « vraie couleur » d'une couleur exprimée à l'aide des composantes L^*, a^* et b^*.

Ses composantes sont obtenues des composantes XYZ par la transformation suivante :

$$\begin{aligned} L^* &= 116(Y/Yn)^{1/3} - 16 \text{ pour } \frac{Y}{Y_n} > 0.008856 \\ &\text{ou} \\ L^* &= 903.3(Y/Yn)^{1/3} - 16 \text{ pour } \frac{Y}{Y_n} \leq 0.008856 \\ a^* &= 500 \cdot \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right] \\ b^* &= 200 \cdot \left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right] \end{aligned} \tag{A.3}$$

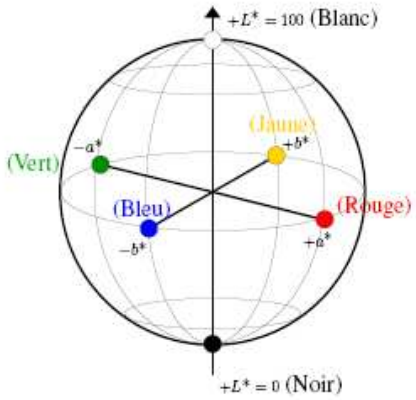
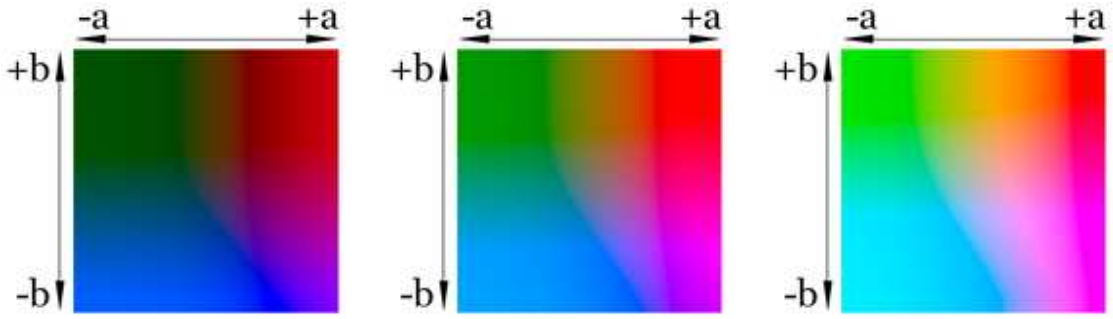


FIG. A.5: Espace uniforme $L^*a^*b^*$

Où $(X_n,\ Y_n,\ Z_n)$ sont les coordonnées du blanc de référence. L'introduction du rapport $\frac{Y}{Y_n}$ permet de simuler très grossièrement l'adaptation de l'oeil à une luminosité donnée. La fonction f est donnée par :

$$f(x) = \begin{cases} \sqrt[3]{x} \text{ si } x > 0.008856 \\ 7.787x + \frac{16}{116} \text{ si } x \leq 0.008856 \end{cases} \tag{A.4}$$

L'espace $L^*a^*b^*$ est basé sur le modèle des couleurs opposées. La luminance s'exprime en pourcentage de 0% pour le noir à 100% pour le blanc. La composante a^* représente l'opposition vert-rouge avec des valeurs comprises entre -100 et $+100$. La composante b^* caractérise l'opposition bleu-jaune à l'aide de valeurs comprises entre -100 à $+100$ (cf. fig. A.6).



(a) 25% de Luminance (b) 50% de Luminance (c) 75% de Luminance

FIG. A.6: Espace uniforme $L^*a^*b^*$

L'inconvénient de cet espace est que les conditions d'acquisition ne sont généralement pas contrôlées. De ce fait les couleurs primaires caractérisant l'image et la référence de blanc caractérisant l'éclairage de la scène, sont généralement inconnues. Ainsi, il est nécessaire de choisir une transformation de l'espace RVB à $L^*a^*b^*$ selon des critères non-calibrés. La littérature montre que la plupart des auteurs choisissent pour toutes les images, la référence D_{65} donnée par la CIE. Malgré cette imprécision, l'apport de cet espace uniforme dans les méthodes de segmentation reste très utile [55]. Cependant, il s'avère que le choix du blanc

de référence traduit par (X_n, Y_n, Z_n) peut avoir des conséquences non négligeables sur les applications de type segmentation [55]. Un choix approprié de l'illuminant peut améliorer de manière conséquente le résultat.

A.3 Les espaces perceptuels

Les espaces perceptuels ont été développés pour être plus intuitifs dans la manipulation des couleurs. A. H. Munsell, peintre et professeur proposa de les ordonner selon une méthode précise. Pour cela, il utilisa les attributs associés à la perception subjective de la couleur définie par Maxwell : la teinte (*Hue*), la luminosité (*value*) et la saturation (*Chroma*). Munsell, classa ces couleurs dans un cylindre selon ces trois critères.

De nombreux systèmes perceptuels ont été développés. Certains d'entre eux expriment en coordonnées polaires des composantes issues de systèmes luminance-chrominance. L'espace $L^*H^*C^*$ et $L^*H^*S^*$ sont respectivement issus des espaces $L^*a^*b^*$ et $L^*u^*v^*$. D'autres sont issus de transformation non-linéaire du système primaire RGB. Il en existe un grand nombre parmi lesquels : le modèle cylindrique ISH[5] et le modèle triangulaire HSL[6]. Le plus utilisé est l'espace HSV[7] (cf. figure A.7). C'est un modèle en cône hexagonal où la teinte est matérialisée par un angle variant de 0^o à 360^o. La saturation S exprime l'éloignement de la couleur par rapport à l'axe achromatique. La luminosité quant à elle, exprimée par la composante V prend la valeur 0 pour le noir et une valeur maximale pour représenter le maximum de clarté que peut atteindre la couleur.

Les composantes de l'espace HSV sont obtenues de l'espace RGB par la transformation suivante :

$$\begin{aligned} H &= arctan\left(\frac{\sqrt{3}(G-B)}{2R-G-B}\right) \\ V &= max(R,G,B) \\ S &= \frac{max(R,G,B)-min(R,G,B)}{max(R,G,B)} \text{ si } max(R,G,B) \neq 0 \text{ sinon } S=0 \end{aligned} \tag{A.5}$$

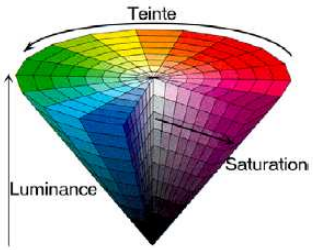
Les systèmes perceptuels présentent l'avantage d'être très proches de la perception visuelle humaine. Ils sont souvent recommandés dans les applications d'indexation [71]. De plus, ils bénéficient de la séparation des valeurs chromatiques et achromatiques. Cependant, ces systèmes ne vérifient pas l'uniformité. De plus, ils comportent quelques singularités. La teinte est par exemple, indéfinie pour des saturations faibles. Une particularité de l'espace HSV est que la saturation constitue un indice de pertinence de la composante de teinte.

A.4 Les espaces d'axes indépendants

Le système de Ohta a été élaboré pour fournir une indépendance des composantes. Il a été estimé à partir d'une décorrélation des composantes obtenue par une transformation de Karhunen-Loeve (KLT), effectuée sur un échantillon de 8 images caractéristiques. Les axes de ce système ont été construits en faisant l'approximation de cette transformation. Les composantes sont données par :

$$I_1 = \frac{R+G+B}{3} \quad I_1 = \tfrac{R-B}{2} \quad I_1 = \frac{2G-R-B}{4} \tag{A.6}$$

[5]Intensity Saturation Hue
[6]Hue Saturation Luminance
[7]Hue Saturation Value



(a)

FIG. A.7: Espace couleur HSV [130]

Remarquons qu'il existe une multitude d'espaces couleur. Nous présentons ici, uniquement les plus utilisés. Nous en profitons pour évoquer la difficulté et la nécessité de soigner le choix de l'espace couleur selon l'application désirée. En effet, il n'existe pas d'espace parfait et chacun présente ses avantages et ses inconvénients.

Annexe B

Segmentation locale par carte d'homogénéité - Résultats

Cette annexe présente les résultats obtenus par segmentation locale initialisée à l'aide de cartes d'homogénéité. Les images tests sont extraites de la base d'images *Corel*. Les résultats ont été obtenus avec un jeu de paramètres identique pour toutes les images. Le nombre de régions par partition est indiqué (en vert) directement sur l'image résultat. La taille des images est de 481×321 pixels ce qui nécessite un temps de traitement d'environ $20s$ par image. Le temps de traitement dépend essentiellement de la taille des zones de segmentation. Plus l'image comporte de zones hétérogènes plus le temps de traitement est important.

La figure B.1 présente les résultats obtenus à l'aide du mélange H_{Rab} utilisant l'espace couleur $L^*a^*b^*$. Tandis que la figure B.2 présente les résultats obtenus à l'aide du mélange H_{TRS} utilisant de l'espace couleur TLS. Les résultats de chaque espace couleur sont divisés en deux groupes. Le premier présente le cas d'images naturelles simples. Tandis que le deuxième présente le comportement des méthodes dans des cas qui sont considérés comme complexes à cause du manque de contraste en luminosité et en couleur des images originales. Il est intéressant de constater la complémentarité des méthodes dans le cas d'images complexes. En effet, l'espace TLS permet d'orienter la segmentation sur la couleur. Tandis que l'utilisation de l'espace $L^*a^*b^*$ permet de prendre davantage en compte l'ensemble des données de l'image.

La figure B.3 présente le comportement de la segmentation en fonction de l'échelle utilisée pour générer les H-images sur chaque composante.

(a) Cas simples

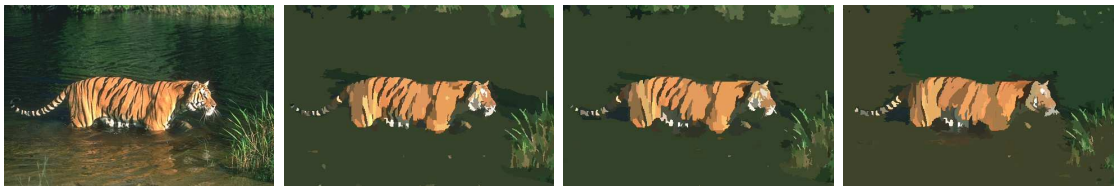
(b) Cas complexes (faibles contrastes luminosité/couleur)

FIG. B.1: Résultats de la segmentation locale initialisée par le mélange H_{Rab}. Le nombre de régions de chaque partition est indiqué en vert

(a) Cas simples

(b) Cas complexes (faibles contrastes luminosité/couleur)

FIG. B.2: Résultats de la segmentation locale initialisée par le mélange H_{TRS}. Le nombre de régions de chaque partition est indiqué en vert



(a) Image originale (b) Echelle 1 (c) Echelle 2 (d) Echelle 3

FIG. B.3: Résultats de segmentation à différentes échelles

Annexe C

Algorithmes rapides du block-matching

C.1 Etat de l'art

Il existe de nombreuses versions rapides de l'algorithme du block-matching (*fast Block-matching algorithm*) mais toutes ne garantissent pas que la correspondance trouvée soit la solution optimale dans l'image entière.

Il est possible de classer les différentes méthodes de *block-matching* rapide en trois catégories. La première catégorie regroupe les techniques qui tentent de diminuer le temps de calcul en restreignant la zone de recherche. Le principal inconvénient est que l'on obtient alors seulement des minima locaux dans cette zone des erreurs de correspondance et non pas des minima globaux. De leur côté, les méthodes de la deuxième catégorie s'efforcent de garder la notion de minimum global et essayent de diminuer le temps de calcul en optimisant le calcul de l'erreur pour chaque position. La troisième catégorie regroupe les techniques combinant les deux approches vues précédemment.

– Categorie I : Les techniques d'ensemble de recherche partielle :

  Les méthodes de cette catégorie procèdent au calcul de la SAD pour un seul ensemble de recherche restreint qui est un sous ensemble de S défini à l'équation 3.30. L'efficacité de ces méthodes repose sur la diminution du nombre de positions de recherche seléctionnées, et sur la façon dont ces positions sont seléctionnées afin d'obtenir le plus petit minimum local de la SAD.

– Categorie II : Les techniques de calcul de l'erreur de correspondance partielle :

  Les techniques de cette catégorie ont pour but d'accélerer le calcul de l'erreur de correspondance pour chaque position de recherche. Au lieu de calculer la SAD complète, on calcul ici une *erreur de correspondance partielle* qui nécessite un temps de calcul moindre mais dont les valeurs sont inférieures voire égales à la SAD.
  Nous pouvons définir la somme partielle des différences absolues (PSAD) de l pixels d'après l'équation 3.29 comme suit :

$$PSAD^{l}_{(x,y)}(u,v) = \sum_{m=0}^{l-1} |I_t(x+i(m), v+j(m)) - I_{t+1}(x+u+i(m), y+v+j(m))| \quad \text{(C.1)}$$

Où $\{(i(m), j(m))|m = 0, ..., B^2 - 1\}$ est l'ensemble des index des pixels appartenant au bloc et B la largeur des blocs en pixels.
Ce dernier définit donc la position et l'ordre des pixels dans le bloc. Il est alors possible de déterminer un sous-ensemble des pixels en parcourant de différentes manières cet ensemble.

Une méthode simple consiste par exemple à sous-échantillonner l'ensemble en prenant en compte un pixel sur quatre, en calculant la $PSAD^{\frac{B^2}{4}}_{(x,y)}(u, v)$ [75]. Bien sûr, cette méthode ne garantit en aucun cas l'obtention du minimum global.

Une autre technique appellée *distance partielle* [4], [44] consiste en un calcul successif des $PSAD^{n}_{(x,y)}(u, v)$ pour chaque position où n prend les valeur de 1 à B^2. Lors du calcul, si l'une des PSAD s'avère être plus grande que le minimum de l'erreur de correspondance calculée précédemment, le calcul pour cette position s'arrête. Par contre, si on atteint le calcul de la $PSAD^{B^2}_{(x,y)}(u, v)$ et qu'elle est encore inférieure à l'erreur minimale déjà calculée, alors l'erreur est mise à jour à cette nouvelle valeur. Toutes les positions de recherche sont ainsi parcourues et les calculs de nombreuses PSAD peuvent être évités. Contrairement à la méthode vue précédemment, celle-ci est susceptible de donner le minimum global.

Une variation de cette méthode, appelée *early-jump out* consiste en l'interruption du calcul des PSAD lorsque l'une d'elles prend une valeur plus élevée qu'un seuil au lieu d'une erreur minimum courante. Cette méthode permet d'économiser encore plus de calculs de PSAD que la méthode de la distance partielle, cependant elle n'est pas en mesure de garantir le minimum global.

Une autre variation de la méthode de la distance partielle définit une nouvelle sorte d'erreur partielle qui est basée sur les projections des valeurs des pixels au lieu des valeurs des pixels eux-mêmes [70] [67]. L'inégalité de Minkowski montre que les projections sont effectivement plus faibles voire égales à l'erreur de correspondance. Ainsi, les projections peuvent être traitées comme une erreur partielle. La méthode consiste donc d'abord à calculer l'erreur complète à la position prédite ce qui constituera la valeur initiale pour l'erreur minimale temporaire. Puis les erreurs partielles sont alors calculées pour chaque position de recherche autre que celle qui a été prédite. Le calcul de l'erreur complète peut être évité à condition qu'au moins une erreur partielle soit plus importante que l'erreur minimale temporaire. Autrement, l'erreur de correspondance complète doit être calculée et l'erreur minimale est alors placée à cette nouvelle valeur si elle est elle-même inférieure à l'erreur minimale temporaire. L'avantage majeur de cette méthode est qu'elle peut fournir le minimum global. Cependant, si la position prédite n'est pas précise, ce qui est le cas par exemple dans une séquence d'images où les objets ont des mouvements changeant rapidement, la valeur de l'erreur minimale initiale risque d'être assez importante. Ainsi, de nombreux calculs inutiles de l'erreur partielle vont être effectués pour finalement calculer l'erreur complète. Dans le pire des cas, le temps de calcul de cette méthode peut arriver à dépasser celui de l'algorithme du full-search.

– Categorie III : Les techniques hybrides :

Récapitulons, les méthodes de la première catégorie réduisent le nombre de positions de recherche en choisissant un ensemble partiel de positions pour optimiser le temps de calcul. Tandis que les méthodes de la deuxième catégorie gagnent du temps en réduisant la fréquence du calcul de l'erreur complète en le substituant le plus souvent possible par

le calcul d'une erreur partielle plus rapide à effectuer. Les techniques hybrides combinent donc ces deux types de méthodes, comme décrit dans [75] [108] pour optimiser l'algorithme. Une autre méthode, qui est la méthode hiérarchique, commence par estimer grossièrement les vecteurs mouvement en basse résolution de l'image puis affine le résultat en haute résolution à l'intérieur d'une petite région de recherche centrée sur le premier résultat grossier.

C.2 Algorithme BSP

L'algorithme BSP [67] (*Block Sum Pyramid Algorithm*) appartient à la deuxième catégorie, il est fondé sur l'algorithme d'élimination successive (SEA) proposé par Li et Salari [70] et introduit en plus une structure de pyramide hiérarchique pour enrichir le SEA de la notion de multirésolution.

Algorithme d'élimination successive

Le critère de mise en correspondance adopté est la minimisation de la SAD, comme pour le FS. Ramenons-nous à une notation plus adaptée à la notion de blocs que l'équation 3.29 pour la SAD : notons SAD(X,Y) la SAD calculée entre les deux blocs X et Y de taille N ; ce qui se formalise de la façon suivante :

$$SAD(X,Y) = \sum_{i=1}^{N}\sum_{j=1}^{N} |X(i,\ j) - Y(i,\ j)| \tag{C.2}$$

Le SEA utilise l'inégalité de Minkowski pour réduire le coût du calcul de l'erreur de mise en correspondance.

$$|(x_1 + x_2) - (y_1 + y_2)| \ \leqslant\ |(x_1 - y_1)| + |(x_2 - y_2)| \tag{C.3}$$

En rapprochant C.2 et C.3, on obtient l'inégalité suivante :

$$SAD(X,Y) \ \geqslant\ |\sum_{i=1}^{N}\sum_{j=1}^{N} X(i,\ j) - \sum_{i=1}^{N}\sum_{j=1}^{N} Y(i,\ j)| \tag{C.4}$$

C'est sur cette inéquation que se base l'algorithme rapide du SEA. Ceci permet de calculer un champ de vecteur trois fois plus rapidement que l'algorithme du Full Search.

Présentation de la pyramide hiérarchique

Le BSPA étend cette inéquation en introduisant la notion de pyramide à multirésolution. En fait, chaque bloc va être représenté par une pyramide où chaque pixel du niveau m va être le résultat de la somme de ses 2 x 2 pixels voisins du niveau $m-1$ comme le montre la figure C.1. Ceci s'exprime de la façon suivante :

$$X^m(i,j) = X^{m-1}(2i-1,2j-1) + X^{m-1}(2i-1,2j) + X^{m-1}(2i,2j-1) + X^{m-1}(2i,2j) \tag{C.5}$$

Pour des blocs N x N (N=2M), la SAD du niveau m est définie par :

$$SAD^m(X,Y) = \sum_{i=1}^{2^{M-m}}\sum_{j=1}^{2^{M-m}} |X^m(i,\ j) - Y^m(i,\ j)| \tag{C.6}$$

Nous obtenons alors l'inégalité de Minkowski avec la notion de multirésolution :

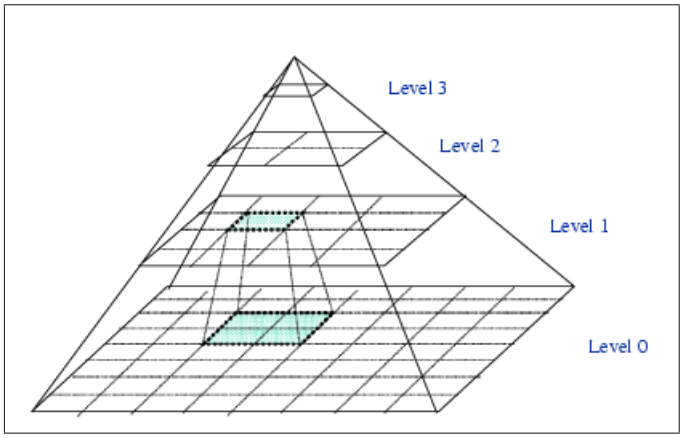


FIG. C.1: Construction de la pyramide hiérarchique (d'après [74])

$$SAD(X,Y) = SAD^0(X,Y) \geqslant SAD^1(X,Y) \geqslant ... \geqslant SAD^{n-1}(X,Y) \geqslant SAD^n(X,Y) \tag{C.7}$$

L'algorithme couplé à la structure de la pyramide le rend plus rapide que le full-search et fournit les mêmes résultats que ce dernier. Nous pouvons résumer les différentes étapes du BSPA comme cela :

1. Initialisation de la valeur SAD courante en considérant un vecteur nul pour le bloc traité.
2. Construction des pyramides des sommes des blocs pour chaque candidat dans la zone de recherche dans l'image courante.
3. Construction de la pyramide du bloc courant.
4. Comparaison pour un bloc candidat des SAD^m hiérarchiques avec la SAD courante.
   - Si la SAD^m calculée est supérieure à la SAD courante, on élimine le candidat et on passe à l'étape 5.
   - Si la SAD^m calculée est inférieure à la SAD courante, on descend d'un niveau dans la pyramide ($m := m - 1$) et on réitère l'étape 4, jusqu'à la base de la pyramide. Si on atteint le niveau 0, on remplace la SAD courante par la SAD calculée et ce bloc devient le correspondant.
5. Retour à l'étape 4 jusqu'à ce que tous les blocs candidats aient été parcourus.

Annexe D

Extraction automatique de régions d'intérêt

Nous n'avouons de petits défauts que pour persuader que nous n'en avons pas de grands. La Rochefoucauld

Cette annexe présente les résultats obtenus avec la méthode d'extraction automatique de ROI présentée au paragraphe 3.4. Le principe est d'initialiser automatiquement la segmentation locale par une étude des régions du premier plan. Ces résultats ont été obtenus avec un même jeu de paramètres concernant la segmentation et la création des masques de segmentation. Les 3 séquences utilisées sont pour les plus classiques *Foreman* et *Vectra* auxquelles se rajoute une séquence issue d'un court métrage d'animation : *Elephant's Dream*[1]. Nous rappelons que cet échantillon de résultats tentent de rendre compte du comportement de la méthode d'extraction. Il n'est pas question ici de proposer un florilège des meilleurs résultats obtenus dans ces séquences mais plutôt de confronter les avantages et les inconvénients. Les défauts d'une extraction de ROI peuvent être classés en deux catégories :

Les faux négatifs : fuite du fond vers la ROI

Les faux positifs : fuite de la ROI vers le fond

Quant aux faux négatifs, ils sont directement liés au modèle de l'objet fondé sur le mouvement apparent. En effet, même si la méthode comporte un aspect de reconstruction de l'objet, la qualité du résultat et le recouvrement de l'objet d'intérêt par la ROI dépend étroitement du mouvement apparent de ce dernier. La $7i^{\text{ème}}$ image du plan-séquence *Foreman* illustre par exemple ce problème. Nous pouvons voir que seule une partie de l'objet est présente dans la ROI (Remarquons comme la ROI floue présente un complémentaire intéressant à la ROI brute). De plus, dans le plan issu de Elephant's Dream, nous pouvons constater que le personnage au deuxième plan n'est pas détecté tout de suite puisqu'il n'a pratiquement pas de mouvement apparent. Ce modèle implique une certaine instabilité de la qualité d'extraction au cours d'une séquence.

Un problème supplémentaire qui n'est pas illustré dans ces exemples est la détection de région du premier plan parasite de grande taille. En effet, ce type de régions peut apparaître à cause d'erreurs commises lors de l'estimation de mouvement. Si leur taille est importante elles peuvent générer une racine et appartenir au résultat final.

Remarquons que la prise en compte des régions non étiquetées dans les ROIs permet parfois de récupérer des zones intéressantes appartenant effectivement à l'objet d'intérêt. Ces

[1] Court métrage d'animation *open source* réalisé par le studio Orange et distribué sous licence « *Creative Commons* »

dernières n'ont pu être détectées comme faisant partie de premier plan à cause de leur faible mouvement ou de leur homogénéité. C'est le cas dans le plan séquence *Foreman* où la ROI floue comporte le casque du personnage qui est en fait une région très homogène.

FIG. D.1: Résultats de l'extraction de ROIs sur quelques images non successives de plusieurs plans. La figure présente de haut en bas : l'image originale, la ROI brute et la ROI floue

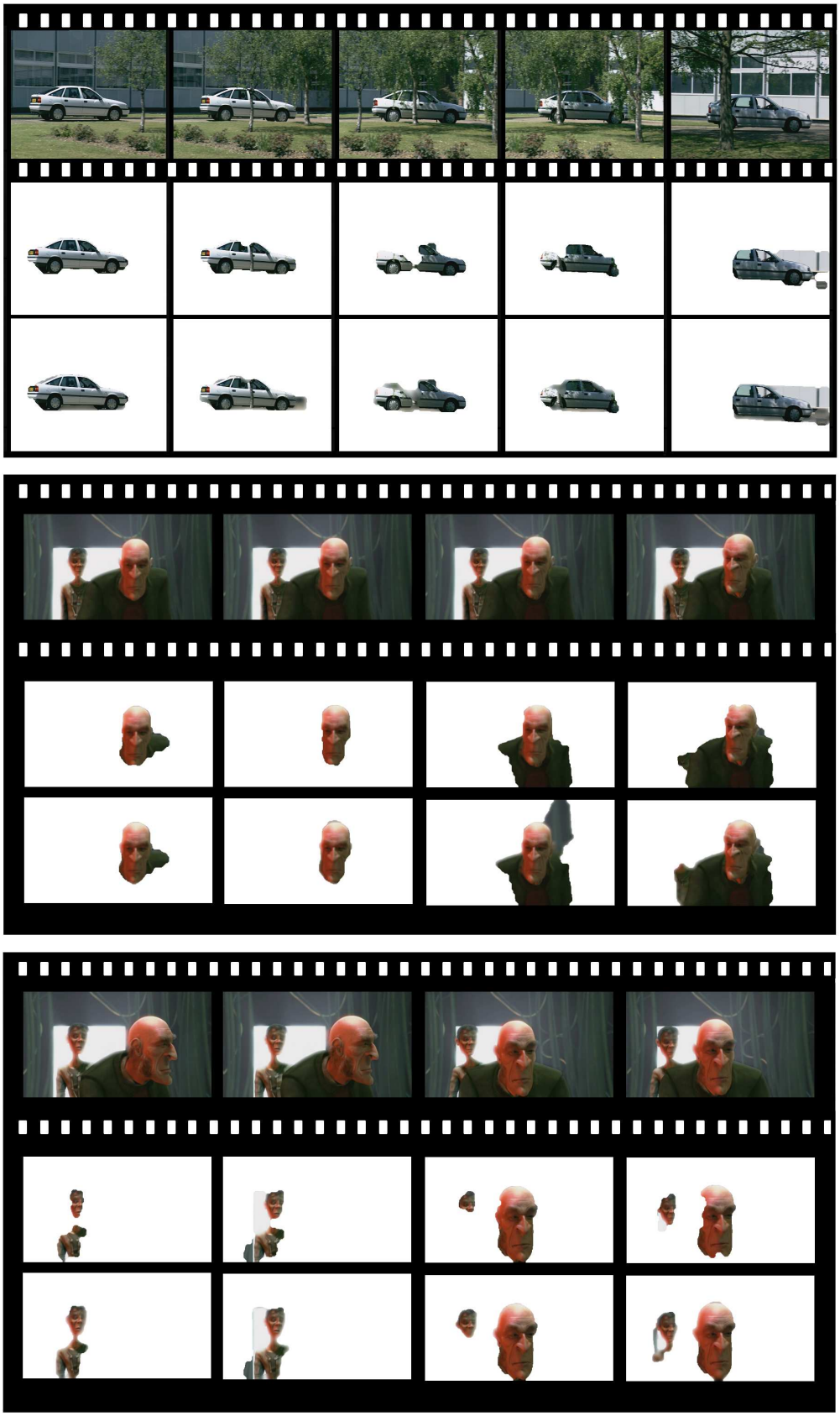
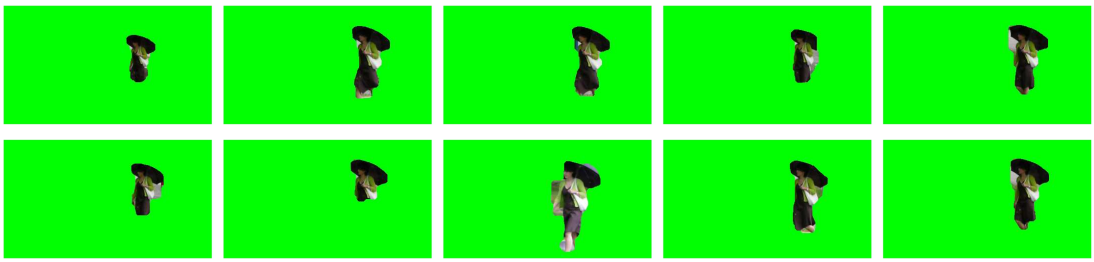


FIG. D.2: Résultats de l'extraction de ROIs sur quelques images non successives de plusieurs plans. La figure présente de haut en bas : l'image originale, la ROI brute et la ROI floue

Annexe E

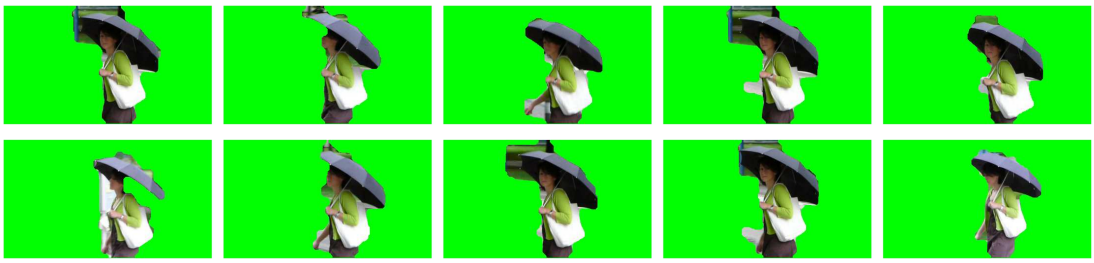
Classification de S-VOPs dans le cas d'un zoom

(a) Aperçu du plan-séquence ($\sim$ 300 images)



(b) Classe aire faible

(c) Classe aire moyenne



(d) Classe aire élevée

FIG. E.1: Exemple de la constitution d'une classe-clé dans le cas d'un zoom sur l'objet d'intérêt : chaque sous-ensemble de $\widehat{C}$ est pertinent

Annexe F

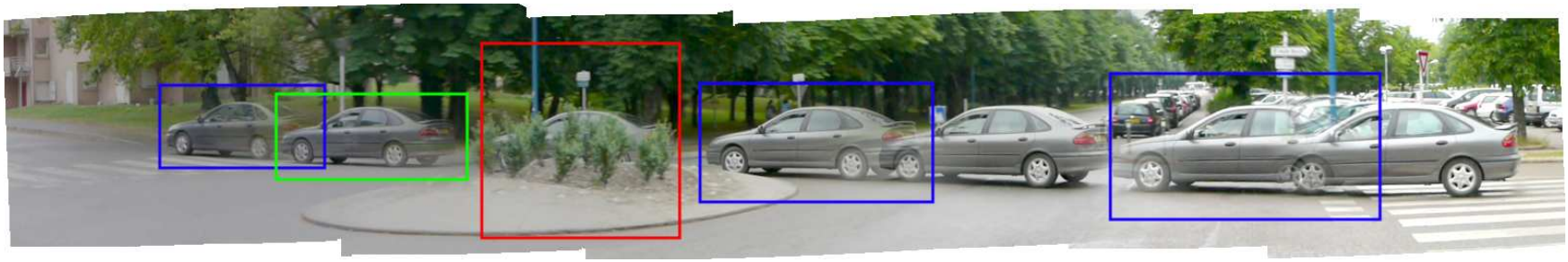
Contrôle de suivi d'objet

(a) Suivi avant de l'objet-clé. Sens chronologique : gauche à droite

(b) Suivi arrière de l'objet-clé. Sens chronologique : droite à gauche

(c) Suivi arrière corrigé

FIG. F.1: Correction du suivi à l'aide de S-VOPs de contrôle : le suivi, initialisé par l'objet-clé (en vert), est effectué de gauche à droite. Les vues-clés (en bleu) permettent la mise à jour du suivi



(a) Aperçu de la séquence vidéo avec marquages : Vert = objet-clé, bleu = S-VOPs de contrôle, rouge = zone d'occultation

(b) Résultat du suivi contrôlé

FIG. F.2: Gestion des occultations dans le suivi de l'objet-clé à l'aide de S-VOPs de contrôle : le suivi initialisé par l'objet-clé (en vert) est effectué de part et d'autre de celui-ci. Les vues-clés (en bleu) permettent la mise à jour du suivi et la récupération de l'objet après la zone d'occultation (en rouge)