Segmentation morphologique interactive pour la fouille de séquences vidéo

Jonathan Weber

Laboratoire des Sciences de l'Image,
de l'Informatique et de la Télédétection
UMR 7005 UDS/CNRS

Numéro d'Ordre
1194

École Doctorale Mathématiques, Sciences de l'Information et de l'Ingénieur

Thèse

présentée pour l'obtention du grade de

Docteur de l'Université de Strasbourg
Discipline : Informatique

par

Jonathan Weber

Segmentation morphologique interactive pour la fouille de séquences vidéo

Soutenue publiquement le 30 Septembre 2011

Membres du jury

Président, Rapporteur :	Olivier Lézoray, Professeur, Université de Caen Basse Normandie
Rapporteur :	Florence Sèdes, Professeur, Université Paul Sabatier Toulouse
Examinateur :	Erchan Aptoula, Professeur-Assistant, Okan Üniversitesi, Turquie
Directeur de thèse :	Pierre Gançarski, Professeur, Université de Strasbourg
Directeur de thèse :	Sébastien Lefèvre, Professeur, Université de Bretagne Sud
Invité :	Christian Dhinaut, Chef de produit, Ready Business System

À Yvan,
qui m'a toujours encouragé
mais est parti avant la fin

Remerciements

Une thèse est par essence un travail personnel. Cependant, sans un environnement propice et des soutiens constants, elle est un but inatteignable. Ainsi, j'aimerais remercier ceux sans qui ce travail n'aurait été possible.

À tout seigneur, tout honneur, mes directeurs, Sébastien Lefèvre et Pierre Gançarski. J'ai travaillé avec Sébastien depuis la Maîtrise jusqu'à la fin de ma thèse. Il a guidé mes premiers pas dans le monde de la Recherche et su me rattrapper lorsque je trébuchais. Il a su me convaincre, dès fois contre vents et marées, que la finalité de ce travail était possible. Nos nombreuses discussions hors-sujet m'ont beaucoup appris. J'ai également beaucoup apprécié de travailler avec Pierre, dans une ambiance toujours décontractée mais néanmoins studieuse, le tout teinté de son humour si particulier.

Les membres de mon jury. En premier lieu, Olivier Lézoray et Florence Sédès qui ont accepté de rapporter mes travaux. Leurs remarques m'ont permis de mieux mettre en perspective mon travail. Je remercie également Olivier Lézoray pour avoir présidé mon jury de soutenance. Mes remerciements vont également à Erchan Aptoula qui n'a pas hésité à traverser toute l'Europe pour assister à ma soutenance. Enfin, je remercie Christian Dhinaut qui fut mon responsable à RBS et qui a su m'apprendre, lors de nos échanges, bien plus de choses qu'il ne l'imagine.

L'ANRT et la société RBS qui ont financé ces travaux.

Germain Forestier qui, bien plus qu'un collègue fut un réel compagnon de route sur les chemins sinueux de la thèse et de l'après-thèse. Nous avons partagé de nombreuses épreuves et su nous remotiver mutuellement plus d'une fois. Thanks bro.

Camille Kurtz et François Petitjean, notre cohabitation lors de ma dernière année m'a permis de travailler dans une très bonne ambiance que j'aurais du mal à retrouver. Vous allez me manquer les loulous.

Laurence Longo, collègue thésarde de RBS qui a également vécu le parcours schizophrène d'une thèse CIFRE, tiraillé entre labo et entreprise.

Aurélie Bertaux, co-bureau des premières années quand nous régnions sans partage sur la C325.

Les thésards et autres membres de l'équipe, Ogier, Emmanuel, Olivier, Alexandre, Sébastien, Sophie et Mathieu, nos discussions de midi et les rires qu'elles entrainaient m'ont permis de travailler avec le sourire.

Mes collègues de l'IUT. Cédric Wemmert, l'inclassable toujours présent quand on en a besoin. Ne change pas. Agnès Braud et Nicolas Lachiche qui m'ont initié aux joies (et aux peines) de l'enseignement et qui avaient toujours un peu de temps pour moi. Franco Zaroli, compagnon des discussions matinales de 7h. Julien Haristoy, le chef, pour nos longues discussions (parfois philosophiques) dans son bureau. Sandrine Heitz, première fan de mes imitations de Maïté. Nos rôles de bon flic/méchant flic lors des jurys me manqueront. Sans oublier, Christiane, Jean, Francine, Véronique, Eliane, Raymond, Bruno, Pascal et les autres.

Mes collègues du LSIIT. Benoit Naegel, pour ses conseils et ses remontages de moral. Nicolas Passat, l'ami du café. Les gens du réseau Julien Montavont, Pascal Mérindol, Julien Beaudaux, Antoine Gallais et leur comparse du parallélisme Julien Gossa pour quelques bonnes tranches de fou rire toujours politiquement correct. Sans oublier, Benjamin, Régis, Manuel, Thomas, Simon, Benoit, Stéphane, Sylvie et les autres.

Mes collègues de RBS, Pascal, Damien, Antoine, Jessica et les autres.

Ma famille et plus particulièrement mes parents qui m'ont toujours laissé choisir ma voie et mon grand-père qui a su me montrer l'importance des études et de la science. Sans oublier ma belle famille et nos vacances au ski, rien de telle qu'une descente pour se remettre d'aplomb.

Mes amis de la vraie vie qui m'ont permis de me ressourcer le temps d'une bière, d'une soirée ou d'un week-end. Dans le désordre, Alexandre, Sébastien, Gilles, Jérôme, Grégory, Céline, Wanda, Michael, Tina, Flora, Julianne, Sylvain, Adrien, Jocelyn, Yann, Elsa, Yannick, Corinne, Cédric, Hui, Christophe, Caroline, Arnaud, Régis, les Guillaume(s), les Florence(s), Julien, Claire, Mike, Benoit, les Émilie(s), Paul, Monia et mes filleules Coraline et Margot.

Last but not least, je remercie celle qui partage ma vie, Laurence qui, en plus de soutenir un thèsard a du (et su) supporter un thésard. Sans son soutien quotidien, je n'aurais pu finir. Promis, maintenant le mot « week-end » aura pour nous la même signification que pour les autres gens et je rentrerais à la maison à des heures décentes.

Table des matières

Notations et sigles

Notations

E	Espace de définition des images ($E \subset \mathbb{N}^2$ pour les images 2D, $E \subset \mathbb{N}^3$ pour les vidéos)
p	Pixel de l'image ($p = (x, y)$ pour les images 2D, $p = (x, y, t)$ pour les vidéos)
E_f	Espace de définition de l'image f
V	Espace de valeurs d'une image ($V = \{vrai, faux\}$ pour les images binaires, $V \subset \mathbb{N}$ pour les images en niveaux de gris, $V \subset \mathbb{N}^n$ pour les images à n bandes)
f	Image (ou séquence vidéo) décrite par une fonction qui à tout pixel $p \in E$ associe une valeur $v \in V$, soit $f : E \to V$, $p \to f(p)$
f_t	$t^{\text{ième}}$ trame d'une séquence vidéo f
$f(p)$	valeur du pixel p de l'image f
$f^R(p)$	valeur de la composante rouge du pixel p de l'image f
$f^V(p)$	valeur de la composante verte du pixel p de l'image f
$f^B(p)$	valeur de la composante bleue du pixel p de l'image f
N_k	voisinage, k indique le type de voisinage (par exemple N_8 représente le 8-voisinage)
X	dimension spatiale représentant l'abcisse
Y	dimension spatiale représentant l'ordonnée
T	dimension temporelle
$\mathbf{x}$	le vecteur x

Sigles

LPE	Ligne de Partage des Eaux
LPEGM	Ligne de Partage des Eaux Guidée par Marqueurs
OV	Objet-Vidéo
SFV	Système de Fouille Vidéo
SFVOO	Système de Fouille Vidéo Orientée Objet
SRG	Seeded Region Growing
ZQP	Zone Quasi-Plate
ZQPGM	Zone Quasi-Plate Guidée par Marqueurs

Introduction

Sommaire

Contexte

Après l'augmentation massive des données textuelles, et plus récemment des images, disponibles dans des bases de données et sur Internet, nous observons aujourd'hui une augmentation importante du volume de données vidéo disponibles. La vidéo est en train de devenir une des principales sources d'informations (YouTube [1] délivre plus de 100 millions de séquences vidéo chaque jour sur Internet) et tend à être de plus en plus omniprésente (Cisco prévoit qu'en 2012, la vidéo représentera 50% du trafic Internet et qu'en 2015, 100 millions de minutes de vidéo seront diffusées sur Internet chaque seconde [CIS11]). Extraire de l'information afin d'utiliser (par ex. fouille) ou d'explorer (par ex. navigation) cette importante masse de données nécessite des méthodes rapides et efficaces. Or, à l'issue du processus d'acquisition, cette masse d'informations est le plus souvent non structurée. En général, les seuls indicateurs disponibles concernent la date et l'heure de l'acquisition, la taille de l'image ou la durée de la séquence vidéo, ou encore le numéro d'ordre du fichier. Ces indices ne reflètent aucunement le contenu des données et ne sont très souvent pas suffisamment pertinents pour permettre une consultation efficace de la part d'un utilisateur. Dans le cas d'un corpus restreint, une structuration manuelle des données est envisageable mais ce choix n'est plus pertinent dès lors que le nombre d'éléments considérés augmente. Or, une structuration automatique de telles bases se heurte au *fossé sémantique* existant entre l'utilisateur de vidéo et les séquences vidéo en elles-mêmes, c'est-à-dire la différence entre la représentation numérique d'une séquence vidéo par des informations brutes (par exemple, les valeurs des pixels) et l'interprétation de son contenu par un être humain. En effet, instinctivement, lorsque nous décrivons une image, nous parlons des objets présents dans cette image (par exemple, « *Il y a deux voitures devant une maison dans cette image* »). La description de séquence vidéo est basée sur le même principe avec éventuellement une évolution temporelle ou une description de l'action (par exemple, « *Il y a deux voitures qui roulent devant une maison dans cette vidéo* »). Cet exemple met en évidence l'existence d'un fossé sémantique (voire gouffre) entre une telle phrase et l'information brute contenue dans les séquences vidéo. Ce fossé sémantique s'est encore élargi par rapport aux images fixes en raison de l'évolution temporelle des objets réels qui peut provoquer des divisions, des fusions, des occlusions et des déplacements de la représentation visuelle de l'objet dans la séquence. Si une voiture en mouvement disparaît derrière une maison et réapparaît par la suite, il s'agit toujours du même objet, même si ses deux occurrences dans la vidéo ne sont ni spatialement ni temporellement connexes. Ainsi, le fossé sémantique n'est pas uniquement présent dans la perception de l'objet mais aussi dans la compréhension de sa place et de son évolution dans son environnement spatio-temporel, en raison des actions relatives des objets. Ainsi, dans les séquences vidéo, on peut décrire les action d'une voiture (par exemple, *avancer* ou *reculer*) mais aussi les conséquences de ces

1. YouTube, http ://www.youtube.com/

actions (par exemple, *s'éloigner de*). Combler le fossé sémantique est un défi majeur pour la fouille vidéo. En contrepartie, l'information sémantique portée par ces objets permet, à notre avis, de mettre en place des algorithmes/méthodes de fouille vidéo plus efficaces et pertinents. Cependant, cette fouille vidéo dite orientée-objet nécessite l'extraction préalable des objets. Cette extraction est généralement effectuée par la segmentation en région des séquences vidéo. Une *région* est un ensemble connexe de pixels ayant des propriétés similaires (par exemple, des couleurs similaires) dans une image ou une vidéo alors qu'un *objet* est une région ou un ensemble de régions ayant une signification sémantique donnée en général par un ensemble de descripteurs portant cette information de haut-niveau. Ainsi, si les régions obtenues par le processus de segmentation ne peuvent être associées à aucune signification sémantique, le processus de fouille ne peut s'effectuer efficacement. Nous pensons que le fossé sémantique ne peut donc être comblé qu'en s'appuyant sur une segmentation en objets spatio-temporels de qualité ainsi que sur des descriptions de ces objets sémantiquement discriminantes. Ainsi, les systèmes de fouille vidéo orientée-objet (SFVOO) sont complexes à mettre en oeuvre, mais représentent pour nous l'avenir de la fouille vidéo de par la sémantique apportée par les éléments qu'ils manipulent.

L'objectif de cette thèse est donc de proposer une approche nouvelle pour la compréhension de séquences vidéo basée sur cette notion d'objet intégrant le temps comme caractéristique de ceux-ci. Cette thèse a été effectuée dans le cadre d'une convention CIFRE avec la société RBS, intégrateur et éditeur de logiciels basé à Entzheim. RBS souhaite intégrer des solutions vidéo aux différentes plateformes logicielles qu'elle développe.

De la notion d'objet

Avant de présenter en détail l'objectif et les contributions de cette thèse, il est important de préciser la notion d'objet, notion centrale dans ce manuscrit.

Le terme objet recouvre plusieurs acceptions, que ce soit dans la vie courante ou dans le domaine informatique. Afin d'éviter les confusions, nous définissons, dans cette thèse, trois acceptions :

- **objet réel** : *toute chose concrète perceptible par la vue ou le toucher*, selon la définition communément admise. Il s'agit des choses qui nous entourent et dont nous avons une perception sémantique, c'est-à-dire que nous sommes capables de nommer, qu'elles soient conçues par l'Homme (objet manufacturé, objet d'art, etc.), vivantes (animal, arbre, etc.) ou naturelles (montagne, océan, etc.). Cette définition peut également recouvrir des concepts de plus haut niveau qui résultent de l'assemblage de plusieurs objets réels, comme une « foule » qui représente une *multitude de personnes réunies en un même lieu*.
- **objet-vidéo** : représentation $2D + t$ d'un objet réel sous la forme d'une suite de représentations $2D$ (séquence d'images). Il s'agit de la représentation informatique d'un objet réel dans une séquence vidéo classique ($2D$ par opposition aux séquences vidéo tri-dimensionnelles en plein essor). Un objet-vidéo consiste donc en un ensemble spatio-temporel de pixels associé sémantiquement à un *objet réel*.
- **objet d'intérêt** : *objet réel* (ou sa représentation par un *objet-vidéo*) qui intéresse un utilisateur ou un système. Il s'agit d'une notion subjective qui dépend de ce que recherche l'utilisateur ou le système. Cette dénomination permet de distinguer les *objets* (réels ou vidéo) pris en considération dans l'analyse par rapport aux autres : tout *objet* (réel ou vidéo) qui n'est pas un objet d'intérêt est considéré comme appartenant au fond.

Objectif

Comme introduit précédemment, l'objectif de cette thèse est d'étudier et proposer les méthodes et mécanismes nécessaires à la fouille de séquences vidéo orientée objet. Cette thèse se décompose de fait en deux parties :

1. Extraction des objets-vidéo (par segmentation) ;
2. Caractérisation et fouille de ces objets-vidéo.

Cet objectif se heurte à deux verrous : un *scientifique* qui concerne le *fossé sémantique* qui sépare les données vidéo de leur interprétation ; un autre *technologique* qui concerne le *temps de calcul* nécessaire pour le traitement des données vidéo.

Dans cette thèse nous montrerons que le *fossé sémantique* peut être comblé par l'implication de l'utilisateur dans le processus d'extraction des objets-vidéo ainsi que dans le processus de fouille. De plus, cette approche permet de personnaliser la structuration (organisation permettant de simplifier leur traitement) par rapport à un utilisateur et donc de correspondre à ses besoins spécifiques.

Nous limiterons le *temps de calcul* en proposant deux méthodes simples et peu coûteuses d'un point de vue calculatoire. Nous verrons également comment appliquer les traitements impliquant l'utilisateur sur des données réduites afin d'en accélérer le traitement et donc d'en améliorer l'interactivité.

Contributions et plan

Les contributions principales de cette thèse sont : une approche de segmentation de séquence vidéo en « pièces de puzzle » spatio-temporelles, une méthode d'assemblage de ces pièces guidée par l'utilisateur afin d'obtenir les objets-video d'intérêt et un cadre générique utilisant les objets-vidéo obtenus comme éléments d'un processus de fouille de données impliquant l'utilisateur.

L'approche de segmentation de séquences vidéo en « pièces de puzzle » spatio-temporelles repose sur l'extension des zones quasi-plates aux données vidéo. Plutôt que d'étendre directement les zones quasi-plates aux séquences vidéo en traitant ces dernières comme des volumes tri-dimensionels, nous avons choisi de traiter séquentiellement les dimensions spatiales et temporelles. Cette application séquentielle est en réalité un cas particulier d'un cadre plus générique d'utilisation des zones quasi-plates que nous introduisons également ici. Cette approche donne de bons résultats qui, couplés à une procédure de filtrage adaptée, permettent une réduction importante des données. Au vu de l'importance du temps de calcul dans le traitement de la vidéo, nous avons également proposé un algorithme efficace pour la construction de zones quasi-plates et étudié certaines améliorations possibles du temps de calcul. Le chapitre 1 traite de ce point. Après une étude des approches existantes pour les images, nous y proposons le cadre générique pour les zones quasi-plates que nous appliquons aux séquences vidéo.

La méthode d'assemblage des zones quasi-plates spatio-temporelles repose sur une interaction simple avec l'utilisateur qui ne fait que marquer les objets d'intérêt. Nous utilisons ces marqueurs pour guider la segmentation des objets d'intérêt par fusion de zones quasi-plates. La fusion étant effectuée sur la réduction de données que représente les zones quasi-plates, elle est rapide et permet une réelle interaction avec l'utilisateur. En outre, la méthode que nous proposons est capable de corriger les zones quasi-plates dans le cas où elles auraient été remises en cause par les marqueurs. Le chapitre 2 concerne l'implication de l'utilisateur dans ce processus d'extraction d'objet-vidéo. Nous présentons quelques approches existantes et proposons notre méthode de segmentation vidéo interactive à faible coût calculatoire.

Le cadre générique (VOMF) que nous proposons pour la fouille vidéo orientée-objet repose sur une implication de l'utilisateur à tous les niveaux. Les objets-vidéo sont porteurs d'une signification sémantique qu'il n'est pas possible d'interpréter automatiquement. C'est pourquoi, nous incluons l'utilisateur à la fois dans le processus d'extraction des objets-vidéo et dans l'étape de fouille de ces objets. L'utilisation de la méthode d'assemblage des zones quasi-plates proposée pour la segmentation des objets-vidéo permet de minimiser le temps utilisateur nécessaire. Pour l'étape de fouille le temps utilisateur est aussi minimisé par l'utilisation de contraintes simples sur les données

permettant de guider efficacement la tâche de fouille. Le chapitre 3 est centré sur ces problématiques. Nous y étudions les approches existantes et plus particulièrement la place des objets-vidéo dans ces approches. Puis, nous proposons le cadre générique VOMF que nous appliquons ensuite au regroupement d'objets-vidéo.

Le dernier chapitre conclut la thèse, et en présente le bilan général. Il ouvre également sur les perspectives offertes par ce travail.

Chapitre 1

Segmentation par zones quasi-plates

Sommaire

Dans ce chapitre, nous traitons des définitions existantes de zones quasi-plates ainsi que de leur extension aux données vidéo. Nous introduisons en premier lieu, la notion de segmentation ainsi que les concepts de zones plates et quasi-plates. Puis, nous évoquons les possibilités d'évaluation de telles zones. Ensuite, nous proposons un état de l'art sur les zones quasi-plates. Par la suite, nous étudions les extensions couleur des zones quasi-plates. Puis, nous proposons une extension des zones quasi-plates aux données vidéo via un cadre générique et incrémental pour leur construction à partir de données variées. Nous traitons ensuite du problème du filtrage des zones quasi-plates. Enfin, avant de conclure, nous discutons les différentes implantations possibles des zones quasi-plates.

1.1 Introduction

Dans cette section, nous introduisons en premier lieu la notion de segmentation puis nous présentons les concepts de zones plates et quasi-plates.

1.1.1 Segmentation

Définition

La segmentation d'une image f est une partition de son espace de définition E en n régions $(r_1, r_2, ..., r_n)$ telle que :

1. $\forall i, \;\; r_i \neq \emptyset$
2. $\forall i, j, \;\; i \neq j, r_i \cap r_j = \emptyset$
3. $\cup_{i=1}^{n} r_i = E$

La segmentation d'une image f vérifiera donc les propriétés précédentes et sera notée :

$$seg(f) = \{r_1, ..., r_n\}, \forall i, r_i \in E_f \tag{1.1}$$

Concrètement, le but de la segmentation est généralement d'obtenir des régions homogènes au sens d'un certain critère de similarité (ex : couleur) ou ayant une signification sémantique. Un exemple de segmentation est présenté sur un cas simple dans la figure 1.1. Dans cet exemple, les régions violette et cyan ont une signification sémantique puisqu'elles représentent les deux perroquets présents dans l'image. notons que les deux autres régions ont également une signification sémantique puisqu'elles représentent le fond de l'image.

a *b*

FIGURE 1.1 – a) Image originale, b) Segmentation de l'image (chaque couleur représente une région différente).

Un problème mal posé

Bien que définie de manière simple, la segmentation est un problème complexe. On dit généralement que la segmentation est un problème mal posé. Un problème est bien posé, si et seulement si :

– Une solution existe ;
– La solution est unique.

Si la première condition est vérifiée par toute segmentation (si $I \neq \emptyset$), la seconde ne l'est absolument pas. En effet, il n'existe pas une segmentation parfaite et unique mais une multitude de segmentations possibles pour chaque image. Comme l'illustre la figure 1.2, la première segmentation donne 2 régions, le groupe d'oisillons et le fond, alors que la deuxième donne 4 régions, une pour chaque oisillon et le fond. Ces deux segmentations sont bonnes en terme de cohérence et les régions obtenues ont un sens. On ne peut pas globalement juger qu'une des segmentations est meilleure que l'autre. La segmentation est donc un problème mal posé.

a *b* *c*

FIGURE 1.2 – a) Image originale, b) Première segmentation (2 régions), c) Deuxième segmentation (4 régions).

Le problème n'est donc pas tant la segmentation, mais l'application de la segmentation pour répondre aux besoins précis de l'utilisateur ou du système. La personnalisation de la segmentation en fonction des besoins de l'utilisateur est un problème important qui sera abordé dans ce manuscrit.

Sur-segmentation et sous-segmentation

Le nombre de régions est un critère important pour apprécier la qualité d'une segmentation. Un nombre trop élevé ou trop faible de régions peut rendre une segmentation inutilisable. Ces deux cas sont nommés respectivement sur-segmentation et sous-segmentation.

Une sur-segmentation est une segmentation comprenant un nombre trop important de régions (cf. figure 1.3.b). Dans ce cas, les régions ne sont absolument pas significatives et ne comportent pas assez de pixels pour pouvoir être utilisées en l'état. Afin de pouvoir exploiter la segmentation, on réduit généralement la sur-segmentation en fusionnant les régions adjacentes similaires.

A l'inverse, une sous-segmentation est une segmentation comprenant un nombre trop faible de régions (cf. figure 1.3.c). Ce problème est plus compliqué à résoudre que la sur-segmentation. Idéalement, il suffit de resegmenter le contenu des régions obtenues. Mais dans certains cas particuliers, par exemple quand une région intéressant l'utilisateur se trouve dans plusieurs régions de la sous-segmentation, la solution nécessitera de resegmenter pour provoquer une sur-segmentation. La résolution de la sur-segmentation avec des fusions permettra alors d'obtenir la segmentation voulue par l'utilisateur.

Ces deux situations sont, dans l'idéal, à éviter. Cependant, la sur-segmentation d'une image suivie de divers traitements réduisant cette sur-segmentation est la base de nombreuses méthodes de segmentation. Une sur-segmentation est donc préférable à une sous-segmentation. La sur-segmentation est même parfois utilisée comme pré-traitement en vue d'une segmentation. Ce cas de figure sera abordé dans ce manuscrit.

Il existe une multitude de méthodes de segmentation, notre choix s'est porté sur les méthodes par zones plates comme discuté dans la suite.

a *b* *c*

FIGURE 1.3 – a) Image originale, b) Image sur-segmentée (9642 régions), c) Image sous-segmentée (2 régions).

1.1.2 Zones plates et quasi-plates

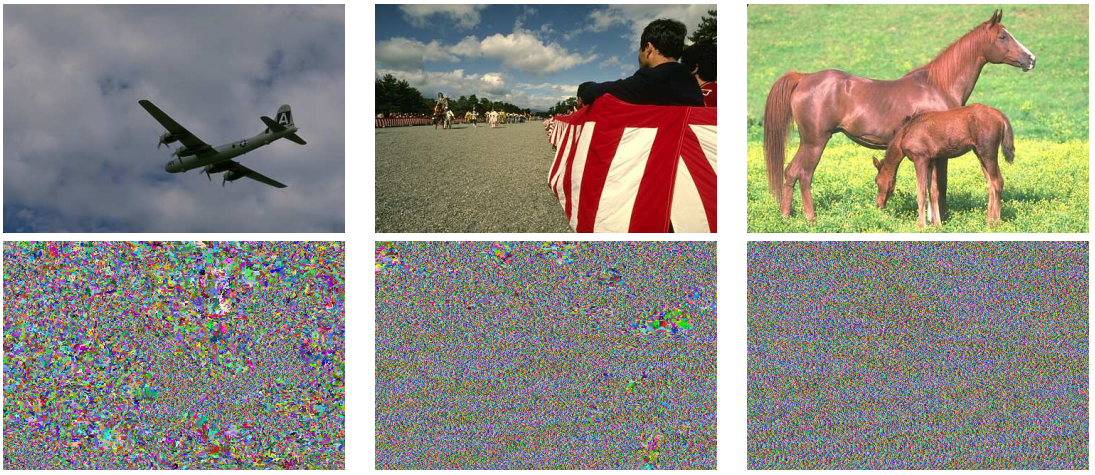
Zones plates

L'un des moyens les plus simples de segmenter une image est de la partitionner en zones plates [SS93]. Les zones plates sont l'extension du concept de composantes connexes aux images non-binaires. Une zone plate est donc un ensemble de pixels connexe, au sens d'un certain voisinage, dont les valeurs de pixel sont identiques.

L'intérêt principal des zones plates est la justesse de leurs frontières. En effet, les frontières des objets d'une image sont généralement situées entre des pixels adjacents de valeurs différentes. Donc en suivant l'hypothèse communément adoptée, on sait que les frontières des objets seront incluses dans les frontières des zones plates. On peut en déduire que les objets d'une image sont souvent décomposables en zones plates. Une fusion judicieuse des zones plates permet donc de segmenter les objets d'une image. Cependant, il est rare de trouver de vastes zones composées de pixels ayant la même valeur. Les zones plates sont souvent composées de quelques pixels seulement (en particulier dans les images multibandes). Ce phénomène induit une sur-segmentation très importante (cf. figure 1.4), ce qui rend les zones plates peu adaptées à la segmentation. Cependant, des méthodes de segmentation basées sur les zones plates ont été développées [CS94][CSS+97][SS95]. Elles s'appuient soit sur des pré-traitements de l'image soit sur des post-traitements des zones plates. Outre la justesse de leurs frontières, les zones plates ont également un coût calculatoire faible. En effet, la construction d'une zone plate se fait par propagation et se limite donc, en chaque pixel, à l'analyse de ses voisins.

Zones quasi-plates

Comme nous venons de l'expliquer, les zones plates offrent des propriétés intéressantes pour la segmentation mais produisent une importante sur-segmentation. Afin de réduire cette sur-segmentation tout en conservant les propriétés intéressantes des zones plates, un critère de construction moins restrictif a été proposé [NMI79]. Ce critère conduit à la production de zones plus grandes, que l'on nomme Zones Quasi-Plates (ZQP) [MM99] (ou parfois zones λ-plates [ZM02]) dans le cadre de la Morphologie Mathématique. Ces zones présentent également un coût calculatoire peu élevé. Cependant les frontières des objets peuvent ne plus être incluses dans les frontières de ces ZQP. La taille des ZQP et la précision de leurs frontières relativement à celles des objets sont dépendantes de différents paramètres. Nous présentons les ZQP dans la section 1.3. Soille [Soi08] précise que les ZQP ne sont pas réellement des méthodes de segmentation mais plutôt des méthodes qui décomposent une image en pièces de puzzle. Il faut donc assembler ces pièces de puzzle pour obtenir la segmentation désirée. Les ZQP sont alors une étape de pré-traitement dans un processus de segmentation d'image basé sur la fusion de pièces de puzzle. La fusion de ces différentes pièces peut-être guidée par différents critères (intensité, taille, etc.) et/ou par l'utilisateur dans le but de résoudre les problèmes de sur-segmentation et de personnalisation de résultats abordés dans la section 1.1.1.



58 328 ZP ($\simeq 2,6$ pixels/ZP) 132 029 ZP ($\simeq 1,2$ pixels/ZP) 151 045 ZP ($\simeq 1,0$ pixel/ZP)

FIGURE 1.4 – Illustration des zones plates de trois images de la base de Berkeley [MFTM01]. Le nombre de zones plates présentes dans chaque image est indiqué, ainsi que le nombre moyen de pixels par zone plate.

Dans la section suivante, nous abordons la question de l'évaluation des zones quasi-plates.

1.2 Évaluation

Dans l'optique de comparer les différentes approches qui permettent la construction de zones quasi-plates, il est nécessaire de disposer d'un moyen d'évaluation. Il existe de nombreuses méthodes pour évaluer des méthodes de segmentation (le lecteur intéressé pourra notamment consulter : [Zha96, Zha01, Cha05, ZFG08]). Cependant, les zones quasi-plates ne sont pas des méthodes destinées à produire une segmentation finale mais plutôt des méthodes de sur-segmentation. Il ne s'agit donc pas d'évaluer la segmentation en elle-même mais la qualité de la sur-segmentation dans l'optique d'un assemblage de ses différentes régions afin d'obtenir un résultat définitif. Nous étudierons en premier lieu des méthodes classiques d'évaluation de segmentation puis nous présenterons des méthodes adaptées à l'évaluation de sur-segmentation. Nous présenterons enfin les vérités-terrain que nous utiliserons pour l'évaluation. Nous n'aborderons dans ce chapitre que l'évaluation du résultat et non l'évaluation du processus ayant permis de l'obtenir.

1.2.1 Méthodes classiques d'évaluation

Les méthodes d'évaluation de segmentation se divisent en deux catégories : les méthodes non supervisées reposant uniquement sur des critères statistiques et les méthodes supervisées utilisant une vérité-terrain. De plus, à l'instar des approches de segmentation qui peuvent être basées contours ou régions, les méthodes d'évaluation peuvent également s'appuyer sur les notions duales que sont les contours ou les régions.

Méthodes non supervisées

Les méthodes d'évaluation de segmentation non supervisées reposent sur des critères purement statistiques. Elles sont utiles lorsqu'il est nécessaire d'évaluer une segmentation sans disposer de la moindre vérité-terrain ou d'une expertise de la part d'un utilisateur.

Pour une évaluation basée région, les méthodes non supervisées visent principalement à évaluer l'homogénéité intra-région [CD85], l'hétérogénéité inter-régions [LN85] ou une conjugaison de ces deux aspects [Zeb88]. L'évaluation non-supervisée des frontières est plus délicate : les méthodes peuvent reposer sur la différence des valeurs de pixels de chaque côté de la frontière [LN85], le gradient moyen le long de la frontière [LN85] ou encore la structure des frontières [KR84].

Les méthodes non-supervisées présentent l'avantage d'être automatiques, elles ne peuvent cependant pas s'adapter aux particularités d'un problème donné.

Méthodes supervisées

Les méthodes d'évaluation de segmentation supervisées nécessitent une vérité-terrain afin de pouvoir comparer la segmentation à évaluer et une segmentation de référence.

L'évaluation supervisée basée région est proche de l'évaluation en classification : il va s'agir de mesurer les erreurs de classification. L'évaluation repose alors sur différents critères de mesure qui vérifient si les pixels de la segmentation à évaluer sont affectés à la même région que dans la segmentation de référence [YMB77]. L'évaluation basée frontières évalue la différence entre les frontières de référence et les frontières de la segmentation à évaluer, que ce soit en mesurant la distance spatiale entre deux frontières [Bad92] ou en utilisant des critères de similarité entre les deux frontières [CP00].

Les méthodes supervisées permettent une évaluation plus fine que les méthodes non-supervisées mais il est parfois difficile d'obtenir une vérité-terrain.

1.2.2 Méthodes adaptées à la sur-segmentation

Nous considérons les zones quasi-plates comme des pièces de puzzle qu'il s'agit d'assembler afin d'obtenir la segmentation désirée par un utilisateur. Nous cherchons donc à évaluer la difficulté d'assemblage et la précision maximale de l'assemblage par rapport à une segmentation de référence.

La difficulté d'assemblage est évaluée par le ratio de sur-segmentation (RSS) [CDW05]. Ce ratio est défini comme le rapport entre le nombre de régions de la segmentation à évaluer et le nombre de régions de la segmentation de référence :

$$RSS = \frac{\text{card}(seg)}{\text{card}(ref)} \tag{1.2}$$

Ce ratio est plus pertinent que le nombre de régions, puisqu'il indique que, pour obtenir la segmentation la plus proche de la segmentation de référence, il faudra en moyenne (pour chaque région de la vérité-terrain) fusionner un nombre de régions égal au RSS. Le RSS détermine donc un degré de fusion de régions nécessaire pour obtenir une segmentation proche de la référence, ce qui marque une certaine difficulté : en effet, plus le RSS est élevé, plus il sera nécessaire d'effectuer des fusions de régions.

La précision maximale (PM) [DFWL10] mesure la similarité basée pixel maximale par rapport à la référence qu'il est possible d'atteindre en fusionnant de manière optimale les régions. On la mesure en affectant chaque région de la sur-segmentation à la région de la référence avec laquelle elle partage le plus grand nombre de pixels. On peut alors définir une matrice de confusion C, où C_{ij} représente le nombre de pixels affectés à la région i mais appartenant à la région j. La PM est définie comme le ratio du nombre de pixels affectés à la bonne région par rapport au nombre total de pixels :

$$PM = \frac{\Sigma_{i=1}^{\text{card}(ref)} C_{ii}}{\text{card}(E_f)} \tag{1.3}$$

Obtenir une PM de 0.80 signifie qu'en assemblant les régions de la sur-segmentation de façon optimale, on obtiendrait au mieux une similarité basée pixel de 80% par rapport à la vérité-terrain.

Ces deux mesures évaluent deux aspects très différents de la qualité d'une sur-segmentation et vont nous permettre par la suite de comparer différentes sur-segmentations. L'objectif pour obtenir une bonne sur-segmentation est de faire tendre RSS vers 1 tout en maximisant PM.

1.2.3 Vérité-terrain

Nous présentons dans cette section les vérités-terrain que nous utilisons dans le reste du chapitre. Nous utilisons la base de segmentation de Berkeley ainsi que quelques séquences vidéo usuelles du domaine pour lesquelles nous avons nous-même construit une segmentation de référence.

1.2.3.1 Base de segmentation de Berkeley

La base de segmentation de Berkeley [MFTM01] est une base de référence dans le domaine de la segmentation d'images. Elle contient 300 images couleurs variées (481 × 321 pixels) qui constituent un échantillon représentatif du panel d'images généralistes que l'on peut être amené à traiter. Cette base contient également 3269 segmentations manuelles effectuées par 28 utilisateurs différents. Ainsi chaque image présente plusieurs segmentations de référence, ce qui permet d'évaluer et de comparer les méthodes de segmentation selon les besoins de différents utilisateurs. Outre la présence de plusieurs segmentations de référence par image, cette base présente un autre intérêt. Par la grande diversité d'images qu'elle présente, elle permet de valider les résultats de méthodes de segmentation de façon générique. Ainsi, une méthode présentant des résultats de bonne qualité sur l'ensemble de la base a de grandes probabilités de fournir des résultats de bonne qualité sur n'importe quel type de photos (hors imagerie spécifique telle que satellitaire ou médicale). Le fait que chaque image ait plusieurs segmentations de référence représentant différents besoins d'un utilisateur la rend particulièrement adaptée à l'évaluation des ZQP. En effet, l'intérêt des ZQP est d'avoir une sur-segmentation de l'image permettant d'obtenir, par assemblage des régions, les objets désirés par l'utilisateur. Le fait de disposer de plusieurs segmentations en objets d'intérêt de la même image permet de vérifier que cette sur-segmentation est robuste aux différents besoins de différents utilisateurs. Un extrait de la base est présenté en figure 1.5.

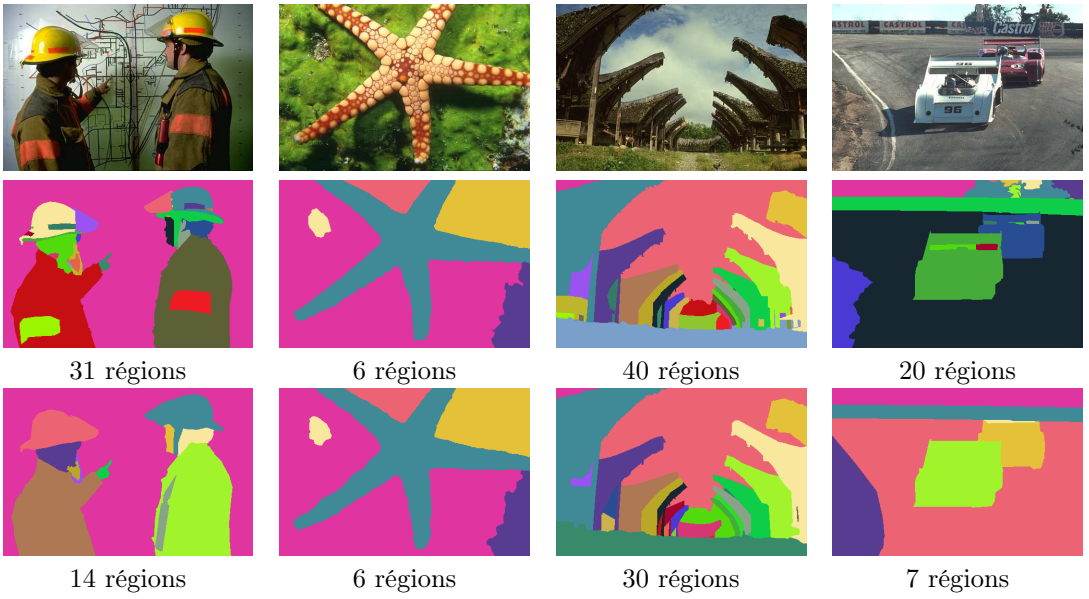


FIGURE 1.5 – Echantillon d'images de la base de Berkeley, accompagnées de deux exemples de segmentations de référence.

1.2.3.2 Segmentations vidéo de référence

L'évaluation des ZQP sur les données vidéo est plus problématique : en effet il n'existe pas de base équivalente à la base de Berkeley dans le cas des séquences vidéo. Il existe des bases vidéo spécialisées, certaines avec des vérités-terrain, mais aucune base vidéo généraliste avec vérité-terrain. Nous avons dû créer notre propre vérité-terrain. Pour cela, nous avons développé ODESSA (cf. annexe C), à l'aide de ce logiciel d'assistance à la création de segmentation de séquences vidéo de référence et à l'évaluation de segmentations vidéo. Utilisant notre logiciel nous avons réalisé des segmentations de référence de deux extraits de séquences vidéo très utilisées par la communauté du traitement de la vidéo : *carphone* et *foreman* [1]. Ces deux séquences présentent des difficultés au niveau de la segmentation. La séquence *carphone* contient un objet d'intérêt (l'*homme*) composé de parties de différentes couleurs et en mouvement. Elle contient également deux types de fond.

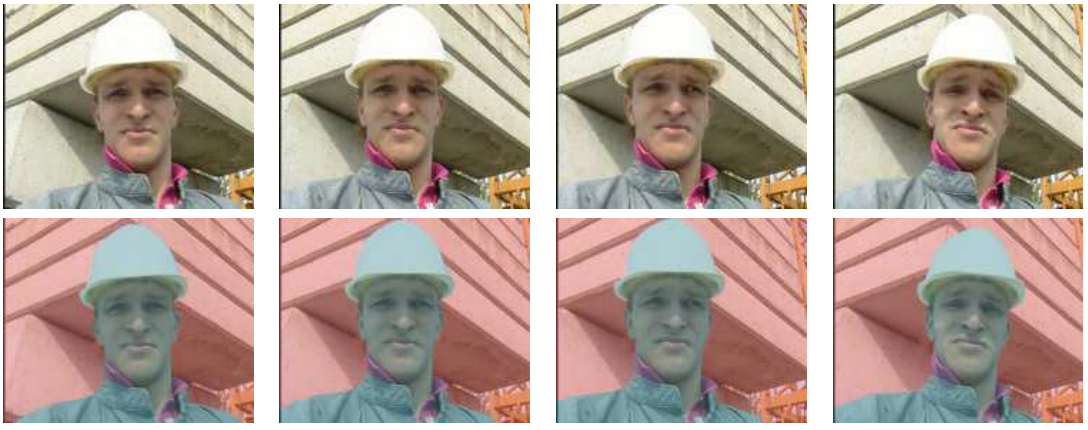
1. Ces séquences vidéo peuvent être récupérées à l'adresse http ://media.xiph.org/video/derf/

Le premier est un fond fixe dont certaines parties sont visuellement proches de régions de l'*homme* (les bandes noires de la vitre arrière de la voiture et la chevelure de l'*homme*. Le second est un fond mobile, qui est le défilement rapide du paysage par les vitres de la voiture, sa segmentation implique de regrouper des choses aussi différentes qu'un ciel bleu et des feuillages verts. La séquence *foreman* présente un ouvrier devant les murs clairs d'un bâtiment. La difficulté réside ici dans les mouvements de l'ouvrier et de la caméra mais aussi dans la proximité chromatique du casque de l'ouvrier et des murs clairs derrière lui qui risquent d'entrainer une sous-segmentation en fusionnant casque et murs.

Pour la séquence *carphone*, nous avons réalisé une segmentation de référence sur un extrait de 80 trames contenant trois classes : l'*homme*, l'*intérieur de la voiture* et l'*extérieur* (cf. figure 1.6). Pour la séquence *foreman*, nous avons réalisé une segmentation de référence sur un extrait de 40 trames contenant deux classes : l'*homme* et le *fond* (cf. figure 1.7).

Trame 1 Trame 27 Trame 57 Trame 80

FIGURE 1.6 – Extrait de la séquence *carphone* et segmentation de référence associée.



Trame 1 Trame 14 Trame 27 Trame 40

FIGURE 1.7 – Extrait de la séquence *foreman* et segmentation de référence associée.

1.3 Zones quasi-plates, un état de l'art

Nous allons maintenant présenter les méthodes existantes pour la production de zones quasi-plates. Ces méthodes ont été développées pour des images fixes en niveaux de gris. Nous verrons ensuite comment étendre ces méthodes aux images couleur dans la section 1.4 puis aux séquences vidéo dans la section 1.5.

1.3.1 Notions préliminaires

Les zones quasi-plates sont basées sur les concepts d'adjacence, de chemins et de connexité lipschitzienne. Dans cette section, nous rappelons ces concepts.

L'adjacence est basée sur les relations de voisinage. Deux pixels étant adjacents selon un certain voisinage N, la figure 1.8 présente le 4-voisinage (où un pixel a 4 pixels voisins) et le 8-voisinage (où un pixel a 8 pixels voisins), le pixel central étant adjacent aux pixels visés par les flèches. Si p appartient au 8-voisinage de q, on note $p \in N_8(q)$. Dans la suite nous utiliserons le 8-voisinage quand nous traiterons des images $2D$.

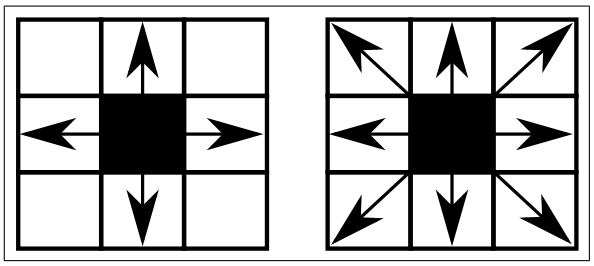


FIGURE 1.8 – Le 4-voisinage et le 8-voisinage.

En s'appuyant sur la notion d'adjacence, nous définissons à présent les chemins. Un chemin $\mathcal{P}(p,q)$ est, selon un voisinage N, une suite de n pixels ayant les pixels p et q comme extrémités, construisant ainsi un n-uplet de pixels $(p_0, p_1, ..., p_{n-1})$ respectant la condition suivante :

$$\mathcal{P}(p,q) = \cup_{i\in[0,n-1]}\{p_i | p_0 = p \text{ et } p_{n-1} = q \text{ et } \forall i \in [0, n-2], p_i \in N(p_{i-1})\} \tag{1.4}$$

Les notions d'adjacence et de chemin sont suffisantes pour définir les zones plates. La ZP à laquelle appartient un pixel p est notée $\mathcal{Z}(p)$ pour *zone de p*, ce principe de notation sera également appliqué aux définitions de ZQP que nous allons présenter dans la section 1.3.2. Une zone plate est définie par :

$$\mathcal{Z}(p) = \{p\} \cup \{Q \mid \forall q \in Q, \forall p_i, p_j \in \mathcal{P}(p,q), f(p_i) = f(p_j)\} \tag{1.5}$$

Afin de pouvoir définir les ZQP, nous avons besoin de la connexité lipschitzienne, basée sur la définition des fonctions lipschitziennes :

$$|f(p) - f(q)| \leq \alpha |p - q| \quad \forall p, q \in I \tag{1.6}$$

où $f : I \to \mathbb{N}$, $\alpha \in V$ et $I \subseteq E_f$.

On en déduit la condition de la connexité lipschitzienne :

$$|f(p) - f(q)| \leq \alpha(d(p,q)) \quad \forall p, q \in E_f \tag{1.7}$$

où $d(p,q)$ est la distance spatiale entre les coordonnées spatiales des pixels p et q.

La connexité lipschitzienne est utilisée pour construire des chemins entre les pixels de l'image. Comme nous traitons des pixels adjacents ($p \in N(q)$), nous posons $d(p,q) = 1$. Un chemin entre deux pixels adjacents est Lipschitz-continu si les pixels du chemin respectent la condition (1.7). Comme nous avons posé $d(p,q) = 1$ si $p \in N(q)$, les pixels adjacents d'un chemin Lipschitz-continu doivent respecter la condition suivante :

$$|f(p) - f(q)| \leq \alpha \quad \forall p, q \in E_f \tag{1.8}$$

La figure 1.9 présente par des segments rouges les chemins Lipschitz-continus pour $\alpha = 3$.

Un chemin est α-connexe si tous les chemins de deux pixels le composant sont Lipschitz-continus. On déduit de la condition 1.8 la définition de l'α-connexité :

Un chemin $\mathcal{P}$, selon un voisinage N, composé de n pixels $(p_0, p_1, ..., p_{n-1})$ est α-connexe si et seulement si :

$$\forall i \in [0, n-2], p_i \in N(p_{i+1}) \text{ et } |f(p_i) - f(p_{i+1})| \leq \alpha \tag{1.9}$$

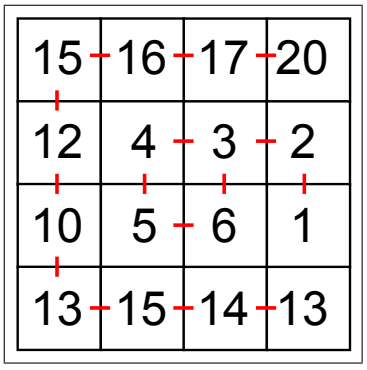


FIGURE 1.9 – Chemins Lipschitz-continus (en rouge) pour $\alpha = 3$.

On note α-$\mathcal{P}(p, q)$, l'ensemble des chemins α-connexes entre p et q.

Nous avons rappelé les notions qui sont nécessaires à la construction des zones quasi-plates. Nous pouvons donc étudier les définitions des ces zones dans la section suivante.

1.3.2 Définitions

Il n'existe pas de définition unique des ZQP. Différentes définitions des ZQP ont été introduites afin de s'adapter aux besoins des utilisateurs. Nous allons maintenant les présenter.

La première définition pouvant s'apparenter à celle des zones quasi-plates a été introduite par Nagao *et al.* [NMI79]. Il s'agit de l'application directe de la notion d'α-connexité présentée dans la section 1.3.1. Elle produit des ZQP que nous noterons α-$\mathcal{Z}$ dans la suite du document. L'α-$\mathcal{Z}$ est définie comme suit :

$$\alpha\text{-}\mathcal{Z}(p) = \{p\} \cup \{Q \mid \forall q \in Q, \alpha\text{-}\mathcal{P}(p, q) \neq \emptyset\} \tag{1.10}$$

L'α-$\mathcal{Z}$ d'un pixel p est donc l'ensemble des pixels auquel il est relié par un chemin α-connexe. On note que les zones plates sont un cas particulier de l'α-$\mathcal{Z}$ avec $\alpha = 0$. Les α-$\mathcal{Z}$ ont la propriété hiérarchique suivante :

$$\forall \alpha' \leq \alpha, \alpha'\text{-}\mathcal{Z}(p) \subseteq \alpha\text{-}\mathcal{Z}(p) \tag{1.11}$$

Cette propriété sera très intéressante quand nous chercherons à produire certains types de ZQP. L'α-$\mathcal{Z}$ est une définition simple mais elle souffre d'un défaut important. En effet, la connexité s'établissant selon une différence maximale α des valeurs de pixel au niveau local, il est possible de rapidement englober toute l'image dans une seule ZQP. Une ZQP peut alors être composée de pixels ayant des valeurs très différentes mais qui sont reliés par des chemins α-connexes. Cette *réaction en chaîne* est illustrée par la figure 1.10 où l'α-$\mathcal{Z}$ ne produit qu'une seule ZQP alors que les valeurs des pixels de l'image présentent une grande variation de niveaux de gris (souvent la plage complète de niveaux de gris d'une image codée sur 8 bits).

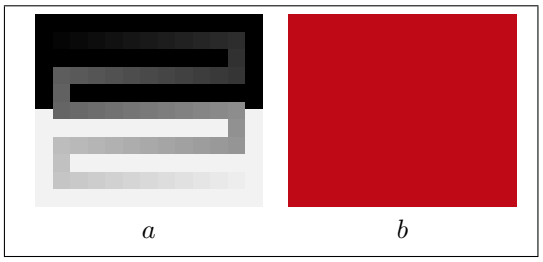


FIGURE 1.10 – Illustration de la *réaction en chaîne* : a) Image originale, b)α-$\mathcal{Z}$ avec $\alpha = 5$ (1 zone).

Afin de contrer la *réaction en chaîne*, Hambrusch *et al.* [HHM94] ont introduit un nouveau paramètre ω. A l'inverse de α, ω est un seuil de variation globale. La différence entre la plus petite et la plus grande valeur de pixel de la ZQP doit être inférieure ou égale à ω. Nous appelons cette différence de valeur *variation globale* et nous la notons Ω. Elle est définie plus formellement par l'équation suivante :

$$\Omega(\mathcal{Z}) = \max_{p,q \in \mathcal{Z}} \{|f(p) - f(q)|\} \tag{1.12}$$

En ajoutant cette contrainte à l'α-$\mathcal{Z}$, ils créent l'(α,ω)-$\mathcal{ZH}$ (pour *zone de Hambrusch*) qui est définie comme suit :

$$(\alpha,\omega)\text{-}\mathcal{ZH}(p) = \{p\} \cup \{Q \mid \Omega(p \cup Q) \leq \omega \text{ et } \forall q \in Q, \alpha\text{-}\mathcal{P}(p,q) \neq \emptyset\} \tag{1.13}$$

L'introduction de ce paramètre permet effectivement de résoudre le défaut de l'α-$\mathcal{Z}$ comme le montre la figure 1.11.

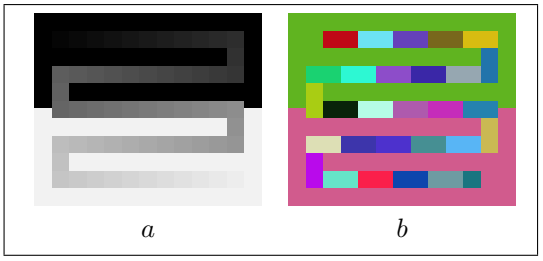


FIGURE 1.11 – Réduction de la *réaction en chaîne* par (α,ω)-$\mathcal{ZH}$: a) Image originale, b) (α,ω)-$\mathcal{ZH}$ avec $\alpha = 5$ et $\omega = 5$ (31 zones).

Cependant l'(α,ω)-$\mathcal{ZH}$ souffre d'un défaut que n'avait pas l'α-$\mathcal{Z}$: la non-unicité de la segmentation qu'elle produit. En effet, comme l'illustre la figure 1.12, les ZQP produites par l'(α,ω)-$\mathcal{ZH}$ sont dépendantes de l'ordre de traitement des pixels. Les deux segmentations obtenues par l'(α,ω)-$\mathcal{ZH}$ de cette figure sont différentes mais cohérentes avec la définition donnée. La non-unicité de la segmentation en ZQP pose alors plusieurs problèmes. Ainsi, une méthode de segmentation dépendant de l'ordre de traitement des pixels sera difficile à évaluer de par la difficulté de savoir si ses résultats sont dépendants des paramètres ou de l'ordre de traitement des pixels. De plus, une définition pouvant mener à différentes segmentations va entrainer un problème de reproductibilité des résultats. Afin de résoudre ces problèmes, Soille [Soi08] a défini un nouveau type de ZQP utilisant les paramètres α et ω mais garantissant l'unicité des ZQP produites, l'(α,ω)-$\mathcal{ZS}$. Pour ce faire, il s'agit de trouver, pour chaque pixel p, la plus grande α-$\mathcal{Z}(p)$ qui vérifie la condition de variation globale. L'α-$\mathcal{Z}$ a la propriété d'unicité : en cherchant la plus grande α-$\mathcal{Z}(p)$ (notée $\max\{\alpha\text{-}\mathcal{Z}(p)\}$) satisfaisant les critères α et ω, nous obtenons donc une ZQP unique. Pour être sûr qu'il existe une telle α-$\mathcal{Z}(p)$, nous nous appuyons sur la propriété 1.11. Il suffit alors de chercher la valeur $\alpha' \leq \alpha$ maximale qui permet d'obtenir une α'-$\mathcal{Z}(p)$ satisfaisant la variation globale maximale ω. Nous avons ainsi la garantie que les ZQP produites par l'(α,ω)-$\mathcal{ZS}$ (pour *zone de Soille*) ne sont pas dépendantes du sens de parcours des pixels. Cette méthode produit donc une segmentation unique comme l'illustre la figure 1.12.c. Elle est définie par :

$$(\alpha,\omega)\text{-}\mathcal{ZS}(p) = \max\{\alpha'\text{-}\mathcal{Z}(p) \mid \alpha' \leq \alpha \text{ et } \Omega(\alpha'\text{-}\mathcal{Z}(p)) \leq \omega\} \tag{1.14}$$

La figure 1.13 présente les (α,ω)-$\mathcal{ZS}$ obtenues sur l'image comportant une forme de type dégradé. Contrairement à l'(α,ω)-$\mathcal{ZH}$, aucune partie du dégradé n'est fusionnée avec les deux parties du fond. De plus, on remarque que, pour garantir l'unicité, les α-$\mathcal{Z}$ produites l'ont été pour des valeurs de α faibles, ce qui a entrainé une sur-segmentation plus importante de la figure de type dégradé. C'est un contre-coup de l'unicité : à valeurs égales, l'(α,ω)-$\mathcal{ZS}$ produira une sur-segmentation plus importante que l'(α,ω)-$\mathcal{ZH}$.

On note que Soille préconise l'utilisation de seuils de variation locale et globale identiques ($\alpha = \omega$). Le respect des paramètres de variation locale et globale ainsi que l'unicité éventuelle nous

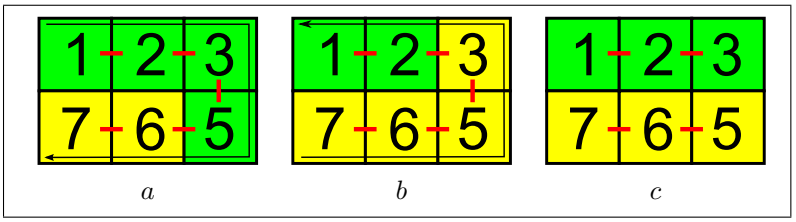


FIGURE 1.12 – Non-unicité de l'(α,ω)-$\mathcal{ZH}$ et unicité de l'(α,ω)-$\mathcal{ZS}$: a et b)(α,ω)-$\mathcal{ZH}$ pour $\alpha = 2$ et $\omega = 4$, c) (α,ω)-$\mathcal{ZS}$.

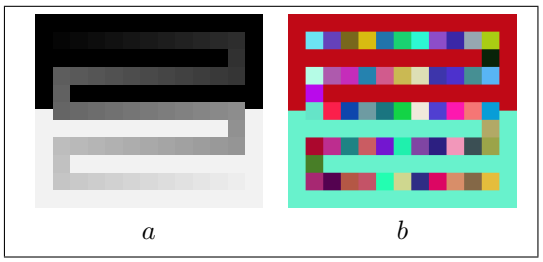


FIGURE 1.13 – Réduction de la *réaction en chaîne* par l'(α,ω)-$\mathcal{ZS}$: a) image originale, b) (α,ω)-$\mathcal{ZS}$ avec $\alpha = 5$ et $\omega = 5$ (61 zones).

donnent une information sur l'homogénéité globale de la ZQP produite, mais aucune information sur l'homogénéité locale entre les pixels voisins dans la ZQP. La figure 1.14 présente une ZQP au sens de l'α-$\mathcal{Z}$ avec $\alpha = 2$. Considérons que nous sommes en 8-voisinage, il existe 11 relations de voisinage entre les 6 pixels de la ZQP. Or, seulement 5 de ces 11 relations internes de voisinage sont α-connexes.

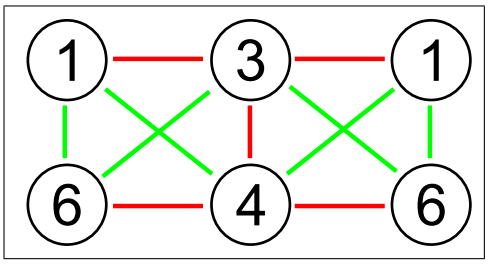


FIGURE 1.14 – ZQP avec 5 arêtes α-connexes sur 11 ($B = 5/11$) pour $\alpha = 2$.

Afin de produire des ZQP avec une homogénéité locale entre les pixels voisins maximale, Soille [Soi08] a proposé un nouveau paramètre β. Ce paramètre est basé sur un *indice de connexité* (B) qui est calculé comme le ratio entre le nombre de relations d'adjacence α-connexes internes à la ZQP et le nombre de relations d'adjacence internes à la ZQP. Pour définir B formellement, nous notons $\#(X)$ l'ensemble des relations d'adjacence internes à X et $\#_\alpha(X)$ l'ensemble des relations d'adjacence internes à X qui sont α-connexes. Nous pouvons alors définir B comme suit :

$$B(\alpha\text{-}\mathcal{Z}(p)) = \frac{card(\#_\alpha(\alpha\text{-}\mathcal{Z}(p)))}{card(\#(\alpha\text{-}\mathcal{Z}(p)))} \tag{1.15}$$

L'indice de connexité doit être supérieur ou égal à β pour que la ZQP soit valide. L'intérêt de ce paramètre est similaire à celui de la variation globale : il permet de réduire le défaut majeur de l'α-$\mathcal{Z}$, c'est-à-dire la possibilité d'obtenir une ZQP englobant toute l'image. En forcant les ZQP à avoir au moins une certaine connexité interne, on réduit le risque d'avoir des ZQP comportant des pixels trop hétérogènes. Si on fixe $\beta = 1$, on obtient l'α-$\mathcal{ZS}$ [Soi08] qui est définie par :

$$\alpha\text{-}\mathcal{ZS}(p) \quad = \quad \max\{\alpha'\text{-}\mathcal{Z}(p) \mid \alpha' \leq \alpha \text{ et } B(\alpha'\text{-}\mathcal{Z}(p)) = 1\} \tag{1.16}$$

La figure 1.15 illustre les α-$\mathcal{ZS}$ obtenues sur l'image de dégradé. Sur cette image, l'α-$\mathcal{ZS}$ produit une seule ZQP pour le serpent dégradé. L'α-$\mathcal{ZS}$ n'extrait cependant pas tous les dégradés, le cas présenté dans cette image est exceptionnel. Néanmoins, l'exemple illustre que l'α-$\mathcal{ZS}$, tout comme l'(α,ω)-$\mathcal{ZS}$, permet de réduire la sous-segmentation en autorisant l'utilisation de valeurs de α qui auraient produit une sous-segmentation avec l'α-$\mathcal{Z}$.

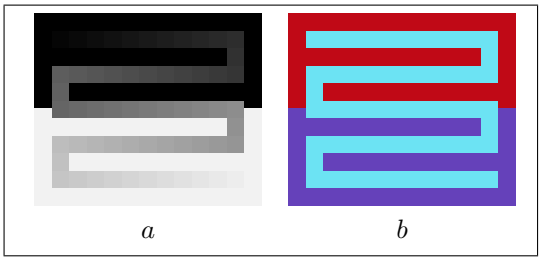


FIGURE 1.15 – Réduction de la *réaction en chaîne* par l'α-$\mathcal{ZS}$: a) image originale, b) α-$\mathcal{ZS}$ avec $\alpha = 5$ (3 zones).

Afin de pouvoir utiliser tous les paramètres présentés mais également la propriété d'unicité, Soille a proposé l'(α,ω,β)-$\mathcal{Z}$:

$$(\alpha,\omega,\beta)\text{-}\mathcal{Z}(p) = \max\{\alpha'\text{-}\mathcal{Z}(p) \mid \alpha' \leq \alpha \text{ et } \Omega(\alpha'\text{-}\mathcal{Z}(p)) \leq \omega \text{ et } B(\alpha'\text{-}\mathcal{Z}(p)) \geq \beta\} \tag{1.17}$$

On note que cette définition est peu utile. En effet, si $\alpha = \omega$, comme cela est préconisé par Soille pour l'(α,ω)-$\mathcal{ZS}$, alors le paramètre β est inutile car l'indice de connexité des ZQP sera toujours égal à 1. Le but de cette définition est principalement d'utiliser ensemble les définitions de variation locale et globale ainsi que celle d'indice de connexité.

Enfin, une définition basée uniquement sur la variation globale, l'ω-$\mathcal{Z}$ [Soi08] a été proposée :

$$\omega\text{-}\mathcal{Z}(p) = \max\{\alpha'\text{-}\mathcal{Z}(p) \mid \Omega(\alpha'\text{-}\mathcal{Z}(p)) \leq \omega\} \tag{1.18}$$

Cette méthode correspond à l'(α,ω)-$\mathcal{ZS}$ dans le cas où $\alpha \geq \omega$.

Nous avons vu qu'il existe différentes définitions des zones quasi-plates et que d'autres pourraient encore être proposées. Nous discutons dans la section suivante de leur unification.

1.3.3 Unification

Définition		α	ω	β	unicité
Zones plates		oui (α=0)	non	oui ($\beta = 1$)	oui
α-$\mathcal{Z}$	[NMI79]	oui	non	non	oui
(α,ω)-$\mathcal{ZH}$	[HHM94]	oui	oui	non	non
(α,ω)-$\mathcal{ZS}$	[Soi08]	oui	oui	non	oui
(α,ω,β)-$\mathcal{Z}$	[Soi08]	oui	oui	oui	oui
α-$\mathcal{ZS}$	[Soi08]	oui	non	oui ($\beta = 1$)	oui
ω-$\mathcal{Z}$	[Soi08]	non	oui	non	oui

TABLE 1.1 – Les différentes définitions de ZQP ainsi que leurs paramètres et propriétés.

Soille et Grazzini [Soi07, SG09] ont proposé un cadre théorique pour unifier les définitions des ZQP, la *connexité des prédicats logiques*. En effet, il suffit d'observer le tableau 1.1 présentant les définitions existantes des ZQP pour noter qu'elles sont issues d'une combinaison de 4 paramètres rendant possible leur unification. La formulation de cette unification considère toutes les conditions présentées dans la section 1.3.2 comme des prédicats logiques. Nous rappelons qu'un prédicat

logique renvoie *vrai* quand l'argument satisfait le prédicat, $faux$ sinon. Soille et Grazzini définissent donc un nouveau type de ZQP qu'on note $(P_1, ..., P_n)$-$\mathcal{Z}$ (avec P l'ensemble des prédicats) et qui permet de produire des ZQP vérifiant les n prédicats logiques. Les auteurs en donnent la formulation mathématique suivante :

$$(P_1, ..., P_n)\text{-}\mathcal{Z}(p) = \bigvee \left\{ \alpha'\text{-}\mathcal{Z}(p) \;\middle|\; \begin{array}{r} \forall k \in \{1, ..., n\} \quad P_k\left(\alpha'\text{-}\mathcal{Z}(p)\right) = vrai \\ \forall \alpha'' \leq \alpha', \forall q \in \alpha'\text{-}\mathcal{Z}(p), \quad P_k\left(\alpha''\text{-}\mathcal{Z}(q)\right) = vrai \end{array} \right\} \tag{1.19}$$

Ce cadre théorique est adapté aux méthodes ayant la propriété d'unicité : en effet, on cherche la plus grande α'-$\mathcal{Z}$ satisfaisant les prédicats logiques utilisés. Ce cadre n'est par contre pas adapté aux méthodes ne produisant pas une segmentation en ZQP unique. Plus qu'un cadre permettant d'unifier les méthodes existantes, la $(P_1, ..., P_n)$-$\mathcal{Z}$ permet néanmoins d'intégrer les autres définitions existantes des ZQP. Actuellement, les méthodes utilisent trois prédicats : la variation locale (représentée par le paramètre α), la variation globale ($P_i = \Omega$) et l'indice de connectivité ($P_i = B$). Nous pouvons considérer dans ce cadre des prédicats portant sur d'autres caractéristiques des ZQP (périmètre, aire, etc.) mais également sur des descripteurs plus complexes (variation de texture, gradient, etc.), sous réserve que ces prédicats respectent la définition (1.19). Etudions le cas de l'aire, où deux possibilités s'offrent tout de suite à nous : un prédicat sur l'aire minimale d'une ZQP, et un prédicat sur l'aire maximale d'une ZQP. Or, si le prédicat sur l'aire maximale est compatible avec le cadre des prédicats logiques (si $\alpha' \leq \alpha$ alors $Aire(\alpha'\text{-}\mathcal{Z}) \leq Aire(\alpha\text{-}\mathcal{Z})$ grâce à la propriété 1.11), le prédicat sur l'aire minimale ne l'est pas. En effet, si la condition sur l'aire minimale n'est pas vérifiée pour α, elle ne le sera pas non plus pour $\alpha' \leq \alpha$.

Nous notons que très récemment, la notion d'α-arbre a été introduite [OS11, Soi11]. Elle se base sur la propriété de hiérarchie des α-$\mathcal{Z}$ (cf. équation 1.11). Il s'agit de créer les α-$\mathcal{Z}$ successives en augmentant progressivement la valeur de α. On obtient ainsi un arbre dont les feuilles sont les zones plates de l'image (puisque obtenues avec $\alpha = 0$) et la racine contient tous les pixels de l'image. Dans le cadre des prédicats logiques, on peut effectuer des coupes dans l'arbre pour obtenir les zones quasi-plates satisfaisant les conditions requises. Pour la contrainte de variation globale, on utilisera le terme de ω-coupes.

Les opérateurs permettant la construction des zones quasi-plates, unifiés par la connexité des prédicats logiques, font partie de la famille des opérateurs morphologiques connexes [SS95, Ser98]. Ces opérateurs segmentent un espace (ici une image) en extrayant ses composantes connexes, selon une connexité définie par un ou plusieurs critères. Les opérateurs connexes respectent la définition de la segmentation que nous avons introduite dans la section 1.1.1, et notamment la non intersection des composantes connexes. Nous observons, cependant, que ces dernières années ont vu le développement de nouveaux types de connexions : les hyper-connexions [BNG03, Ser06, Wil, PLC11] dont les opérateurs connexes morphologiques sont des cas particuliers. La différence principale est la disparition de la condition de non intersection des composantes connexes et son remplacement par une définition du recouvrement entre composantes connexes propre à chaque opérateur hyper-connexe (recouvrement nul dans le cas particulier des opérateurs morphologiques connexes). Ainsi, nous pouvons avoir un recouvrement des composantes, ce qui permet dans le cas des zones quasi-plates d'utiliser simultanément plusieurs niveaux de la hiérarchie qu'ils produisent, c'est-à-dire par exemple plusieurs valeurs des paramètres α et ω. Nous n'abordons pas l'hyper-connexité dans ce document mais indiquons simplement que les méthodes permettant la construction de zones quasi-plates s'inscrivent dans un contexte plus général d'opérateurs connexes et hyper-connexes. Nous notons, par ailleurs, que les zones quasi-plates ont été utilisées récemment pour le filtrage d'attributs hyperconnexes [OW11].

Nous avons evoqué les différentes définitions de zones quasi-plates et étudié leur unification via la connexité des prédicats logiques. Nous nous intéressons dans la section suivante à leur application.

1.3.4 Applications

Les ZQP sont utilisées principalement comme outil de simplification d'image et comme première étape d'un processus de segmentation.

Dans l'optique de la simplification d'image, l'intérêt des ZQP est de produire des zones d'intensité homogène. L'idée est alors d'affecter à chaque ZQP une valeur qui sera identique pour tous les pixels de la ZQP dans l'image simplifiée. Soille [Soi08] propose d'affecter la valeur moyenne des pixels de la ZQP dans l'image d'origine aux pixels de la ZQP dans l'image simplifiée. Des travaux ont également été réalisés par Brunner et Soille [BS07] qui, en plus de simplifier l'image spectralement, simplifient également la forme des ZQP pour faciliter la vectorisation de l'image. La simplification d'image est utilisée ici principalement pour préparer une future compression ou comme filtrage de l'image avant l'application d'un opérateur de segmentation.

Dans un but de segmentation, les ZQP sont utilisées comme une première étape de segmentation. Cette segmentation sera ensuite affinée par divers processus de fusion de régions. Zanoguera [Zan01], dans ses travaux de thèse, utilise les ZQP produites par α-$\mathcal{Z}$ comme marqueurs pour une ligne de partage des eaux. Les régions obtenues par la ligne de partage des eaux sont considérées comme des nœuds d'un arbre dont les arêtes sont des relations d'adjacence. La fusion de nœuds de l'arbre permet d'obtenir une hiérarchie de partitions dans laquelle l'utilisateur navigue pour adapter la segmentation à ses besoins. On peut noter que Zanoguera fixe, comme paramètre α, la valeur qui permet aux ZQP de taille (en pixels) supérieure à un seuil d'occuper 80-90% de l'image. Cette méthode a été utilisée pour segmenter des séquences d'images couleur. L'inconvénient de cette approche est le réglage du paramètre α : en effet, cette valeur est différente selon les images et difficile à déterminer.

Angulo et Serra [AS03] proposent une méthode de segmentation similaire. Une fois la partition de l'image en ZQP obtenue, les ZQP et leurs relations de voisinage sont modélisées par un graphe d'adjacence. Chaque ZQP dont l'aire est inférieure à un seuil a est fusionnée avec la ZQP voisine la plus spectralement similaire. Cette méthode est destinée aux images couleur (dans l'espace IHLS), l'opération précédente étant appliquée indépendamment sur chaque bande. Les résultats sont ensuite combinés pour obtenir une segmentation unique. Les auteurs n'explicitent pas comment se font les choix des paramètres α et a. Toutefois, on peut affirmer que ces paramètres ne sont pas uniques et qu'ils dépendent des images sur lesquelles la méthode est appliquée. Déterminer une valeur optimale pour ces paramètres n'est pas trivial.

Crespo *et al.* [CSS+97] ont utilisé les ZP et non les ZQP. Ils produisent d'abord les ZP de l'image, puis sélectionnent les n ZP les plus significatives selon plusieurs critères. Ces ZP seront les graines d'un algorithme de croissance de régions. Cet algorithme, au lieu d'accroitre les régions "graines" en les fusionant avec les pixels connexes, va les fusionner avec les ZP connexes n'appartenant pas encore à des régions "graines" (une ZP étant fusionnée avec la région "graine" connexe la plus similaire selon divers critères). Cette méthode a été développée pour les images en niveaux de gris et également adaptée aux images couleur [CS94]. Un des avantages est la possibilité de fixer le nombre de régions que l'utilisateur souhaite obtenir. On peut également utiliser des critères différents pour la sélection des graines et pour la croissance de régions. L'intérêt d'utiliser une croissance de régions plutôt que de fusionner les ZP similaires est que l'on est assuré de garder toutes les régions d'intérêt de la première étape de sélection.

Nous pensons qu'il est préférable de fusionner les ZQP en s'appuyant sur différents descripteurs, et en faisant intervenir l'utilisateur dans le processus pour guider et corriger la segmentation. Contrairement aux approches présentées ici, nous voulons obtenir un résultat propre à chaque utilisateur. De plus, nous souhaitons que la segmentation d'une séquence vidéo puisse guider la segmentation d'autres images. Autrement dit, que les opérations effectuées par l'utilisateur pour guider la segmentation doivent être impactées sur le processus de segmentation d'autres séquences vidéo. L'objectif est de permettre à un utilisateur de segmenter une base de séquences vidéo sans avoir à segmenter chaque vidéo indépendamment.

1.3.5 Discussion

Dans cette section, nous avons présenté un état de l'art sur les zones quasi-plates. Nous avons rappelé les différentes définitions existantes ainsi que leurs applications. Nous avons également exposé les travaux d'unification de Soille et Grazzini. Les définitions actuelles des ZQP sont nombreuses et le cadre offert par la connexité des prédicats logiques permet d'en créer de nouvelles, simplement en rajoutant de nouveaux prédicats. Bien qu'il soit intéressant d'explorer les possibilités offertes par l'ajout de nouveaux prédicats afin de contraindre les ZQP à converger vers une segmentation qui sied mieux à l'utilisateur, nous allons nous restreindre dans la suite de ce document à l'utilisation de deux définitions. Nous ne considérerons donc plus que l'α-$\mathcal{Z}$ et l'(α,ω)-$\mathcal{ZS}$, que nous définirons dans le cadre de la connexité des prédicats logiques comme :

$$\alpha\text{-}\mathcal{Z}(p) = (P_1,...,P_n)\text{-}\mathcal{Z}(p) \mid P = \emptyset \quad (1.20)$$

$$(\alpha,\omega)\text{-}\mathcal{ZS}(p) = (P_1,...,P_n)\text{-}\mathcal{Z}(p) \mid P = \{\Omega\} \quad (1.21)$$

Nous pourrions nous attarder sur la connexité des prédicats logiques et particulièrement sur la définition de nouveaux prédicats. Cependant le but de nos travaux sur les ZQP est d'obtenir une sur-segmentation dont la réduction par fusion de ZQP permettra d'obtenir une segmentation proche des désirs de l'utilisateur. Il s'agit donc de minimiser le nombre de régions de la sur-segmentation tout en essayant d'avoir suffisamment de régions pour que différents utilisateurs ayant différents besoins puissent utiliser la même pré-segmentation. Or, comme le montre la figure 1.16 qui compare les résultats obtenus par l'α-$\mathcal{Z}$ et l'(α,ω)-$\mathcal{ZS}$ (cf. section 1.2.2 pour les méthodes d'évaluation), l'(α,ω)-$\mathcal{ZS}$ présente une très nette amélioration de l'α-$\mathcal{Z}$ en termes de précision maximale pour un ratio de sur-segmentation comparable. L'(α,ω)-$\mathcal{ZS}$ présente ainsi déjà une réduction importante de l'espace des données tout en garantissant une précision maximale élevée. Cette définition répond donc aux besoins de nos travaux.

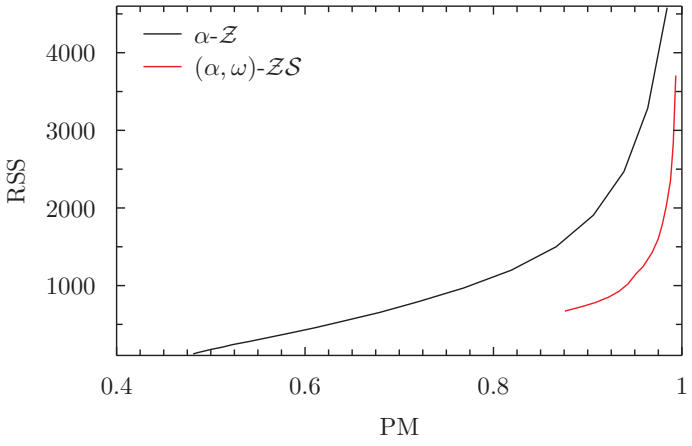


FIGURE 1.16 – Comparaison des valeurs de précision maximale par rapport au ratio de sur-segmentation obtenues par l'α-$\mathcal{Z}$ et de l'(α,ω)-$\mathcal{ZS}$ sur les images de la base de Berkeley pour différentes valeurs de α et ω.

Nous pourrions choisir d'étudier uniquement l'extension de l'(α,ω)-$\mathcal{ZS}$ dans les prochaines sections mais l'(α,ω)-$\mathcal{ZS}$ repose sur l'α-$\mathcal{Z}$ qui est la base des ZQP. Nous étudions donc ces deux définitions dans la suite du chapitre au travers des extensions couleur (cf. section 1.4) et vidéo (cf. section 1.5) des ZQP ainsi que de leur filtrage (cf. section 1.6).

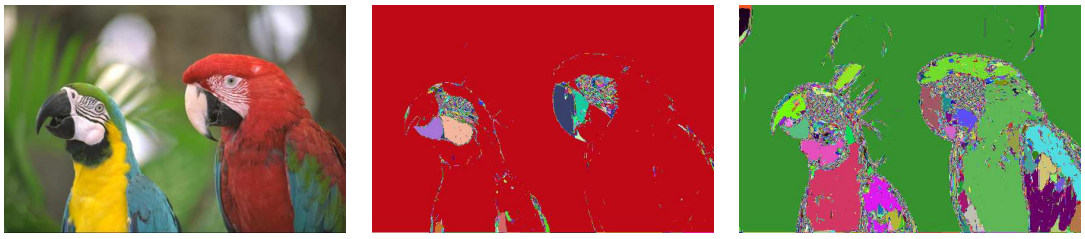
1.4 Extension couleur

Les définitions présentées précédemment ne concernaient que les images en niveaux de gris. Dans cette section nous traitons de leur extension à la couleur. Nous abordons en premier lieu

la nécessité de les étendre à la couleur. Puis, nous rappelons brièvement les approches existantes. Ensuite, nous explorons les possibilités d'extension des ZQP à des images couleur en s'appuyant sur les deux approches alternatives que sont la stratégie marginale et la strastégie vectorielle. Nous évoquons ensuite l'impact de l'extension couleur sur la formulation des ZQP dans le cadre de la connexité des prédicats logiques. Enfin, nous discutons de l'utilisation de ces différentes extensions.

1.4.1 Pourquoi adapter les ZQP aux images couleur ?

Les définitions présentées dans la section précédente sont dédiées aux images à niveaux de gris où chaque pixel est représenté par une valeur scalaire. Elles ne peuvent donc pas être appliquées directement sur des images multibandes, telles que des images couleur ou des images multispectrales. Dans une image multivariée, chaque pixel n'est plus représenté par une valeur scalaire mais par une valeur vectorielle. L'information portée par l'image est plus riche : par exemple, dans une image couleur, chaque bande contient l'information relative à une certaine longueur d'onde. Les traitements appliqués sur des images en couleur fournissent alors des résultats plus précis. Ainsi, la figure 1.17 illustre la différence entre les ZQP extraites respectivement d'une image en couleur et de sa version en niveaux de gris, en conservant les mêmes paramètres.



a *b* *c*

FIGURE 1.17 – Comparaison de ZQP niveaux de gris et couleur : a) Image originale, b) α-$\mathcal{Z}$ en niveaux de gris avec $\alpha = 5$ (5433 ZQP), c) α-$\mathcal{Z}$ en couleur avec $\alpha = 5$ par la méthode de Zanoguera en utilisant la distance euclidienne dans l'espace RVB (23639 ZQP).

On remarque dans cette figure que les ZQP extraites d'une image couleur sont plus nombreuses que celles issues d'une image en niveaux de gris pour un α identique. La réaction en chaine qui apparait très vite en niveau de gris est plus limitée en couleur (pour une valeur α identique).

L'extension des approches de calcul des ZQP au cas des images multivariées n'est cependant pas triviale. En effet, de nombreuses définitions de ZQP ont besoin d'un ordre sur les valeurs des pixels. Cet ordre est trivial en niveaux de gris mais, dans les espaces multivariés, les valeurs ne sont pas ordonnées naturellement car vectorielles. Ce problème a déjà été abordé dans le cadre de la morphologie mathématique. Les solutions proposées ont été comparées par Aptoula et Lefèvre [AL07]. Ils distinguent 2 types d'approches. L'approche marginale qui consiste en le traitement de chaque bande indépendamment, ce qui permet d'utiliser les méthodes existantes en niveaux de gris. Cette approche n'est cependant pas adaptée à tous les problèmes et fait apparaître des couleurs qui n'existaient pas dans l'image d'origine. L'approche vectorielle permet de traiter les vecteurs directement mais nécessite de choisir un ordre vectoriel partiel ou total (selon les cas). Or il existe plusieurs ordres vectoriels partiels et totaux mais pas d'ordre meilleur de façon universelle : les performances de chaque ordre dépendent du problème considéré. Notons qu'en outre, le traitement morphologique d'images couleur peut nécessiter la définition d'une relation d'ordre sur des données angulaires. Ainsi, dans le cas des représentations polaires de la couleur, la teinte doit être traitée avec attention puisque d'une part sa nature est angulaire et non scalaire, et que d'autre part sa pertinence dépend du niveau de saturation considéré. Nous n'aborderons pas ces questions ici mais invitons le lecteur intéressé à consulter [AL09].

Dans ce chapitre, nous nous plaçons uniquement dans le cadre des images couleur. Cependant, sauf mention explicite, nous notons que les méthodes présentées peuvent également être appliquées

à d'autres types d'images multibandes telles que les images multispectrales rencontrées en imagerie astronomique ou en télédétection.

Nous allons à présent rappeler les approches existantes.

1.4.2 Rappel des approches existantes

Plusieurs auteurs ont proposé des adaptations des ZQP au contexte de la couleur, nous allons les présenter dans cette section. Les premières extensions ont seulement été développées pour l'α-$\mathcal{Z}$.

Zanoguera [Zan01] a proposé une extension où les paramètres sont toujours des scalaires. L'adaptation aux images couleurs se fait par l'utilisation d'une mesure de distance pour calculer la différence entre les valeurs de deux pixels voisins. Elle a étudié cette méthode dans quatre espaces couleurs : RVB, YUV, Lab et TSV. La distance utilisée a été la distance euclidienne (ou norme L_2) dont la formulation pour l'espace TSV est particulière. Ces distances sont calculées par les équations suivantes :

$$\begin{aligned}
d_{L_2}^{RVB}(p,q) &= \sqrt{(f^R(p)-f^R(q))^2+(f^G(p)-f^G(q))^2+(f^B(p)-f^B(q))^2} \\
d_{L_2}^{YUV}(p,q) &= \sqrt{(f^Y(p)-f^Y(q))^2+(f^U(p)-f^U(q))^2+(f^V(p)-f^V(q))^2} \\
d_{L_2}^{Lab}(p,q) &= \sqrt{(f^L(p)-f^L(q))^2+(f^a(p)-f^a(q))^2+(f^b(p)-f^b(q))^2} \\
d_{L_2}^{TSV}(p,q) &= \frac{1}{\sqrt{5}}\sqrt{(f^V(p)-f^V(q))^2+(\mathrm{dcossh}(p,q))^2+(\mathrm{ssinsh}(p,q))^2} \\
&\text{avec } \mathrm{dcossh}(p,q) = f^S(p)\cos f^T(p) - f^S(q)\cos f^T(q) \\
&\text{et } \mathrm{ssinsh}(p,q) = f^S(p)\sin f^T(p) - f^S(q)\sin f^T(q)
\end{aligned} \tag{1.22}$$

Les expériences menées par l'auteure ont montré que les espaces RVB et YUV donnaient de meilleurs résultats que les espaces perceptuellement uniformes Lab et TSV.

Angulo [AS03] a développé une extension couleur de l'α-$\mathcal{Z}$ adaptée à l'espace IHLS [Han03]. Il a dans un premier temps testé des approches simples comme la transformation de l'image dans l'espace IHLS en une image en niveaux de gris sur laquelle il a appliqué une segmentation par α-$\mathcal{Z}$. Il a également appliqué l'α-$\mathcal{Z}$ sur chaque bande pour voir si une seule bande pouvait suffir pour produire une segmentation intéressante. Constatant l'échec de ces deux stratégies, il a utilisé les propriétés de l'espace IHLS pour créer une nouvelle extension combinant les informations chromatiques et achromatiques issues de l'image. Pour ce faire il binarise la bande de saturation selon un seuil paramètrable afin d'obtenir l'image binaire $f^{S_{bin}}$ ($\overline{f^{S_{bin}}}$ désigne son complémentaire). Il applique ensuite l'α-$\mathcal{Z}$ indépendamment sur la teinte et la luminance et les combine dans la formule suivante :

$$\alpha\text{-}\mathcal{Z}_{Angulo} = (\alpha\text{-}\mathcal{Z}(f^H) \wedge f^{S_{bin}}) \vee (\alpha\text{-}\mathcal{Z}(f^L) \wedge \overline{f^{S_{bin}}}) \tag{1.23}$$

avec $\wedge$ l'opérateur de conjonction logique. La binarisation de la saturation a pour but de classer les pixels. Pour une valeur supérieure ou égale au seuil, les pixels apportent une information chromatique : on utilise alors l'α-$\mathcal{Z}$ calculée sur la teinte pour segmenter en ZQP ces pixels. A l'inverse, pour ceux qui n'ont aucune information chromatique, on utilise l'α-$\mathcal{Z}$ sur la luminance.

Soille [Soi08] a quant à lui proposé une extension couleur applicable à l'ensemble des ZQP. L'extension consiste en l'utilisation de paramètres vectoriels et non plus scalaires. En utilisant un α vectoriel, nous obtenons en réalité un paramètre α pour chaque bande de l'image. Dès lors, deux pixels connexes appartiennent à la même α-$\mathcal{Z}$ si et seulement si la différence de leurs valeurs dans chaque bande est inférieure ou égale à la valeur de α pour cette même bande. Ceci revient à redéfinir l'équation 1.9 définissant les chemins pour obtenir la définition suivante :

$$\forall i \in [0, n-2], p_i \in N(p_{i+1}) \text{ et } |f^j(p_i) - f^j(p_{i+1})| \leq \alpha^j, \forall\ j \in [1, b] \tag{1.24}$$

où b est le nombre de bandes dans l'image. La définition de l'α-$\mathcal{Z}$ reste par contre inchangée mais utilise l'équation 1.24 pour définir les chemins. Ce traitement marginal de la couleur souffre d'un défaut majeur. En effet, s'il permet de définir des α-$\mathcal{Z}$, il ne permet en aucun cas de définir des ZQP plus complexes comme les (α, ω)-$\mathcal{ZS}$. Les définitions ayant la propriété d'unicité nécessitent de disposer d'un ordre total sinon la condition 1.11 n'est pas vérifiée. Or, cette condition est nécessaire pour permettre de rechercher l'α-$\mathcal{Z}$ la plus grande. Quand l'α-$\mathcal{Z}$ ne convient pas, on décrémente α, ce qui n'est pas possible sans ordre. Pour contourner ce problème, Soille fixe comme contrainte que le paramètre α soit un vecteur ayant la même valeur dans chaque bande. Ainsi on peut aisément hiérarchiser les α et donc disposer d'un ordre total (la décrémentation de $\alpha = (3, 3, 3)$ donne $\alpha = (2, 2, 2)$). La variation globale est traitée comme la variation locale, la condition de variation globale n'étant vérifiée que si elle est vérifiée marginalement pour chaque bande.

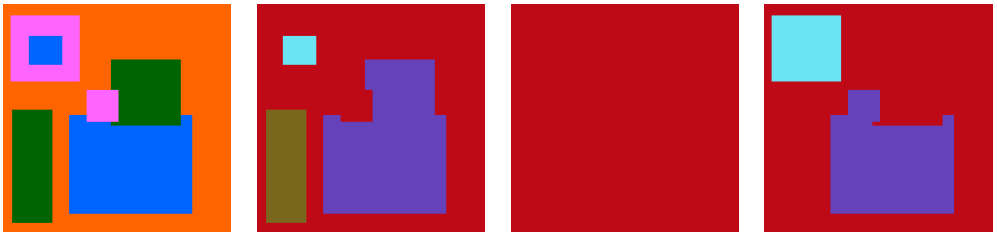
Il existe deux stratégies alternatives pour le traitement morphologique des images couleur : l'approche marginale et l'approche vectorielle. Nous allons maintenant explorer leur utilisation pour étendre les ZQP à la couleur.

1.4.3 Extension marginale

La stratégie marginale, de par sa simplicité, est fréquemment utilisée en analyse d'image couleur. Dans le cas du calcul des ZQP, l'application de cette stratégie revient à construire, pour chaque bande de l'image, la partition associée. Selon le contexte, il est possible d'utiliser les mêmes paramètres de segmentation (α, ω, etc) pour toutes les bandes ou de définir des valeurs spécifiques à chacune des bandes. La définition de l'α-$\mathcal{Z}$ devient alors un ensemble d'α-$\mathcal{Z}$ produit sur chaque bande :

$$\alpha\text{-}\mathcal{Z}(p) = \left\{ \begin{array}{l} \{p\} \cup \{Q^R \mid \forall q^r \in Q^R, \alpha\text{-}\mathcal{P}^R(p, q^r) \neq \emptyset\}, \\ \{p\} \cup \{Q^V \mid \forall q^v \in Q^V, \alpha\text{-}\mathcal{P}^V(p, q^v) \neq \emptyset\}, \\ \{p\} \cup \{Q^B \mid \forall q^b \in Q^B, \alpha\text{-}\mathcal{P}^B(p, q^b) \neq \emptyset\} \end{array} \right\} \tag{1.25}$$

L'inconvénient majeur de cette stratégie réside dans l'utilisation qui peut être faite du résultat. En effet, aucune garantie n'est donnée quant à la cohérence des partitions produites par l'analyse de chacune des bandes, comme l'illustre la figure 1.18. La combinaison des informations issues des différentes partitions est un problème complexe car, comme on le remarque sur la figure, aucune des partitions obtenues sur les trois différentes bandes ne contient la même information.



a *b* *c* *d*

FIGURE 1.18 – Approche marginale pour l'α-$\mathcal{Z}$ couleur ($\alpha = 15$) : a) Image originale, b) α-$\mathcal{Z}$ sur la bande rouge, c) α-$\mathcal{Z}$ sur la bande verte, d) α-$\mathcal{Z}$ sur la bande bleue.

Une exploitation concrète des ZQP nécessite donc de fusionner ces partitions afin de produire une unique partition en ZQP d'une image couleur. Différentes stratégies peuvent être appliquées, et nous distinguons ici deux cas de figure : soit la fusion ne tient compte que des partitions produites par la segmentation en ZQP, soit elle tient également compte de l'image originale.

Si on ne tient compte que des ZQP produites marginalement, on effectue la fusion en introduisant une nouvelle définition de ZQP. Les ZQP sont produites marginalement sur les n bandes de l'image ($n = 3$ pour une image couleur). Pour la fusion, nous considérons que deux pixels p et q appartiennent à la même ZQP s'ils appartiennent à la même ZQP dans au moins ι bandes (avec $1 \leqslant \iota \leqslant n$). On note ce type de ZQP par ($\iota_{M\text{-}\mathcal{Z}}$)-$\mathcal{ZF}$ ($\mathcal{ZF}$ pour zone fusionnée), où M-$\mathcal{Z}$ est la définition utilisée pour créer les ZQP marginales sur chaque bande (par exemple ($\iota_{\alpha\text{-}\mathcal{Z}}$)-$\mathcal{ZF}$). On les définit formellement par :

$$(\iota_{M\text{-}\mathcal{Z}})\text{-}\mathcal{ZF}(p) = \{p\} \bigcup \{Q | \forall q \in Q, \sum_{i=1}^{n} \overline{\mathcal{Z}}^i(p,q) \geqslant \iota\} \tag{1.26}$$

avec

$$\overline{\mathcal{Z}}(p,q) = \begin{cases} 1, \text{ si } q \in \mathcal{Z}(p) \\ 0, \text{ si } q \notin \mathcal{Z}(p) \end{cases} \tag{1.27}$$

La figure 1.19 présente l'application de l'(ι_{ZP})-$\mathcal{ZF}$, de l'($\iota_{\alpha\text{-}\mathcal{Z}}$)-$\mathcal{ZF}$ et de l'($\iota_{(\alpha,\omega)\text{-}\mathcal{ZS}}$)-$\mathcal{ZF}$ à l'image **macaws** avec différentes valeurs de ι.

On remarque qu'avec $\iota = 1$, on produit une sous-segmentation de certaines zones sans supprimer réellement la sur-segmentation dans d'autres, mis à part pour les zones plates où l'(ι_{ZP})-$\mathcal{ZF}$ produit des résultats intéressants et moins sur-segmentés que les zones plates en niveaux de gris. Pour $\iota = 2$, l'($\iota_{M\text{-}\mathcal{Z}}$)-$\mathcal{ZF}$ produit un nombre de régions proche de celui obtenu en niveaux de gris pour une qualité qui semble dépendre de la méthode utilisée marginalement. Tandis que pour $\iota = n$, l'($\iota_{M\text{-}\mathcal{Z}}$)-$\mathcal{ZF}$ produit un résultat de qualité (pas de sous-segmentation) mais avec un nombre de régions largement supérieur à celui obtenu en niveaux de gris. Ce résultat pourrait nous convaincre du peu d'intérêt de cette extension à la couleur. Cependant, ces résultats sont très variables et peut-être dépendants des paramètres (α, ω) que nous avons choisis. Nous avons donc effectué des tests avec différentes valeurs sur la base de Berkeley (cf. section 1.2.2). Les résultats de ces tests sont présentés dans les figures 1.20 et 1.21.

Cette expérience montre que, globalement, c'est-à-dire pour des paramètres de valeurs possiblement différentes, la fusion des ZQP marginales donne des résultats légèrement inférieurs aux ZQP en niveaux de gris en terme de rapport entre précision maximale et ratio de sur-segmentation. L'($\iota_{M\text{-}\mathcal{Z}}$)-$\mathcal{ZF}$, qui utilise uniquement la fusion de partitions en ZQP obtenues marginalement pour corréler les informations provenant de chaque bande, ne permet pas d'obtenir de meilleures ZQP qu'un traitement sur l'image en niveaux de gris. Cela est dû au fait que si l'information portée dans les différentes bandes de l'image est corrélée, la stratégie marginale ignore cette propriété. Seule l'étape de fusion permet de rattraper partiellement ce défaut. Or, nos images en niveaux de gris sont construites à partir des images couleurs de la base, elles sont donc le fruit d'une corrélation d'information, ce qui explique les meilleurs résultats obtenus en niveaux de gris.

Nous allons maintenant voir comment étendre les ZQP à la couleur par des approches vectorielles.

1.4.4 Extension vectorielle

La stratégie vectorielle consiste à considérer l'ensemble des bandes de l'image simultanément dans le calcul des ZQP. Ainsi, elle permet de tenir intrinsèquement compte de la corrélation entre les différentes bandes de l'image, mais aussi d'éviter l'étape de fusion des partitions discutée précédemment.

Pour construire les ZQP d'une image multivariée, où chaque pixel est représenté sous forme vectorielle, deux alternatives peuvent être suivies dans le cas de la stratégie vectorielle. Les paramètres de segmentation (α, ω, etc) peuvent être définis de façon scalaire ou vectorielle.

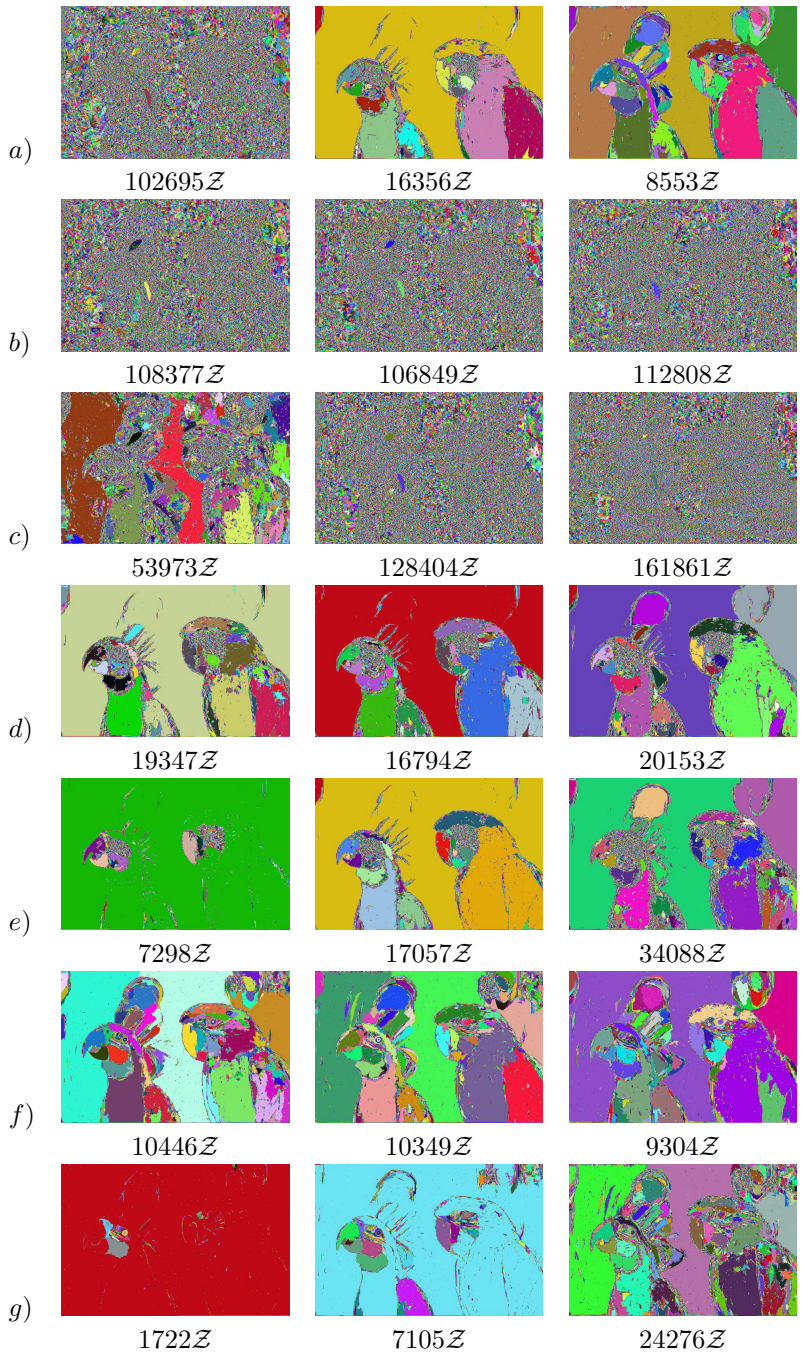


FIGURE 1.19 – Résultats de différentes approches marginales de ZQP : a) zones plates ndg, α-$\mathcal{Z}$ ndg, (α,ω)-$\mathcal{ZS}$ ndg , b) zones plates marginales sur R,V et B, c) (ι_{ZP})-$\mathcal{ZF}$ avec $\iota = 1, 2$ ou 3, d) α-$\mathcal{Z}$ marginal sur R,V et B e) $(\iota_{\alpha\text{-}\mathcal{Z}})$-$\mathcal{ZF}$ avec $\iota = 1, 2$ ou 3, f) (α,ω)-$\mathcal{ZS}$ marginal sur R,V et B, g) $(\iota_{(\alpha,\omega)\text{-}\mathcal{ZS}})$-$\mathcal{ZF}$ avec $\iota = 1, 2$ ou 3.

Si l'on considère uniquement des paramètres scalaires, on se place dans le cadre des définitions proposées par Zanoguera et Angulo présentées dans la section 1.4.2. Zanoguera a testé la distance L2 dans différents espaces et a obtenu les meilleurs résultats dans les espaces RVB et YUV. Nous avons voulu testé deux autres distances $L1$ et L_∞ également dans différents espaces afin de déterminer la combinaison optimale.

Les résultats de ces tests, pratiqués sur la base de Berkeley, sont présentés dans les figures 1.22

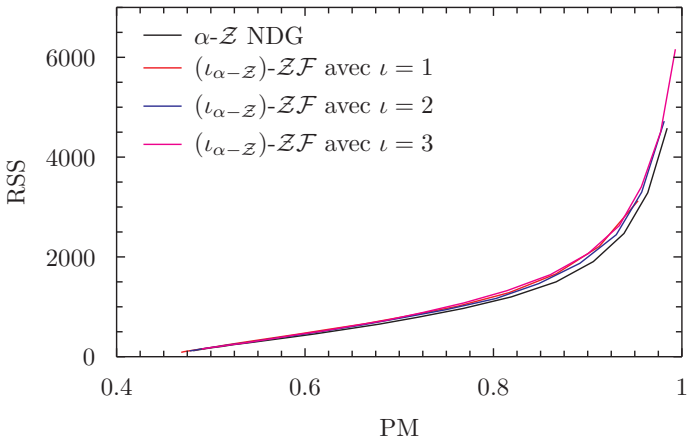


FIGURE 1.20 – Comparaison sur l'ensemble des images de la base de Berkeley pour différentes valeurs de α de l'α-$\mathcal{Z}$ en niveaux de gris et de $(\iota_{\alpha\text{-}\mathcal{Z}})$-$\mathcal{ZF}$ selon différentes valeurs de ι.

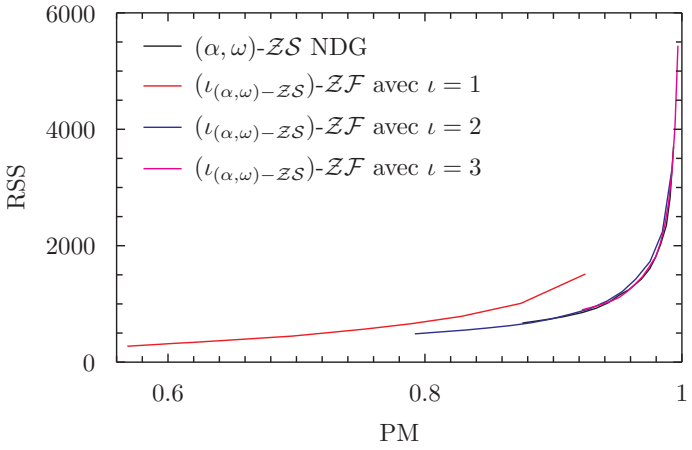


FIGURE 1.21 – Comparaison sur l'ensemble des images de la base de Berkeley pour différentes valeurs de α et ω de l'(α, ω)-$\mathcal{ZS}$ en niveaux de gris et de $(\iota_{(\alpha,\omega)\text{-}\mathcal{ZS}})$-$\mathcal{ZF}$.

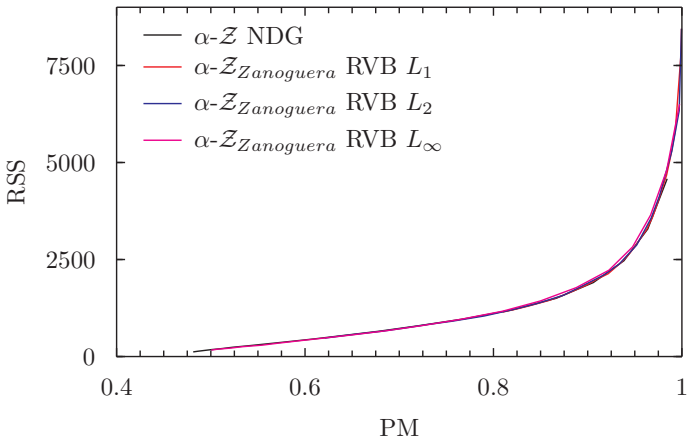


FIGURE 1.22 – Comparaison des distances sur l'ensemble des images de la base de Berkeley pour différentes valeurs de α pour l'α-$\mathcal{Z}_{Zanoguera}$.

et 1.23. Les expériences montrent que le type de distance n'a pas d'influence significative sur la qualité des résultats produits. Il en est de même pour l'espace couleur. De plus, cette extension

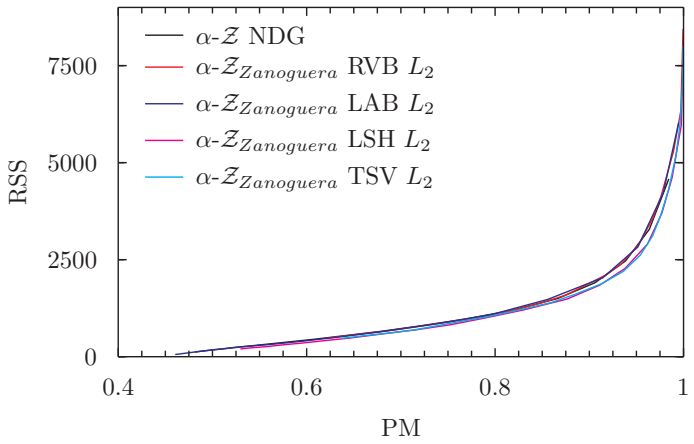


FIGURE 1.23 – Comparaison des espaces couleurs sur l'ensemble des images de la base de Berkeley pour différentes valeurs de α pour l'α-$\mathcal{Z}_{Zanoguera}$.

couleur donne des résultats de même qualité que ceux produits par l'application de l'α-$\mathcal{Z}$ en niveaux de gris sur une image couleur préalablement convertie en niveaux de gris. On peut en tirer deux observations possibles quant à l'intérêt de l'information chromatique sur l'information achromatique (niveaux de gris) : soit celle-ci n'est pas utilisée ici de façon efficace, soit elle ne permet tout simplement pas de résoudre le problème traité (la segmentation en zones quasi-plates) dans le cas particulier de la base de Berkeley.

Si elle produit des résultats comparables à ceux obtenus en niveaux de gris pour l'α-$\mathcal{Z}$, l'extension proposée par Zanoguera n'est pas adaptable à tous les prédicats, notamment au prédicat de la variation globale. En effet, le prédicat de variation globale vérifie que l'écart maximal entre deux valeurs de la ZQP est inférieur à un seuil. Son calcul est donc très simple en niveaux de gris, puisque les valeurs des pixels sont scalaires et ordonnées. Cependant, en couleur il n'y a plus d'ordre naturel entre les pixels, l'utilisation d'une distance comme le fait Zanoguera nécessiterait de comparer toutes les valeurs de pixel d'une ZQP deux à deux. Dès lors, le calcul de la variation globale aurait un coût calculatoire très important, ce qui la rendrait inutilisable.

La méthode proposée par Angulo utilise deux paramètres scalaires pour produire des ZQP couleur. Outre un paramètre α classique, elle nécessite une valeur de seuil pour la saturation (cf. équation 1.23). Ce deuxième paramètre rend plus complexe l'utilisation de cette méthode. Les tests pratiqués sur la base de Berkeley concernant la méthode d'Angulo sont présentés dans la figure 1.24. On observe que la valeur du seuil de saturation a une influence très importante sur les résultats. Si l'on compare cette méthode à l'α-$\mathcal{Z}$ en niveaux de gris, on observe que les résultats obtenus sont différents, contrairement à ce que l'on avait observé avec les autres extensions à la couleur. Plus le seuil de saturation est élevé, plus l'α-$\mathcal{Z}_{Angulo}$ tend à se rapprocher du comportement de l'α-$\mathcal{Z}$ en niveaux de gris (conformément à sa définition). Ainsi, comme l'α-$\mathcal{Z}_{Zanoguera}$ étudiée précédement, l'α-$\mathcal{Z}_{Angulo}$ ne permet pas d'obtenir de meilleurs résultats que l'α-$\mathcal{Z}$ en niveaux de gris.

Si l'on utilise des paramètres vectoriels, on peut utiliser un ordre marginal (qui est un ordre partiel). On ajoute alors un pixel à une ZQP uniquement s'il respecte l'ensemble des conditions définies sur toutes les bandes. C'est sur cette définition qu'est basée l'extension des ZQP à la couleur proposée par Soille (cf. section 1.4.2). Les résultats présentés dans la figure 1.25 ont été obtenus en utilisant des paramètres α vectoriels à valeurs identiques dans chacune des bandes. On remarque que l'α-$\mathcal{Z}$ couleur proposée par Soille produit des résultats très proches de l'α-$\mathcal{Z}$ en niveaux de gris comme les approches étudiées précédemment.

A l'inverse des approches précédentes, l'approche utilisée par l'α-$\mathcal{Z}_{Soille}$ permet l'utilisation du prédicat de variation globale. Dans ce cas, le prédicat est vérifié si la condition de variation globale

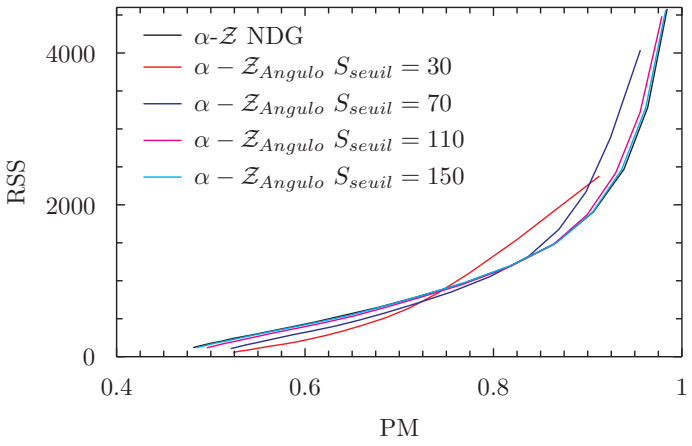


FIGURE 1.24 – Comparaison des résultats obtenus sur l'ensemble des images de la base de Berkeley pour différentes valeurs de α par l'α-$\mathcal{Z}$ en niveaux de gris et l'α-$\mathcal{Z}_{Angulo}$ selon différents seuils de saturation.

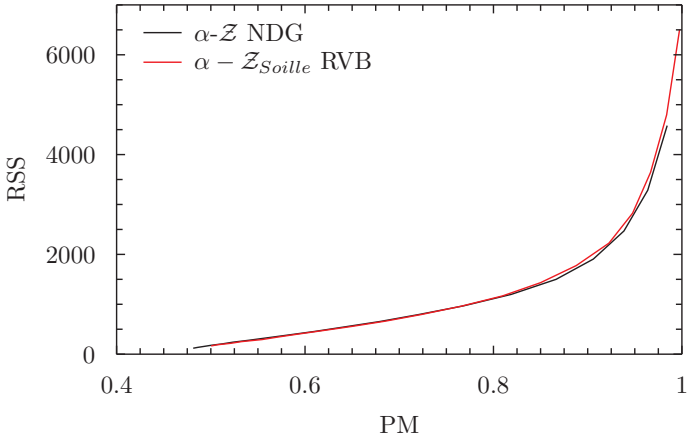


FIGURE 1.25 – Comparaison des résultats obtenus sur l'ensemble des images de la base de Berkeley pour différentes valeurs de α par l'α-$\mathcal{Z}$ en niveaux de gris et l'α-$\mathcal{Z}_{Soille}$.

est vérifiée marginalement dans chacune des bandes. La figure 1.26 compare l'(α, ω)-$\mathcal{ZS}$ en niveaux de gris et l'(α, ω)-$\mathcal{ZS}_{Soille}$ dans l'espace RVB. L'(α, ω)-$\mathcal{ZS}_{Soille}$ donne de meilleurs résultats que ceux obtenus en niveaux de gris. Cette méthode arrive à obtenir une précision élevée pour un ratio de sur-segmentation toujours élevé mais plus restreint qu'en niveaux de gris. L'information supplémentaire apportée par la couleur a été utilisée pour améliorer les résultats, contrairement aux méthodes vues précédemment.

Outre les propositions précédentes, il est possible d'utiliser directement des ordres vectoriels. Nous rappelons qu'un ordre est une relation binaire (ici entre deux vecteurs) notée $\preceq$ qui possède les propriétés de réflexivité ($\mathbf{x} \preceq \mathbf{x}$), de transitivité (($\mathbf{x} \preceq \mathbf{y}$ et $\mathbf{y} \preceq \mathbf{z}$) $\Rightarrow \mathbf{x} \preceq \mathbf{z}$) et d'anti-symétrie (($\mathbf{x} \preceq \mathbf{y}$ et $\mathbf{y} \preceq \mathbf{x}$) $\Rightarrow \mathbf{x} = \mathbf{y}$). Un préordre est quant à lui une relation binaire que l'on note également $\preceq$ mais qui ne possède que les propriétés de réflexivité et de transitivité. Dans un préordre, deux éléments peuvent être considérés comme égaux sans être identiques. Ces ordres ou préordres sont totaux quand tout élément est comparable avec tout autre élément d'un même ensemble ($\forall \mathbf{x}, \mathbf{y} \in V, \mathbf{x} \preceq \mathbf{y}$ ou $\mathbf{y} \preceq \mathbf{x}$).

Il existe de très nombreux ordres et préordres mais il n'existe pas un ordre ou préordre qui serait meilleur que tous les autres. Les performances d'un ordre vectoriel dépendent des images et de ce que souhaite l'utilisateur. Le choix de l'ordre vectoriel devient donc un paramètre particulièrement

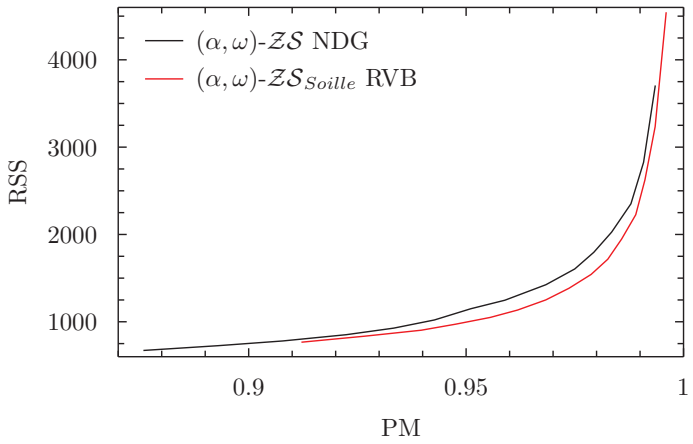


FIGURE 1.26 – Comparaison des résultats obtenues sur l'ensemble des images de la base de Berkeley pour différentes valeurs de α par l'α-$\mathcal{Z}$ en niveaux de gris et l'(α,ω)-$\mathcal{ZS}_{Soille}$.

difficile à maîtriser. L'utilisation d'ordres vectoriels pour produire des ZQP nécessite de modifier la définition des chemins α-connexes de l'équation 1.9 pour obtenir une définition basée sur les ordres vectoriels. Nous obtenons alors la définition : *Un chemin $\mathcal{P}$, selon un voisinage N, composé de n pixels $(p_0, p_1, ..., p_{n-1})$ est α-connexe selon l'ordre vectoriel $\preceq$ si et seulement si :*

$$\forall i \in [0, n-2], p_i \in N(p_{i-1}) \text{ et } \mathbf{d}_{\preceq}(\mathbf{f}(p_i), \mathbf{f}(p_{i-1})) \preceq \boldsymbol{\alpha} \tag{1.28}$$

où $\mathbf{d}_{\preceq}(\mathbf{a}, \mathbf{b})$ est un vecteur représentant la différence entre les vecteurs $\mathbf{a}$ et $\mathbf{b}$ selon l'ordre vectoriel $\preceq$. On note $(\alpha, \preceq)$-$\mathcal{P}(p, q)$, l'ensemble des chemins α-connexes entre p et q selon l'ordre vectoriel $\preceq$.

La définition de l'α-$\mathcal{Z}$ est alors modifiée pour devenir :

$$(\alpha, \preceq)\text{-}\mathcal{Z}(p) = \{p\} \cup \{Q \mid \forall q \in Q, (\alpha, \preceq)\text{-}\mathcal{P}(p, q) \neq \emptyset\} \tag{1.29}$$

Même si ces deux alternatives semblent différentes, il existe néanmoins des similitudes. Ainsi, l'approche proposée par Zanoguera (ZQP construites par comparaison des distances euclidiennes entre pixels à un paramètre α scalaire) peut être formulée comme une approche vectorielle. Considérons $\alpha = (\alpha\ 0\ 0)'$ la représentation vectorielle du paramètre α, et l'ordre vectoriel réduit $\preceq_E$ défini par $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $\mathbf{x} \preceq_E \mathbf{y} \Leftrightarrow \|\mathbf{x}\| \leq \|\mathbf{y}\|$ avec $\|\cdot\|$ la norme euclidienne. Avec cet ordre vectoriel, la différence entre deux vecteurs peut être définie par $\mathbf{d}_{\preceq_E}(\mathbf{a}, \mathbf{b}) = \mathbf{a} - \mathbf{b}$. Les ZQP de Zanoguera sont alors construites en assemblant tous les pixels voisins p et q vérifiant : $\mathbf{d}(f(p), f(q)) \preceq_E \alpha$.

Bien que l'on puisse définir formellement les ZQP dans le cadre d'une approche vectorielle, la mise en place d'une telle approche pose plusieurs problèmes :

- si fixer la valeur de α n'est pas trivial dans le cas de valeurs scalaires, fixer une valeur vectorielle de α est une tâche encore plus complexe ;
- l'α-connexité, principe de base des ZQP, est basée sur la comparaison de la différence entre les valeurs de deux pixels à un seuil. Dans le cadre d'une approche vectorielle, comment calculer la différence entre les valeurs de deux pixels ? L'utilisation de la différence bande par bande n'a pas de sens pour tous les ordres vectoriels. La différence devra donc être adaptée à l'ordre vectoriel utilisé ;
- certains prédicats sont difficiles à adapter au cadre vectoriel. Prenons l'exemple du prédicat de variation globale. Si l'on dispose d'un ordre vectoriel, on peut aisément obtenir la ou les valeurs les plus faibles ainsi que la ou les plus élevées au sens de l'ordre vectoriel utilisé. Cependant, comme ci-dessus, il est difficile de trouver une mesure pour l'écart $d_{\leq}$ adaptée à l'ordre vectoriel choisi. De plus, à l'instar de la valeur α, il est difficile de fixer un paramètre vectoriel pour ω.

Nous allons à présent reformuler ces extensions dans le cadre de la connexité des prédicats logiques.

1.4.5 Dans le cadre de la connexité des prédicats logiques

Les extensions possibles des ZQP aux images multivariées (et plus particulièrement couleur) ont été présentées dans le cadre de l'α-$\mathcal{Z}$ et de l'(α,ω)-$\mathcal{ZS}$. Nous allons dans cette section les replacer dans le cadre de la connexité des prédicats logiques afin de pouvoir y intégrer de futurs prédicats adaptés à la couleur.

La formulation de la connexité des prédicats logiques présentée dans l'équation 1.19 est toujours valable dans un cadre multivarié. Néanmoins, elle nécessite quelques adaptations. En premier lieu, si l'α-$\mathcal{Z}$ est définie de façon unique en niveaux de gris, il en est tout autrement en couleur. En effet, nous avons vu que la multidimensionnalité des données offre un certain nombre de possibilités et de variantes pour définir l'α-$\mathcal{Z}$. Le choix de l'α-$\mathcal{Z}$ à utiliser devient donc un paramètre à part entière et peut se formaliser sous la forme d'un prédicat de type particulier qui donnerait lieu à la notation suivante $(P_\alpha, P_1, ..., P_n)$-$\mathcal{Z}$ où P_α est le prédicat vérifiant si deux pixels sont α-connexes. Cette modification conduit à une reformulation de la connexité des prédicats logiques dans le cadre d'une adaptation à la couleur (et par généralisation aux images multivariées) en modifiant l'équation 1.19 pour obtenir la formulation suivante :

$$(P_\alpha, P_1, ..., P_n)\text{-}\mathcal{Z}(p) = \bigvee \left\{ \alpha'_{P_\alpha}\text{-}\mathcal{Z}(p) \;\middle|\; \begin{array}{r} \forall k \in \{1,...,n\} \quad P_k\left(\alpha'_{P_\alpha}\text{-}\mathcal{Z}(p)\right) = vrai \\ \forall \alpha'' \leq \alpha', \forall q \in \alpha'_{P_\alpha}\text{-}\mathcal{Z}(p), \quad P_k\left(\alpha''_{P_\alpha}\text{-}\mathcal{Z}(q)\right) = vrai \end{array} \right\} \tag{1.30}$$

où α_{P_α}-$\mathcal{Z}$ désigne l'α-$\mathcal{Z}$ construite en utilisant le prédicat d'α-connexité P_α. Elle est définie par :

$$\alpha_{P_\alpha}\text{-}\mathcal{Z}(p) = \{p\} \cup \{Q \mid \forall q \in Q, P_\alpha\text{-}\mathcal{P}(p,q) \neq \emptyset\} \tag{1.31}$$

où P_α-$\mathcal{P}(p,q)$ désigne l'ensemble des chemins α-connexes entre p et q selon le prédicat d'α-connexité P_α. Nous noterons $P_{\alpha_{Soille}}$ le prédicat permettant l'extension des ZQP à la couleur par la définition de Soille et $P_{\alpha_{Zanoguera}}$ celui utilisant la définition de Zanoguera.

L'extension des prédicats logiques aux images couleurs est dépendante de leur nature. Les prédicats basés sur des mesures purement géométriques (périmètre, aire,etc.) sont identiques. Des prédicats basés sur des descripteurs (texture, gradient,etc.) devraient utiliser des mesures adaptés au contexte couleur. Les prédicats utilisés dans les approches existantes, c'est-à-dire la variation globale ($P_i = \Omega$) et l'indice de connexité ($P_i = B$), présentent une adaptation différente à la couleur. L'extension de l'indice de connexité consiste à adapter l'α-connexité à la couleur. Cette adaptation modifie l'équation 1.15 et donne la formulation suivante :

$$B(\alpha_{P_\alpha}\text{-}\mathcal{Z}(p)) = \frac{\text{card}(\#_{\alpha_{P_\alpha}}(\alpha_{P_\alpha}\text{-}\mathcal{Z}(p)))}{\text{card}(\#(\alpha_{P_\alpha}\text{-}\mathcal{Z}(p)))} \tag{1.32}$$

La différence principale est que l'on n'utilise plus l'$\#_\alpha(X)$ mais l'$\#_{\alpha_{P_\alpha}}(X)$ qui est l'ensemble des relations d'adjacence de X α-connexes selon le prédicat d'α-connexité P_α. Il convient donc de choisir un prédicat d'α-connexité pour utiliser l'indice de connexité dans un contexte multivarié. La variation globale est plus complexe à adapter de par la notion d'écart maximal de valeurs à l'intérieur d'une ZQP. Comme nous en avons discuté dans les sections précedentes, la plupart des cadres proposés pour adapter les ZQP à la couleur ne permettent pas d'y intégrer la variation globale. Seul le cadre proposé par Soille permet de définir simplement la variation globale. D'autres cadres le permettent mais sont confrontés (en l'absence d'ordre total) à l'impossibilité d'obtenir des valeurs extrêmes au sein de la ZQP, et nécessitent alors la comparaison de tous les pixels deux à deux, ce qui rend le calcul très coûteux. En effet, la variation globale est la différence maximale entre tout couple de valeurs de la ZQP, ce qui correspond en niveaux de gris à la différence entre

la valeur minimale et la valeur maximale de la ZQP, ce n'est plus le cas dans le cadre vectoriel. Nous noterons Ω_{Soille} le prédicat de variation globale défini dans le cadre proposé par Soille pour l'extension des ZQP à la couleur.

Nous pouvons donc proposer deux versions de l'(α,ω)-$\mathcal{ZS}$ en couleur formulées selon la connexité des prédicats logiques. La $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ basée sur le prédicat d'α-connexité $P_{\alpha_{Zanoguera}}$ et la $(P_{\alpha_{Soille}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ basée sur le prédicat d'α-connexité $P_{\alpha_{Soille}}$.

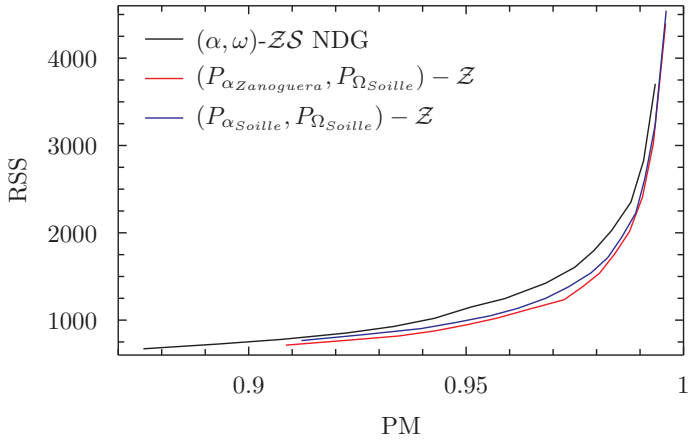


FIGURE 1.27 – Comparaison des résultats obtenus sur les images de la base de Berkeley pour les prédicats d'α-connexité $P_{\alpha_{Soille}}$ et $P_{\alpha_{Zanoguera}}$ pour la $(P_\alpha, P_1, ..., P_n)$-$\mathcal{Z}$ avec le prédicat Ω_{Soille} et de ceux obtenus pour l'(α,ω)-$\mathcal{ZS}$ en niveau de gris.

Nous avons comparé ces deux extensions de l'(α,ω)-$\mathcal{ZS}$, le résultat est présenté dans la figure 1.27. Les deux approches présentent des résultats similaires même si, pour certaines valeurs de précision, le prédicat $P_{\alpha_{Zanoguera}}$ donne légèrement de meilleurs résultats. Les deux approches produisent par contre de meilleurs résultats que l'(α,ω)-$\mathcal{ZS}$ en niveaux de gris donnant un intérêt à l'utilisation de la couleur.

Nous allons maintenant discuter de l'utilisation des différentes extensions couleur que nous avons évoquées jusqu'ici.

1.4.6 Discussion

Les résultats présentés dans les sections précédentes montrent que l'utilisation de la couleur ne permet pas d'obtenir de meilleurs résultats que ceux obtenus en niveaux de gris pour l'α-$\mathcal{Z}$, définition au coeur de la connexité des prédicats logiques. En effet, les extensions couleur de l'α-$\mathcal{Z}$ produisent des résultats très proches de ceux obtenus en niveaux de gris. On pourrait douter de l'intérêt des extensions existantes pour l'α-$\mathcal{Z}$, cependant l'approche couleur peut se révéler intéressante dans le cadre de la connexité des prédicats logiques que nous avons étendu à la couleur. En effet, l'utilisation du prédicat de variation globale en couleur produit de meilleurs résultats qu'en niveaux de gris.

Si un utilisateur ne souhaite utiliser que l'α-$\mathcal{Z}$, nous conseillons d'utiliser les traitements en niveaux de gris, la couleur n'apportant pas de meilleurs résultats mais augmentant la complexité dûe au plus grand nombre de données à traiter. Mais dans le cadre où nous nous plaçons, c'est-à-dire la connexité des prédicats logiques avec le prédicat de variation globale, la couleur apporte une plus-value au niveau des résultats.

Considérant les expériences menées sur l'α-$\mathcal{Z}_{Zanoguera}$ et concernant différents espaces couleur et différentes distances, le choix de l'espace couleur et de la distance ne semble pas avoir d'influence

sur la qualité du résultat produit. Nous produirons donc les ZQP dans l'espace RVB et utiliserons la distance L_2 si besoin.

Dans la suite, nous ne considérerons que la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ qui est la définition de ZQP couleur qui donne les meilleurs résultats, même s'ils sont très proches de ceux obtenus par la $(P_{\alpha_{Soille}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$.

Maintenant que nous disposons de ZQP adaptées à la couleur, nous pouvons envisager leur extension aux données vidéo.

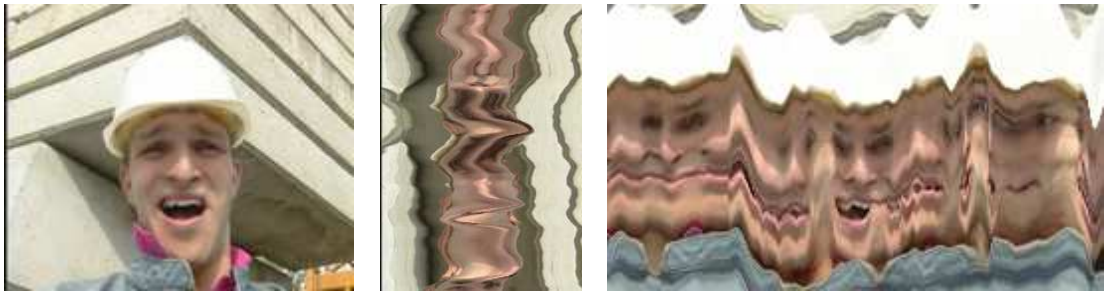
1.5 Extension vidéo

Dans cette section, nous étudions l'extension des ZQP aux données vidéo. Nous expliquons d'abord la nécessité de cette adaptation. Puis, nous abordons les problèmes posés par une extension directe des ZQP aux séquences vidéo. Nous posons ensuite l'extension des ZQP à la vidéo dans un cadre plus générique. Enfin nous discuterons de l'utilisation des différentes approches des ZQP vidéo.

1.5.1 Pourquoi adapter les ZQP aux séquences d'images ?

Traiter les séquences vidéo de la même manière que l'on traite les images fixes aurait peu de sens. En effet, cela conduirait à obtenir une segmentation par trame. Nous aurions donc une succession de segmentations au lieu d'obtenir une segmentation unique de la vidéo. De plus, un même objet dans des trames adjacentes se retrouverait dans différentes ZQP. Il est donc nécessaire d'adapter les définitions de ZQP aux séquences d'images afin de pouvoir segmenter de façon pertinente les séquences d'images en utilisant l'information apportée par la dimension temporelle.

Étant définis dans un espace tri-dimensionel composé de dimensions spatiales et temporelle, les pixels d'une séquence d'images présentent un voisinage spatio-temporel. L'intérêt principal des séquences vidéo est la persistance temporelle des objets qui s'y trouvent. Ceci permet le suivi d'un objet et l'extraction du fond. Ces deux points peuvent améliorer la segmentation. Cependant, le traitement de l'espace tri-dimensionnel des séquences vidéo n'est pas trivial. En effet, traiter des séquences vidéo a un coût calculatoire plus élevé que celui de traiter des images fixes dû au volume plus important des données. La sous et la sur-segmentation sont spatiales comme pour les images fixes mais également temporelles. Contrairement à une image volumique, les trois dimensions ne sont pas identiques. La dimension temporelle est différente des deux dimensions spatiales comme l'illustre la figure 1.28. Il sera donc difficile de traiter la dimension temporelle comme on traite une troisième dimension spatiale (comme la profondeur par exemple).



XY XT YT

FIGURE 1.28 – Séquence *foreman* représentée dans différentes dimensions.

1.5.2 Traiter une séquence d'image comme un volume spatio-temporel

L'extension la plus directe des ZQP aux séquences vidéo est de considérer une séquence vidéo comme un volume tri-dimensionnel spatio-temporel. On peut ainsi réutiliser les définitions existantes, la seule différence étant relative aux voisinages pris en compte qui sont spatio-temporels et non plus uniquement spatiaux.

Comme l'approche 3D implique seulement un voisinage différent, toutes les méthodes vues précédemment sont compatibles. Intéressons-nous au cas de l'α-$\mathcal{Z}$. Des exemples de résultats obtenus sur un extrait de la séquence *foreman* sont présentés figure 1.29. Ces résultats montrent que pour l'α-$\mathcal{Z}$, le passage de la 2D à la 3D entraine une sous-segmentation induite par la dimension temporelle. En effet, si l'on compare les résultats obtenus indépendamment sur chaque trame avec l'α-$\mathcal{Z}$ (fig 1.29.b) avec ceux obtenus avec les mêmes paramètres mais sur le volume spatio-temporel de la vidéo (fig 1.29.c), on remarque que des pixels n'appartenant pas à la même ZQP en 2D appartiennent à la même ZQP en 3D (ie. le casque de l'ouvrier et certaines parties du fond). Ce phénomène est du à la *réaction en chaîne* observée en $2D$ (cf. section 1.3.2) et amplifiée par la connectivité temporelle. La dimension temporelle donne naissance à des sortes de tunnels connectifs qui via le temps rendent α-connexes des pixels qui ne l'étaient pas spatialement. Par conséquent, l'α-$\mathcal{Z}$ est totalement inadaptée à un traitement $3D$ des séquences vidéo. En effet, même en utilisant des valeurs α très faibles comme dans la figure 1.29.c (où $\alpha = 2$), on obtient une importante sous-segmentation. Si pour une valeur aussi faible nous obtenons une sous-segmentation rendant inutilisable les résultats, nous ne pourrons obtenir que pire en augmentant les valeurs de α (cf. équation 1.11). Cette remarque est vérifiée sur la figure 1.30 où l'on observe que l'α-$\mathcal{Z}$ $3D$ produit à précision maximale égale une sur-segmentation plus importante que l'α-$\mathcal{Z}$ $2D$ qui présente pourtant une sur-segmentation temporelle extrême.

Si on applique la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ (cf figure 1.29.d et 1.29.e), on obtient une sur-segmentation spatiale beaucoup plus élevée que le même traitement en $2D$. La raison est qu'en traitant les volumes en 3D, le voisinage comporte plus de pixels et donc une α-$\mathcal{Z}$ comportera plus de pixels (cf. *réaction en chaîne* amplifiée par le tunnel connectif spatio-temporel discuté précédemment), ce qui augmente naturellement le risque de violer la contrainte de variation globale Ω_{Soille}. Ainsi, la plus grande α-$\mathcal{Z}$ satisfaisant la contrainte sera souvent celle produite avec une valeur de α faible. Ceci conduit à obtenir de très petites régions ne comportant que quelques pixels. La figure 1.30 montre ainsi qu'à l'instar de l'α-$\mathcal{Z}$, la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ $3D$ produit, à précision maximale égale, une sur-segmentation supérieure à la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ $2D$. Ainsi, la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ et plus généralement la connexité des prédicats logiques sont également inadaptées à l'approche $3D$.

Étendre les ZQP aux séquences vidéo en ne modifiant que le voisinage, peut sembler naturel. Cependant, le cadre actuel des ZQP est totalement inadapté à un traitement $3D$ des séquences vidéo.

1.5.3 Vers une approche incrémentale pour la construction des ZQP

Constatant les lacunes de l'approche $3D$ pour les ZQP vidéo, il nous apparait nécessaire de ne pas traiter les séquences vidéo comme un bloc spatio-temporel. Nous pourrions à l'instar de ce qui se fait dans d'autres approches, construire les ZQP sur la première trame et les propager aux trames suivantes, on dénomme ce type d'approche "$2D + t$". Bien que répandu, ce type d'approche peut s'avérer coûteux d'un point de vue calculatoire (notamment si l'on souhaite utiliser des informations de mouvement) et pose des problèmes lors de l'apparition de nouvelles régions ou d'occlusions de régions existantes. Dans cette section, nous proposons une nouvelle approche pour les ZQP. Elle consiste à traiter la séquence vidéo dans son intégralité mais selon diffèrents critères appliqués de façon successive. Un tel procédé conduit à une construction incrémentale des ZQP.

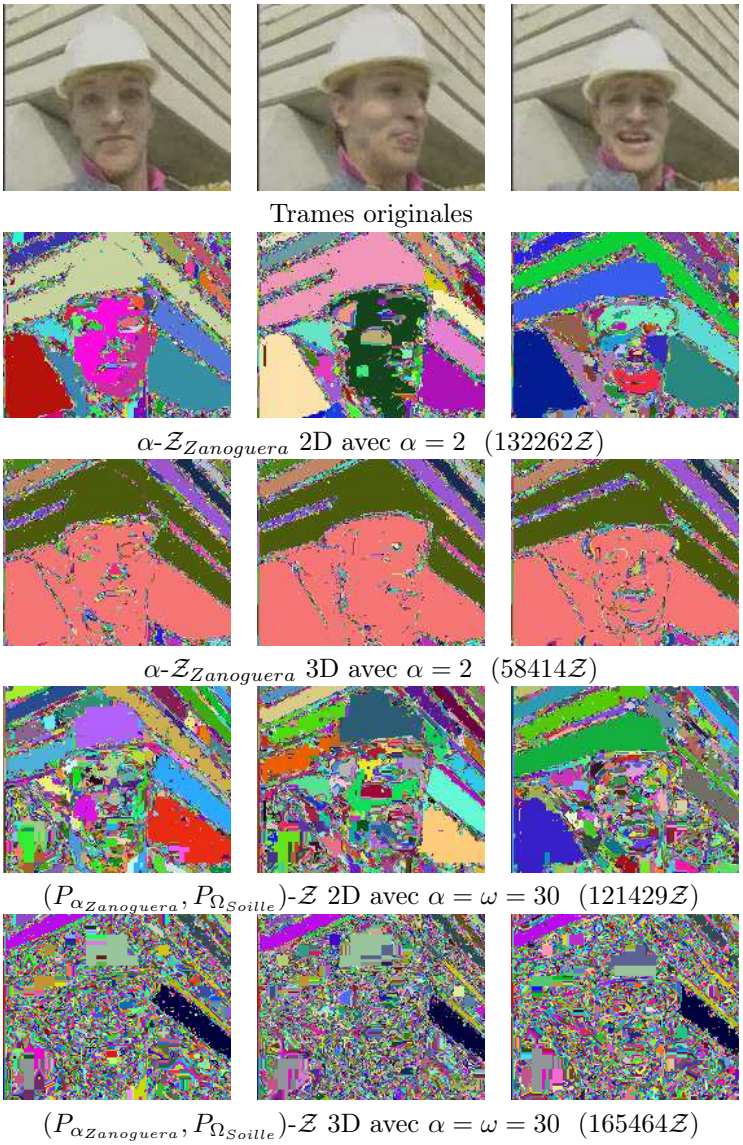


FIGURE 1.29 – Résultats de ZQP 3D sur un extrait de la séquence foreman (trame 91 à 120) et comparaison avec des ZQP 2D aux mêmes paramètres.

1.5.3.1 Construire les ZQP par incrément

Les ZQP, dans leur formulation actuelle de la connexité des prédicats logiques, permettent l'application de différents critères (par exemple variation locale, globale, connexité interne ...) en prenant en compte un certain voisinage. Cependant, tous les critères sont appliqués simultanément. Il en est de même pour le voisinage pris en compte, qui ne peut être subdivisé en voisinages plus restreints. Cela induit, lors de l'utilisation de plusieurs critères et d'un voisinage étendu, une importante sur-segmentation. En effet, plus on multiplie les critères plus on augmente le risque que l'ajout d'un pixel à la ZQP viole un critère : on risque donc de ne pouvoir construire une ZQP satisfaisant tous les critères qu'avec des valeurs faibles de α. Il en est de même pour le voisinage : plus un voisinage comporte de points, plus un de ces points est susceptible de violer un des critères. De plus, dans la formulation actuelle, le voisinage est identique pour tous les critères, il n'est par exemple pas possible de n'appliquer un critère que sur la partie spatiale d'un voisinage spatio-temporel. De la même manière, les critères sont appliqués de façon homogène dans tout le voisinage, il n'est pas possible d'utiliser, par exemple, des valeurs différentes de α selon les différentes parties du voisinage que l'on traite. Le cadre actuel de la connexité des prédicats logiques permet donc

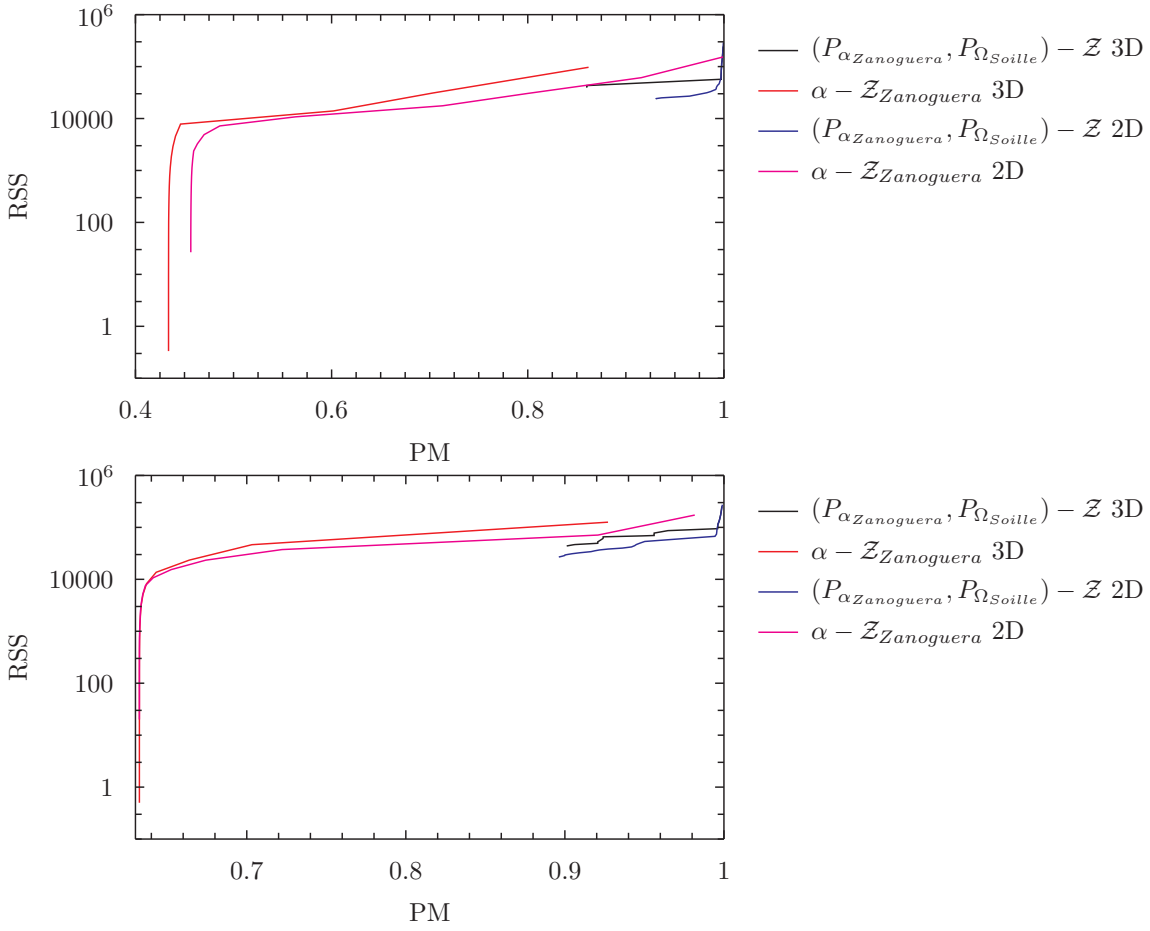


FIGURE 1.30 – Ratio de sur-segmentation et précision maximale par l'approche $3D$ de la $\alpha - \mathcal{Z}_{Zanoguera}$ et de la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ sur les séquences vidéo *carphone* (en haut) et *foreman* (en bas) selon différents paramètres α et ω.

l'utilisation de nombreux critères dans toutes sortes d'espace mais comporte des restrictions et un risque important de sur-segmentation.

Ce constat nous conduit à proposer un nouveau cadre pour la connexité des prédicats logiques. Ce cadre consiste à ne plus considérer les ZQP comme fruit d'une opération unique mais comme le résultat de l'application d'un ou plusieurs opérateurs successifs. Les ZQP consistent en une réduction de l'espace des données permettant d'obtenir des pièces de puzzle (ou superpixels). Il suffit ensuite d'assembler ces pièces pour obtenir les objets d'intérêt désirés par l'utilisateur. Nous effectuions jusque là cette réduction en utilisant la connexité des prédicats logiques. Nous allons continuer à utiliser cette connexité mais dans un schéma incrémental impliquant une réduction progressive de l'espace de données par l'application successive de la connexité des prédicats logiques selon différents critères et voisinages.

Le cadre que nous proposons est illustré dans la figure 1.31. La première étape consiste à transformer la séquence vidéo. Nous considérons dès lors les pixels de la séquence comme les nœuds d'un graphe. Une fois cette transformation effectuée, nous entrons dans la partie incrémentale de notre approche.

1. Description des nœuds du graphe : Cette étape consiste à décrire le ou les pixels représentés par le nœud. Cette description peut-être simple (p. ex. couleur ou couleur moyenne) ou plus complexe (p. ex. mouvement ou texture) ;
2. Ajout de la connexité : Pour l'instant, les nœuds ne sont pas reliés entre eux. Il nous faut donc

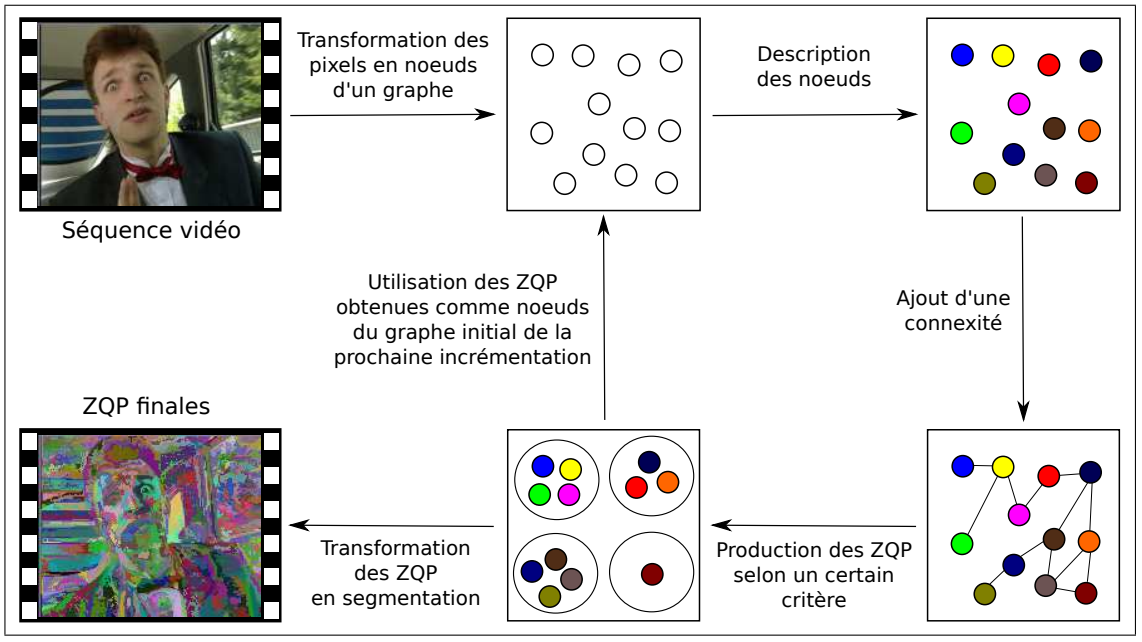


FIGURE 1.31 – Construction de ZQP par l'approche incrémentale.

ajouter une connexité afin de pouvoir produire des ZQP. La connexité peut-être basée sur des notions de voisinage simples (p. ex. 8-voisinage spatial ou 10-voisinage spatio-temporel) ou dépendre de concepts plus complexes (p. ex. vecteurs de mouvement) ;

3. Production des ZQP : Les ZQP sont produites en utilisant la connexité des prédicats logiques. Les prédicats utilisés doivent être adaptés à la ou les descriptions des nœuds du graphe. Les ZQP obtenues sont des fusions des nœuds du graphe ;
4. S'il y a encore d'autres applications de ZQP à effectuer on retourne à l'étape 1 en utilisant les ZQP produites comme nœuds du graphe initial, sinon on transforme les ZQP obtenues sur le graphe en segmentation des pixels de la séquence vidéo.

Cette application incrémentale de la connexité des prédicats logiques permet d'utiliser plusieurs types de connexité et prédicats logiques. Nous notons $A \rightarrow B$, l'application de la méthode B sur les ZQP produites par la méthode A. Par exemple dans le cadre d'un traitement séquentiel de données spatio-temporelles, nous pourrions avoir la $(P^{couleur}_{\alpha_{Zanoguera}}, P^{couleur}_{\Omega_{Soille}})$-$\mathcal{Z}_{2D} \rightarrow (P^{coulmoy}_{\alpha_{Zanoguera}}, P^{coulmoy}_{\Omega_{Soille}})$-$\mathcal{Z}_t$. Cette définition correspond à la production des ZQP spatialement indépendamment sur chaque trame par la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$. Puis on procède à la production de ZQP temporelles également par la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ basée sur les ZQP spatiales produites précédemment en utilisant leur couleur moyenne comme descripteur (*coulmoy* représente la couleur moyenne).

Si chaque application incrémentale des ZQP vérifie la propriété de hiérarchie 1.11, la combinaison des applications successives ne la vérifie pas. Cette perte de propriété est liée à l'utilisation d'un descripteur unique pour l'ensemble des pixels d'un nœud (p. ex. la couleur moyenne). Ainsi, deux pixels connexes qui n'aurait pas vérifié un prédicat, peuvent appartenir à deux nœuds différents connexes qui vérifient ce prédicat grâce à leurs descriptions. La perte de cette propriété n'est pas problématique car elle n'est utilisée que dans la construction d'une ZQP et non dans la combinaison de constructions incrémentales de ZQP.

Dans les deux sections suivantes, nous allons tester notre cadre pour l'approche incrémentale de ZQP sur les séquences vidéo. Nous allons étudier l'application d'un critère spatial puis d'un critère temporel et l'application d'un critère temporel puis d'un critère spatial. Par abus de notation, nous parlerons dans la suite de ce mémoire d'approche $2D + t$ pour les approches produisant d'abord les ZQP spatialement puis temporellement et d'approche $t + 2D$ pour les approches produisant

d'abord les ZQP temporellement puis spatialement. Dans la suite nous utilisons principalement les approches suivantes :

- Approche $2D + t$
  - $\alpha^{couleur}$-$\mathcal{Z}_{Zanoguera_{2D}} \rightarrow \alpha^{coulmoy}$-$\mathcal{Z}_{Zanoguera_t}$
  - $(P^{couleur}_{\alpha_{Zanoguera}}, P^{couleur}_{\Omega_{Soille}})$-$\mathcal{Z}_{2D} \rightarrow (P^{coulmoy}_{\alpha_{Zanoguera}}, P^{coulmoy}_{\Omega_{Soille}})$-$\mathcal{Z}_t$
- Approche $t + 2D$
  - $\alpha^{couleur}$-$\mathcal{Z}_{Zanoguera_t} \rightarrow \alpha^{coulmoy}$-$\mathcal{Z}_{Zanoguera_{2D}}$
  - $(P^{couleur}_{\alpha_{Zanoguera}}, P^{couleur}_{\Omega_{Soille}})$-$\mathcal{Z}_t \rightarrow (P^{coulmoy}_{\alpha_{Zanoguera}}, P^{coulmoy}_{\Omega_{Soille}})$-$\mathcal{Z}_{2D}$

1.5.3.2 Application d'un critère spatial puis d'un critère temporel

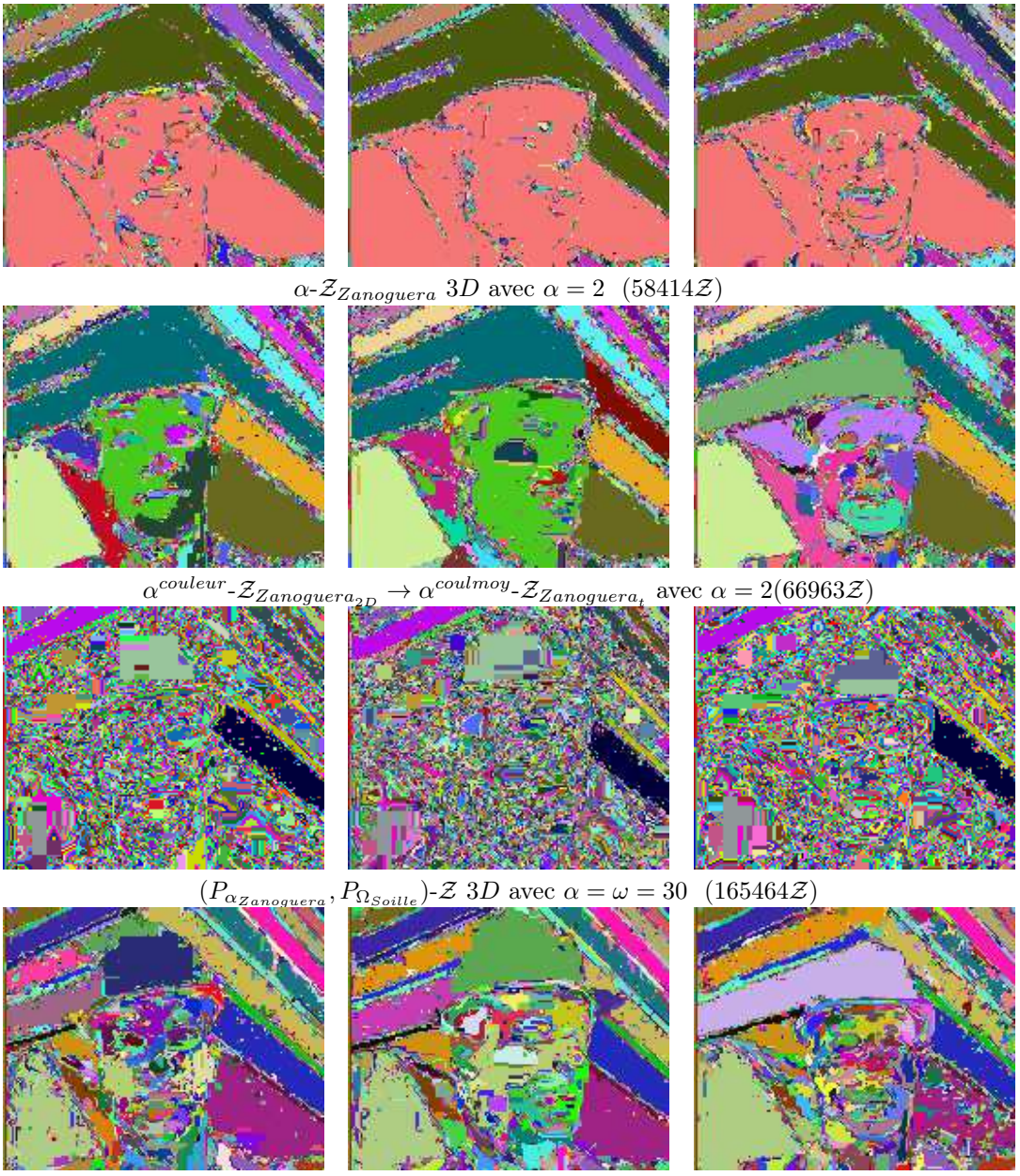
L'approche $2D + t$, définie dans le cadre incrémental, consiste à produire les ZQP spatialement puis à les étendre temporellement tout en gardant les règles de production des ZQP. Elle commence la production des ZQP sur chaque trame indépendamment. Il en résulte une sur-segmentation temporelle extrême puisque chaque ZQP n'est présente que dans une seule et unique trame. Nous obtenons donc un grand nombre de zones quasi-plates spatiales mono-trame qui représentent des régions spatiales quasi-homogènes et vont constituer les données sur lesquelles nous allons produire temporellement les ZQP. Pour ce faire, nous considérons chaque ZQP spatiale non plus comme un ensemble de pixels mais comme un seul et unique objet valué. Nous décrivons chaque objet par la couleur moyenne des pixels qui le composent. Afin de pouvoir produire les nouvelles ZQP, il nous faut définir un voisinage adapté à cette réduction de données. Nous considérons que deux ZQP sont adjacentes si et seulement si, elles sont situées dans des trames adjacentes et elles ont au moins un pixel ayant des coordonnées spatiales communes. On peut définir l'adjacence plus formellement comme :

$$\begin{aligned} &\mathcal{Z}_1 \text{ et } \mathcal{Z}_2 \text{ adjacentes si et seulement si} \\ &\qquad (|T(\mathcal{Z}_1) - T(\mathcal{Z}_2)| = 1) \quad \wedge \quad (\mathcal{D}_{2D}(\mathcal{Z}_1) \cap \mathcal{D}_{2D}(\mathcal{Z}_2) \neq \emptyset) \end{aligned} \tag{1.33}$$

où $T(\mathcal{Z}_1)$ est la coordonnée temporelle de la trame qui contient $\mathcal{Z}_1$ et $\mathcal{D}_{2D}(\mathcal{Z}_1)$ le domaine spatial de $\mathcal{Z}_1$. Nous pouvons alors produire temporellement les ZQP. Notons qu'il est possible que cette production temporelle crée des ZQP contenant deux ZQP de la même trame spatialement connexes mais qui n'avaient pas donné une unique ZQP lors du traitement $2D$. Il s'agit d'un effet du tunnel connectif spatio-temporel.

La figure 1.32 présente une comparaison des résultats obtenus par les approches $3D$ et $2D + t$. On constate que pour l'α-$\mathcal{Z}_{Zanoguera}$, l'approche $2D + t$ produit plus de zones mais réduit la sous-segmentation spatiale que nous avions observé dans la section 1.5.2. A l'inverse, pour la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$, elle produit nettement moins de régions en $2D + t$, réduisant ainsi l'extrême sur-segmentation que nous avions constaté dans la section précédente. La figure 1.33 compare les ratios de sur-segmentation obtenus par les approches $2D+t$ et $3D$ à précision maximale égale sur les séquences *foreman* et *carphone*. Cette comparaison quantitative confirme ce que nous constations visuellement sur la figure 1.32, c'est-à-dire que l'approche $2D+t$ produit une sur-segmentation inférieure à l'approche $3D$.

L'obtention de meilleurs résultats en $2D + t$ qu'en $3D$ s'explique par le traitement différent des deux types de dimensions. En $3D$, les dimensions spatiales et temporelles sont mélangées et traitées comme un seul bloc ce qui accroit la réaction en chaîne pour l'α-$\mathcal{Z}$: des zones non-connexes spatialement appartiennent alors à la même zone plate puisqu'elles sont reliées par un tunnel connectif spatio-temporel. Ce phénomène entraine une sous-segmentation pour l'α-$\mathcal{Z}$ qui déclenche une sur-segmentation pour la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$. Cette sur-segmentation est dûe à la violation de la contrainte de variation globale qui n'est vérifiée que pour des valeurs de α très faible à cause de la sous-segmentation de l'α-$\mathcal{Z}$. En traitant en premier lieu uniquement l'aspect spatial, on minimise la réaction en chaîne, ceci donne des α-$\mathcal{Z}$ plus homogènes car moins sujettes à la sous-segmentation. Ces α-$\mathcal{Z}$ plus homogènes violent, par conséquent, moins souvent la contrainte de variation globale. Ce qui permet d'obtenir des $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ plus étendues spatialement, et donc de réduire



α-$\mathcal{Z}_{Zanoguera}$ $3D$ avec $\alpha = 2$ $(58414\mathcal{Z})$

$\alpha^{couleur}$-$\mathcal{Z}_{Zanoguera_{2D}} \rightarrow \alpha^{coulmoy}$-$\mathcal{Z}_{Zanoguera_t}$ avec $\alpha = 2(66963\mathcal{Z})$

$(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ $3D$ avec $\alpha = \omega = 30$ $(165464\mathcal{Z})$

$(P^{couleur}_{\alpha_{Zanoguera}}, P^{couleur}_{\Omega_{Soille}})$-$\mathcal{Z}_{2D} \rightarrow (P^{coulmoy}_{\alpha_{Zanoguera}}, P^{coulmoy}_{\Omega_{Soille}})$-$\mathcal{Z}_t$ avec $\alpha = \omega = 30(40444\mathcal{Z})$

FIGURE 1.32 – Comparaison des approches $3D$ et $2D + t$.

la sur-segmentation spatiale très importante observée dans l'approche $3D$. Le traitement de la dimension temporelle effectué sur la valeur moyenne des ZQP et non sur la valeur de chaque pixel limite également la réaction en chaîne de l'α-$\mathcal{Z}$. En effet, deux pixels voisins temporellement dont la différence de valeurs est inférieure à α peuvent appartenir à des ZQP dont la différence de valeur moyenne est supérieure à α. Cette situation empêche leur fusion et limite les effets du tunnel connectif spatio-temporel, notamment sur les zones ayant des effets de dégradés. Cette limitation du tunnel entraine pour l'$(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ des limitations dans la réduction de la sur-segmentation temporelle. En effet, pour des ZQP temporellement adjacentes et très homogènes, la sur-segmentation temporelle sera réduite efficacement. Mais, les ZQP représentant des régions d'un même objet mais ayant selon les trames un nombre de pixels trop différents induisant alors une différence de valeurs moyennes trop grande peut provoquer une sur-segmentation temporelle (cf. figure 1.32.d, le casque de l'ouvrier appartient à une classe différente dans chaque trame présentée

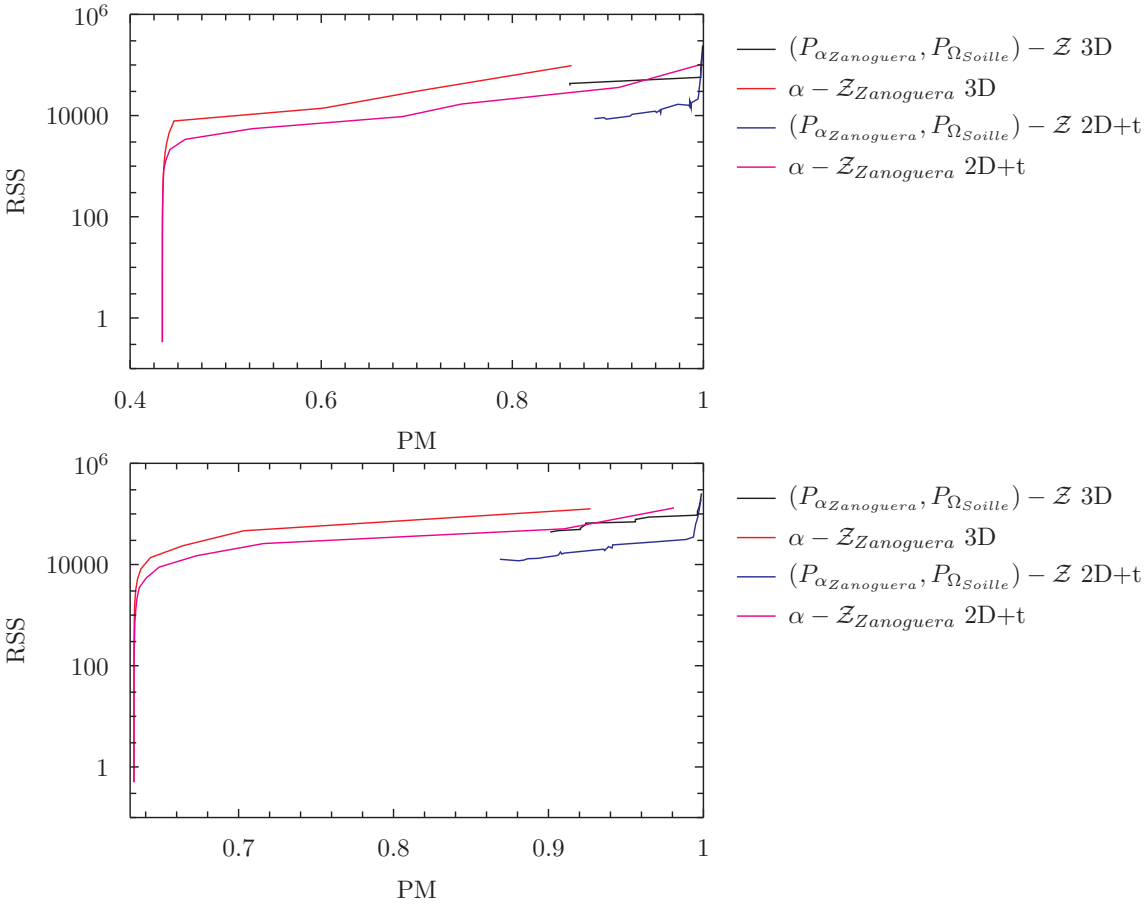


FIGURE 1.33 – Comparaison des approches $2D + t$ et $3D$ pour la $\alpha - \mathcal{Z}_{Zanoguera}$ et la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ sur les séquences vidéo *carphone* (en haut) et *foreman* (en bas) selon différents paramètres α et ω.

notamment parce que les ZQP ont été mal produites spatialement entrainant une trop grosse hétérogénéité et donc des moyennes de couleurs trop éloignées pour regrouper les ZQP spatiales lors du traitement temporel). Cette sur-segmentation temporelle est dûe à la segmentation spatiale en ZQP initiale. L'application de la dimension temporelle réduit la sur-segmentation spatiale mais ne peut réduire la sous-segmentation spatiale. En effet, le traitement de la dimension temporelle ne resegmente pas les ZQP obtenues spatialement, il ne fait que les fusionner. Si les frontières entre certains objets sont difficiles à définir d'un point de vue de différence de valeurs de pixels, la segmentation spatiale initiale en ZQP risque d'échouer (comme dans la figure 1.32.d où la frontière entre le casque de l'ouvrier et le mur n'est pas toujours évidente).

1.5.3.3 Application d'un critère temporel puis d'un critère spatial

L'approche temporelle puis spatiale consiste à produire les ZQP pour chaque coordonnée spatiale indépendamment selon la dimension temporelle. On cherche ensuite à les étendre spatialement tout en gardant les règles de production des ZQP. Il s'agit donc de produire en premier lieu des ZQP en utilisant uniquement la dimension temporelle, ce qui produit une sur-segmentation spatiale extrême puisque pour chaque trame, chaque pixel appartient à une ZQP différente. Nous obtenons donc un grand nombre de zones quasi-plates temporelles mono-coordonées spatiales qui représentent deux types de pixels : des pixels appartenant à des régions statiques dont les valeurs n'évoluent pas au cours du temps et des pixels appartenant à des régions dynamiques mais homogènes dont les valeurs sont proches même s'ils ne représentent pas le même endroit de la région. L'étape temporelle consiste donc à produire des ZQP représentant un pixel et son évolution. Le

traitement spatial va donc avoir pour but de regrouper les ZQP temporelles spatialement adjacentes et similaires afin d'obtenir des volumes homogènes et non plus seulement des évolutions temporelles de pixels homogènes. A l'instar de ce que nous avons fait pour l'approche $2D+t$, nous ne considérons plus les pixels mais les ZQP temporelles comme élément unitaire, nous permettant de réduire l'espace des données. Nous les décrivons par la couleur moyenne des pixels qui les composent. Afin de pouvoir produire les ZQP spatio-temporelles, il nous faut définir un voisinage. Nous décidons que deux ZQP sont adjacentes si et seulement si, leurs coordonnées spatiales sont adjacentes et qu'elles contiennent au moins un pixel ayant des coordonnées temporelles communes. Cette adjacence est définie plus formellement :

$$\begin{array}{c} \mathcal{Z}_1 \text{ et } \mathcal{Z}_2 \text{ adjacentes si et seulement si} \\ (|X(\mathcal{Z}_1) - X(\mathcal{Z}_2)| \leq 1) \quad \wedge \quad (|Y(\mathcal{Z}_1) - Y(\mathcal{Z}_2)| \leq 1) \quad \wedge \quad (\mathcal{D}_t(\mathcal{Z}_1) \cap \mathcal{D}_t(\mathcal{Z}_2) \neq \emptyset) \end{array} \tag{1.34}$$

où $X(\mathcal{Z}_1)$ est la coordonnée spatiale x des pixels de $\mathcal{Z}_1$, $Y(\mathcal{Z}_1)$ est la coordonnée spatiale y des pixels de $\mathcal{Z}_1$ et $\mathcal{D}_t(\mathcal{Z}_1)$ le domaine temporel de $\mathcal{Z}_1$. Nous pouvons alors produire spatialement les ZQP. Il est possible que cette production spatiale créé des ZQP contenant des pixels ayant la même coordonnée spatiale mais qui n'étaient pas dans la même ZQP à l'issue du traitement temporel. Il s'agit, comme en $2D+t$, d'un effet du tunnel connectif spatio-temporel.

La figure 1.34 présente une comparaison des résultats obtenus pour les approches $3D$ et $t+2D$. On constate que pour l'α-$\mathcal{Z}_{Zanoguera}$, l'aproche $t+2D$ produit plus de zones mais ne réduit pas la sous-segmentation que nous avions observée dans la section 1.5.2 contrairement à l'approche $2D+t$. L'aproche $t+2D$ est plus efficace pour la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$, elle produit nettement moins de régions, même que l'approche $2D+t$. Elle réduit ainsi l'extrême sur-segmentation de l'approche $3D$ que nous avions constatée dans la section 1.5.2. Cette observation visuelle est confirmée par la figure 1.35 qui montre qu'à précision maximale égale, l'approche $t+2D$ produit une sur-segmentation plus faible que l'approche $3D$.

Les mauvais résultats obtenus pour l'α-$\mathcal{Z}$ peuvent être expliqués par le fait que l'approche $t+2D$ est plus sensible aux effets du tunnel spatio-temporel que l'approche $2D+t$. Cela est lié au traitement initial de la dimension temporelle qui provoque une réaction en chaîne purement temporelle et possiblement extrême (tous les pixels d'une coordonnée spatiale dans la même ZQP temporelle). Le traitement de la dimension spatiale est effectué sur la couleur moyenne des ZQP. L'utilisation de valeurs lissées rend possible la fusion de ZQP dont les pixels voisins violent la contrainte α mais dont les couleurs moyennes ne la violent pas (à condition qu'il y ait un nombre suffisamment important de pixels dans la ZQP pour diminuer l'influence, sur le calcul de la valeur moyenne, des pixels voisins violant la contrainte α). Ce phénomène n'existait pas en $2D+t$, car le premier traitement était spatial et construisait des ZQP comportant plus de pixels hétérogènes, phénomène du à la *réaction en chaîne* qui est bi-dimensionnelle dans le traitement spatial. Les moyennes obtenues étaient donc plus différentes car basées sur des ZQP plus hétérogènes (malgré l'effet de lissage de la moyenne). A l'inverse, pour la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$, l'approche $t+2D$ se révèle plus efficace en terme de réduction de sur-segmentation temporelle et de sous-segmentation spatiale que l'approche $2D+t$. La différence s'explique par le fait qu'en $t+2D$, la sur-segmentation temporelle est réduite. En effet, dans un premier temps on crée des ZQP purement temporelles afin que tous les pixels temporellement connexes d'une même coordonnée spatiale appartenant à la même région homogène au cours de la vidéo (par exemple un morceau de mur) appartiennent à la même ZQP. Nous obtenons donc une sur-segmentation temporelle très limitée en $t+2D$ alors qu'en $2D+t$ elle était très élevée car cette dimension était traitée secondairement. Concernant la sous-segmentation spatiale observée en $2D+t$, elle est nettement réduite en $t+2D$. Cette différence est due au premier traitement temporel. En effet, le traitement spatial n'est pas effectué sur les valeurs de pixels mais sur les valeurs moyennes des ZQP. L'utilisation de la valeur moyenne réduit l'effet de la réaction en chaîne en introduisant des paliers plus forts dans les zones de transition.

α-$\mathcal{Z}_{Zanoguera}$ $3D$ avec $\alpha = 2$ (58414 $\mathcal{Z}$)

$\alpha^{couleur}$-$\mathcal{Z}_{Zanoguera_t} \rightarrow \alpha^{coulmoy}$-$\mathcal{Z}_{Zanoguera_{2D}}$ avec $\alpha = 2$ (74118 $\mathcal{Z}$)

$(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ $3D$ avec $\alpha = \omega = 30$ (165464 $\mathcal{Z}$)

$(P^{couleur}_{\alpha_{Zanoguera}}, P^{couleur}_{\Omega_{Soille}})$-$\mathcal{Z}_t \rightarrow (P^{coulmoy}_{\alpha_{Zanoguera}}, P^{coulmoy}_{\Omega_{Soille}})$-$\mathcal{Z}_{2D}$ avec $\alpha = \omega = 30$ (16135 $\mathcal{Z}$)

FIGURE 1.34 – Comparaison des approches $3D$ et $t + 2D$.

1.5.4 Discussion

Dans cette section nous ne discuterons des ZQP qu'à travers la $(P_\alpha, P_{\Omega_{Soille}})$-$\mathcal{Z}$, l'$\alpha$-$\mathcal{Z}$ n'étant que la base de la connexité des prédicats logique et n'est pas utilisée en soi pour produire des ZQP du fait de sa trop grande sensibilité à la réaction en chaîne. Les approches traitant séparément les dimensions spatiales et temporelle donnent de meilleurs résultats que l'approche $3D$ génératrice d'une très importante sur-segmentation. De par l'ordre dans lequel elles traitent les dimensions, les approches $2D + t$ et $t + 2D$ induisent des sur-segmentations différentes. L'approche $2D + t$ provoque une sur-segmentation spatiale réduite mais une forte sur-segmentation temporelle. À l'inverse, l'approche $t + 2D$ produit une sur-segmentation spatiale forte mais une sur-segmentation temporelle réduite. Les deux approches produisent néanmoins des résultats de meilleure qualité que l'approche $3D$. Le choix de l'une ou l'autre des approches sera dépendant de la vidéo à traiter.

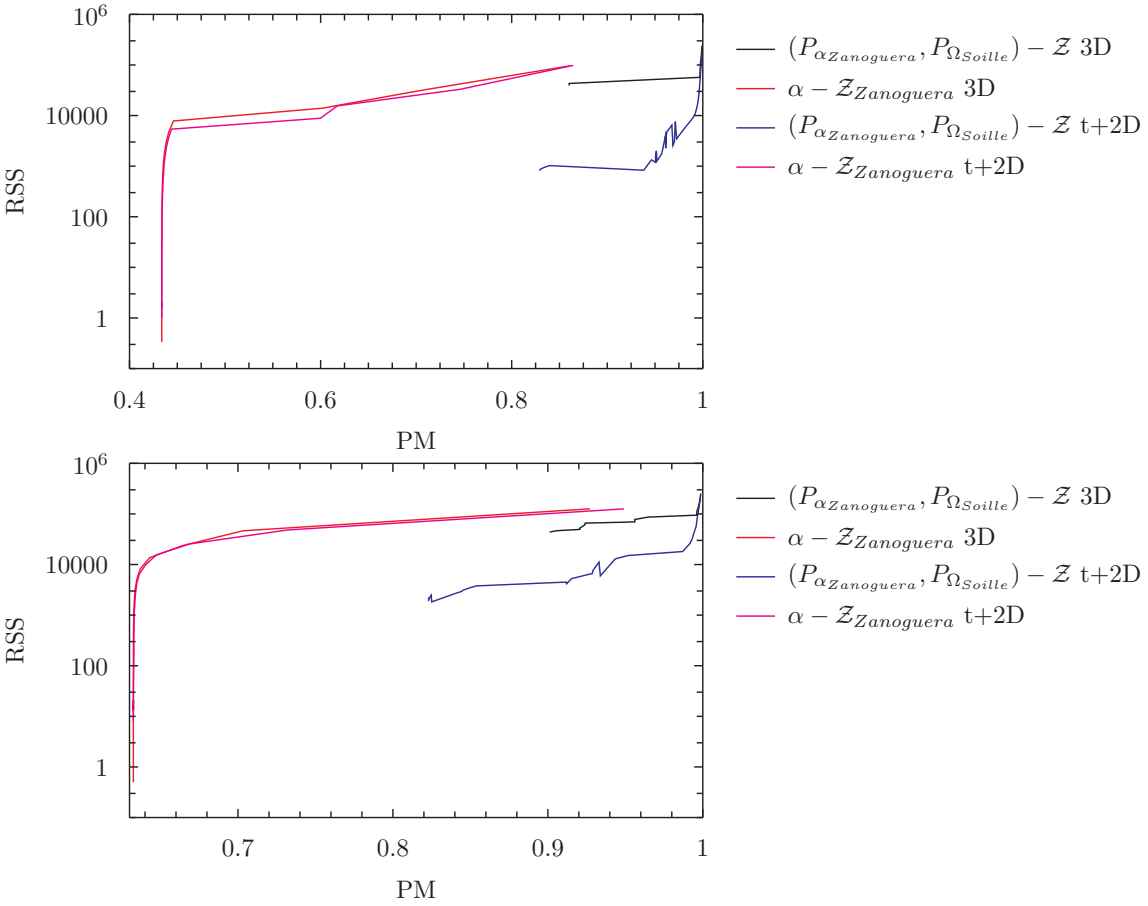


FIGURE 1.35 – Comparaison des approches $t + 2D$ et $3D$ pour la $\alpha - \mathcal{Z}_{Zanoguera}$ et la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ sur les séquences vidéo *carphone* (en haut) et *foreman* (en bas) selon différents paramètres α et ω.

On en déduit qu'il sera préférable de traiter avec l'approche $2D + t$ des séquences vidéo de grande taille en termes de résolution et courtes en terme de durée, tandis qu'il sera préférable de traiter les séquences de résolution plus faibles et de plus longue durée avec l'approche $t + 2D$. Cependant, en appliquant les différentes approches sur les séquences *carphone* et *foreman* et en les évaluant par rapport à une segmentation de référence (cf figure 1.36), nous constatons que l'approche $t + 2D$ donne des résultats nettement plus intéressants en termes de rapport précision maximale/ratio de sur-segmentation. Sur cette figure, nous avons également présenté les résultats obtenus en rajoutant un traitement dimensionnel supplémentaire, c'est-à-dire en faisant du $2D + t + 2D$ et du $t + 2D + t$ afin de voir si nous arrivions à diminuer la sur-segmentation tout en gardant une bonne précision maximale. L'intérêt de retraiter le premier type de dimension traitée (spatiale ou temporelle) est d'appliquer ce traitement non pas sur les pixels comme lors du premier traitement mais sur la valeur moyenne des ZQP qui représente une réduction de l'espace de données. Il sera donc possible de fusionner au sein d'une même ZQP des pixels qui ne l'étaient pas lors du premier traitement de ce type de dimension et ainsi de réduire la sur-segmentation. Les résultats montrent que si le gain est important pour la $2D + t + 2D$, il est négligeable pour la $t + 2D + t$. Cela s'explique par le fait que si la sur-segmentation spatiale est plus réduite en $2D + t$ qu'en $t + 2D$, elle reste encore importante et peut donc être réduite par l'ajout d'un nouveau traitement spatial. Alors qu'en $t + 2D$ la sur-segmentation temporelle est déjà très réduite et ne peut donc plus être fortement réduite par l'ajout d'un traitement temporel supplémentaire.

Comme nous traitons des données vidéo, il aurait pu être intéressant de calculer les ZQP sur des données propres à la vidéo, comme par exemple le mouvement. En effet, nous aurions pu produire

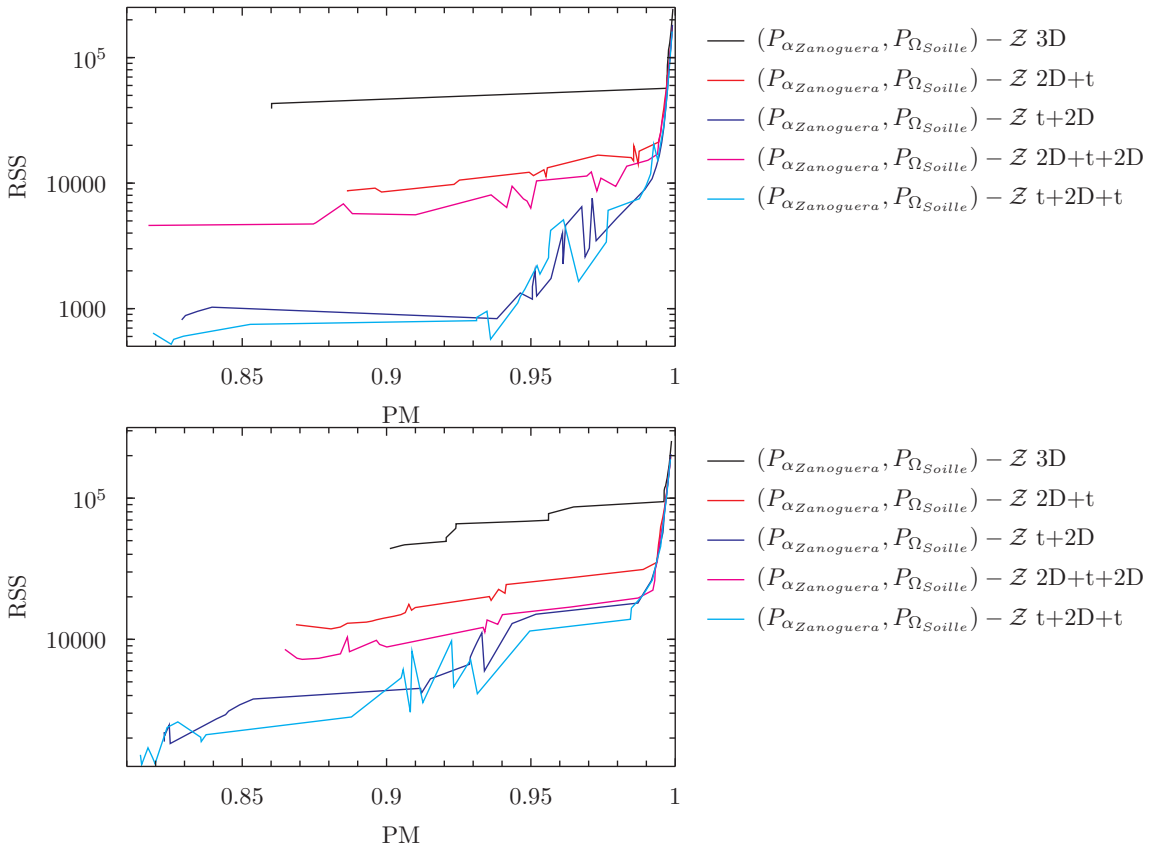


FIGURE 1.36 – Comparaison des approches 3D, 2D+t, t+2D, 2D+t+2D et t+2D+t pour les ZQP vidéo pour la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ sur les séquences vidéo *carphone* (en haut) et *foreman* (en bas).

les ZQP sur le flot optique au lieu de les produire sur les valeurs de pixels. La similarité ne se serait plus faite sur les valeurs des pixels mais sur la direction et la force du flot optique de chaque pixel, des pixels adjacents ayant une direction et une force de mouvement similaire appartenant vraisemblablement à la même ZQP car représentant sans doute le même objet. Cependant, un flot optique calculé directement sur les pixels d'une séquence vidéo est généralement très bruité et ne permet donc pas d'extraire une information de mouvement exploitable au niveau pixel. De plus, c'est une opération coûteuse en temps de calcul qui alourdirait fortement la production des ZQP en termes de temps de calcul.

Outre les approches $2D+t$ et $t+2D$, le cadre générique pour une construction incrémentale des ZQP que nous avons proposé permet une grande variété de combinaisons. Nous pourrions envisager par exemple la $(P^{couleur}_{\alpha_{Zanoguera}}, P^{couleur}_{\Omega_{Soille}})$-$\mathcal{Z}_{2D} \rightarrow (P^{flot\ optique}_{\alpha_{Zanoguera}}, P^{flot\ optique}_{\Omega_{Soille}})$-$\mathcal{Z}_t$ qui produirait d'abord les ZQP spatialement puis appliquerait le traitement temporel selon le mouvement. Le flot optique, produit sur les ZQP spatiales, serait alors moins bruité que s'il était appliqué directement sur les pixels. Des travaux portant sur l'apport des différentes combinaisons possibles pour la construction incrémentale des ZQP vidéo sont une perspective prometteuse pour l'extension des ZQP aux séquences vidéo.

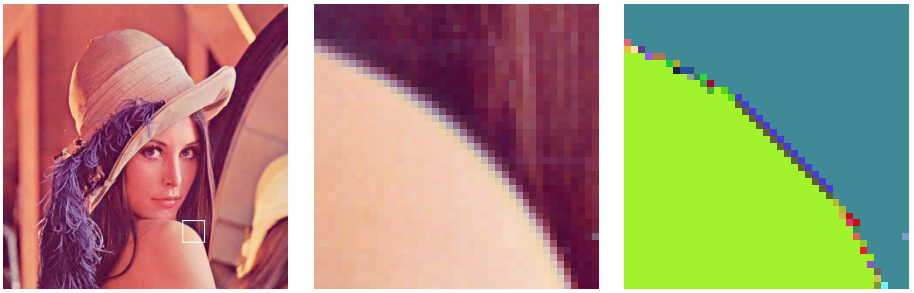
Au vu des performances médiocres de l'approche $3D$, nous ne considérerons dans la suite que les approches $2D+t$ et $t+2D$. Cependant, bien que donnant de meilleurs résultats, ces approches produisent toujours une sur-segmentation importante. Pour la réduire, nous traitons dans la section suivante de leur filtrage.

1.6 Filtrage

Dans cette section, nous abordons le problème des régions de transition commun à toutes les définitions de ZQP, ainsi que la problématique du filtrage des ZQP en vue de réduire la sur-segmentation. Nous étudions ensuite les solutions mises en œuvre pour le résoudre.

1.6.1 Problématique des régions de transition et du filtrage

Toutes les définitions des ZQP souffrent du problème des régions de transition. Les régions de transition sont les régions entre deux objets où se manifeste un phénomène d'escalier sur les valeurs des pixels frontaliers (cf. figure 1.37.b). Cet escalier est dû à la discrétisation de l'image et l'interpolation des valeurs qu'elle entraine. Ce phénomène provoque une sur-segmentation à proximité de cette frontière qui se retrouve composée de ZQP de très petite taille (cf. figure 1.37.c).



a *b* *c*

FIGURE 1.37 – Problème des régions de transition : a) Image originale, b) Phénomène d'escalier présent dans le carré blanc de l'image originale, c) $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ avec $\alpha = \omega = 100$.

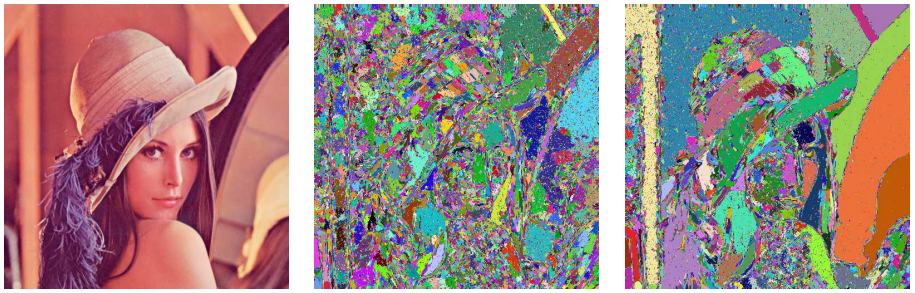
Cette sur-segmentation locale est importante et nécessite d'être traitée afin de réduire le nombre de ZQP. En outre, cette sur-segmentation ne contribue pas à la qualité des "pièces de puzzle" que sont les ZQP. En effet, ces petites zones se trouvant aux frontières des objets, elles n'augmentent pas la précision maximale mais ne font qu'augmenter la sur-segmentation.

Cependant, les régions de transition ne sont pas les seules responsables de la sur-segmentation. En effet, elles n'apparaissent que lors des phénomènes d'escalier et sont majoritairement présentes sur des frontières contrastées, comme dans l'exemple présenté dans la figure 1.37 où les régions de transition se trouvent entre l'épaule très claire de Lenna et ses cheveux sombres. On peut constater dans la figure 1.38, qu'il y a notamment de nombreuses ZQP ne contenant qu'un pixel et se trouvant enclavés dans d'autres ZQP. De telles ZQP ne sont pas des régions de transition et ne représentent pas des régions utiles dans l'assemblage des ZQP, il est donc nécessaire de les fusionner avec les ZQP dans lesquelles elles sont enclavées. Nous constatons également qu'il y a des régions qui sont segmentées en un trop grand nombre de ZQP (comme par exemple l'épaule de Lenna, que ce soit avec les paramètres $\alpha = \omega = 50$ ou $\alpha = \omega = 100$). Il est donc également nécessaire de filtrer ces régions pour obtenir une sur-segmentation plus faible et plus exploitable.

Afin de réduire la sur-segmentation, il est nécessaire de disposer de méthodes capables de filtrer les ZQP et d'être capable de supprimer les régions de transition, les ZQP mono-pixel enclavées dans d'autres ZQP et les ZQP créant une trop forte sur-segmentation dans une zone donnée.

1.6.2 Approches existantes

Plusieurs solutions ont été proposées pour résoudre le problème de la trop forte sur-segmentation des ZQP, nous allons à présent les étudier.



a *b* *c*

FIGURE 1.38 – Illustration de la sur-segmentation : a) Image originale, b) $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ avec $\alpha = \omega = 50$ (58219 $\mathcal{Z}$), c) $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ avec $\alpha = \omega = 100$ (34651 $\mathcal{Z}$).

Soille et Grazzini [SG09] définissent les régions de transition comme des ZQP ne contenant que des pixels de transition. Un pixel de transition est un pixel qui n'est pas un extremum local. Une région de transition est alors définie comme une région qui ne comporte aucun extremum local. Un pixel est un extremum local, en niveaux de gris, si et seulement si tous ses pixels voisins ont une valeur supérieure ou égal à la sienne ou si tous ses pixels ont une valeur inférieure ou égale à la sienne. Dans le contexte de la morphologie mathématique, un pixel p est un extremum local de l'image f si et seulement si le minimum entre le gradient par erosion (ρ^{ε}) et le gradient par dilatation (ρ^{δ}) pour p est égal à 0, l'élément structurant utilisé représentant le voisinage considéré :

$$p \text{ extremum local de } f \Leftrightarrow [\rho^{\varepsilon}(f) \wedge \rho^{\delta}(f)](p) = 0 \tag{1.35}$$

ici $\wedge$ représente l'infimum. Nous rappelons les définitions de ρ^{ε} etρ^{δ} :

$$\rho^{\varepsilon}(f)(p) = f(p) - \varepsilon(f)(p) \tag{1.36}$$

$$\rho^{\delta}(f)(p) = \delta(f)(p) - f(p) \tag{1.37}$$

avec ε et δ représentant respectivement les opérations d'érosion et de dilatation morphologique. On en déduit la définition d'un pixel de transition :

$$p \text{ pixel de transition de } f \Leftrightarrow [\rho^{\varepsilon}(f) \wedge \rho^{\delta}(f)](p) \neq 0 \tag{1.38}$$

La méthode consiste à supprimer toutes les ZQP qui sont des régions de transition, c'est-à-dire toutes celles qui ne comportent que des pixels de transition. L'espace que ces ZQP laissent libre va être comblé en utilisant un algorithme de croissance de régions, en l'occurence le Seeded Region Growing (SRG) [AB94]. Les ZQP non supprimées vont être utilisées comme graines et vont croître jusqu'à remplir tout l'espace laissé libre par les régions de transition. Les auteurs ont également proposé une extension de leur méthode à la couleur en adaptant la définition des pixels de transition. Cette extension consiste à considérer qu'un pixel est un pixel de transition seulement s'il est un pixel de transition dans chacune des bandes de l'image.

La figure 1.39 illustre la méthode de Soille et Grazzini en niveaux de gris et en couleur, elle présente également les masques des pixels de transition et des régions de transition. Si le nombre de ZQP est fortement réduit (de 60,7% en niveaux de gris et de 36,8% en couleur) la sur-segmentation reste encore très importante après le filtrage des régions de transition. Pourtant, les pixels de transition sont majoritaires dans la segmentation en ZQP. Mais les régions de transition représentent une part nettement moins importante car ce sont des régions composées exclusivement de pixels de transition et les extrema locaux, bien que minoritaires, sont très nombreux dans la segmentation initiale. L'importante différence entre les résultats obtenus en niveau de gris et en couleur est due à la définition d'un pixel de transition couleur. La nécessité pour un pixel de transition couleur d'être un pixel de transition indépendamment sur chaque bande est une contrainte plus forte qu'en niveaux de gris où il n'y a qu'une seule bande. Cette contrainte explique la différence en nombre de pixels de transition dont découle la différence en nombre de régions de transition.

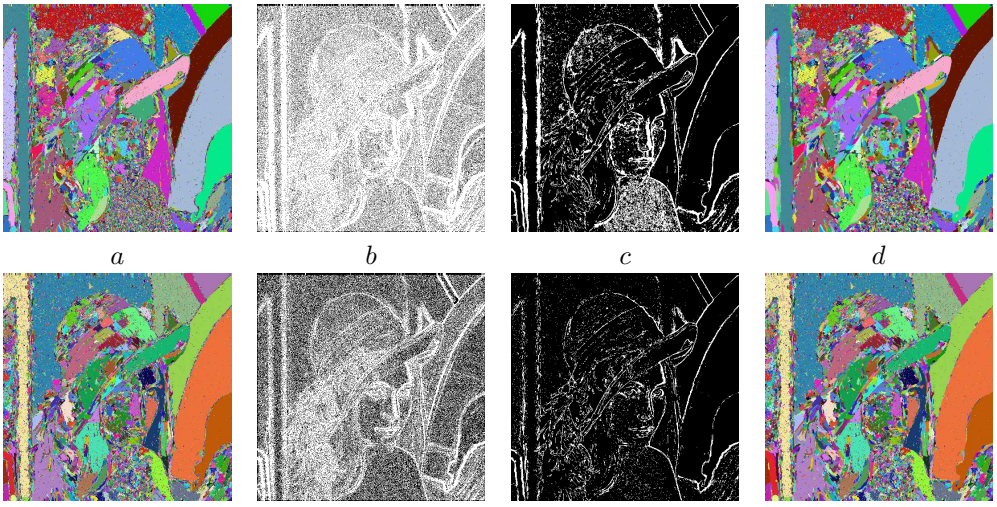


FIGURE 1.39 – Filtrage des régions de transition : (en haut) en niveaux de gris, a) $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ avec $\alpha = \omega = 100$ (31385 $\mathcal{Z}$), b) Masque des pixels de transition (82% des pixels de l'image), c) Masque des régions de transition (19073 régions de transition représentant 17% des pixels de l'image), d) ZQP après filtrage (12313 $\mathcal{Z}$), (en bas) en couleur, e) $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ avec $\alpha = \omega = 100$ (34651 $\mathcal{Z}$), f) Masque des pixels de transition (57% des pixels de l'image), g) Masque des régions de transition (12743 régions de transition représentant 9% des pixels de l'image), h) ZQP après filtrage (21909 $\mathcal{Z}$).

La solution proposée par Soille et Grazzini ne dépend d'aucun paramètre et repose sur une définition précise de ce qu'est une région de transition. Cependant, il reste de nombreuses régions après filtrage qui ne sont composées que de quelques pixels. Ces régions ne rentrent pas dans la définition des régions de transition mais provoquent néanmoins une forte sur-segmentation. Donc, même si cette méthode réduit la sur-segmentation en supprimant les régions de transition, elle est encore insuffisante. Cela corrobore ce que nous expliquions dans la section 1.6.1 : les régions de transition ne sont pas les seules responsables de l'importante sur-segmentation des ZQP.

Constatant les lacunes du filtrage des régions de transition, Soille [Soi10] propose de procéder à un pré-traitement des données initiales, c'est-à-dire l'image, plutôt que de faire un post-traitement de la segmentation, c'est-à-dire des ZQP. Ce pré-traitement consiste en un renforcement du contraste de l'image basé sur les extrema locaux. Il s'agit en premier lieu d'extraire les extrema locaux que nous avons définis précédemment (figure 1.40.a). Les extrema locaux sont ensuite étendus par une croissance de régions (ici un SRG) afin de couvrir l'ensemble de l'image pour former une mosaïque des extrema locaux (figure 1.40.b). Les pixels de chaque région de la mosaïque sont ensuite valués par la valeur de l'extremum local dont la région est issue. Le résultat est une image dont le contraste a été renforcé (figure 1.40.c). Ce pré-traitement a pour effet de renforcer les contours et donc de limiter l'effet d'escalier responsable d'une grande partie des régions de transition. On produit ensuite les ZQP sur l'image à contraste renforcé (figure 1.40.d).

Le filtrage par mosaïque d'extrema locaux, de par la simplification de l'image qu'il induit en créant des zones plates, permet d'obtenir une segmentation en ZQP moins sur-segmentée que si elle avait été effectuée sur l'image originale. On obtient même une sur-segmentation plus faible que par la méthode de filtrage des régions de transition. En effet, on constate une réduction des ZQP de 63% entre le filtrage des régions de transition et le filtrage par mosaïque d'extrema locaux sur l'image de Lenna. Cependant, cette méthode de filtrage n'est pas encore suffisante : de nombreuses ZQP composées d'un unique pixel subsistent, et la sur-segmentation bien que réduite par rapport à la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ sans filtrage, reste encore très importante. De plus, le renforcement du contraste de l'image altère les contours, phénomène particulièrement visible sur les bords du miroir

a *b* *c* *d*

FIGURE 1.40 – Filtrage par mosaïque d'extrema locaux : a) Masque des extrema locaux (43% des pixels de l'image), b) Mosaïque des extrema locaux (112461 régions), c) image originale à contraste renforcé, d) $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ avec $\alpha = \omega = 100$ sur l'image à contraste renforcé (8045 $\mathcal{Z}$).

de Lenna (figures 1.38 et 1.40.d). Cette altération a un effet direct sur la précision maximale et altère la qualité des ZQP produites en termes de précision des frontières. Le filtrage par mosaïque d'extrema locaux, bien que réduisant significativement la sur-segmentation, ne la réduit pas suffisamment et altère les contours dans l'image : il ne convient donc pas à notre objectif de fournir des ZQP précises en nombre restreint, permettant par assemblage d'obtenir les objets d'intérêt propres à l'utilisateur.

Le problème récurrent et non résolu des ZQP composées de quelques pixels et qui survivent aux filtrages précédents nécessite d'autres approches de filtrage. Ainsi, d'autres auteurs ont proposé des méthodes de filtrage des ZQP basées sur l'utilisation d'un seuil d'aire minimale.

Angulo [Ang03] propose de fusionner les ZQP dont l'aire est inférieure à un seuil. Il place les ZQP dans un graphe d'adjacence de régions (GAR) où les nœuds représentent les ZQP et sont valués par la couleur moyenne de la ZQP et leur aire. Nous notons que cette approche est proche en cela de notre approche incrémentale pour la production de ZQP. Les arêtes représentent les liens d'adjacence entre les ZQP et sont valuées de la différence entre les couleurs moyennes des ZQP qu'elles relient. Une fois cette structure de données construite, il va s'agir de la réduire jusqu'à ce que toutes les ZQP aient une aire supérieure ou égale au seuil. Le processus de fusion est le suivant : on sélectionne la ZQP ayant la plus petite aire (ou une des ZQP ayant la plus petite aire en cas d'égalité), on la fusionne avec la ZQP adjacente la plus similaire chromatiquement. Le GAR est mis à jour par fusion des deux nœuds et modification des valeurs des arêtes concernées. Ce processus est répété jusqu'à ce qu'il n'y ait plus aucune ZQP dont l'aire est inférieure au seuil. Cette méthode permet d'éliminer les ZQP de taille inférieure à un seuil ce qui élimine, même pour des valeurs de seuil basses, les ZQP de quelques pixels et les régions de transition. Cependant, les ZQP dont l'aire est inférieure au seuil étant fusionnées avec la ZQP adjacente la plus similaire chromatiquement, il est possible que cette ZQP soit également une ZQP d'aire inférieure au seuil et ainsi, de fusion en fusion on peut obtenir une ZQP dont l'aire est supérieure au seuil mais composée uniquement de régions de transition. Sachant que les frontières des objets sont généralement présentes dans les régions de transition cela peut avoir un impact négatif sur la précision maximale.

Zanoguera [Zan01] a proposé une méthode basée sur un principe proche. Elle supprime les ZQP dont l'aire est inférieure à un seuil, ce qui à l'instar de la méthode précédente inclut généralement les ZQP de quelques pixels et les régions de transition. Afin de combler l'espace laissé vide par la suppression des ZQP, les ZQP restantes sont utilisées comme marqueurs dans un algorithme de ligne de partage des eaux basée marqueurs [RBD92]. Les ZQP conservées sont ainsi étendues dans l'espace occupé précédemment par les ZQP supprimées. Cette méthode de filtrage est simple et repose sur une méthode de segmentation très utilisée en Morphologie Mathématique. Il n'y a pas de risque d'obtenir des ZQP composées de régions de transition comme dans la méthode précédente et les ZQP de quelques pixels sont filtrées. Cependant, la ligne de partage des eaux est appliquée sur les pixels, on perd alors les avantages de la réduction de données induite par les ZQP supprimées. Cela aura peu d'incidence si on utilise une valeur de seuil de quelques pixels mais, dans le cas de valeurs plus élevées qui auraient pour but, outre le filtrage des ZQP de quelques pixels et des

régions de transition, de réduire fortement la sur-segmentation, il aurait été intéressant d'utiliser les ZQP et non les pixels.

La méthode proposée dans Crespo *et al.* [CSS+97], que nous avons présentée dans la section 1.3.4, traite des ZP (et non des ZQP) et sélectionne les n ZP les plus significatives (au sens de différents critères) comme graines dans un processus de croissance de régions appliqué sur la partition en ZP. On obtient ainsi une réduction de sur-segmentation précise vu que l'on fixe le nombre de ZP finales. De plus, on n'applique pas la croissance de régions sur les pixels mais sur les ZP elles-mêmes. Ceci permet de conserver leurs frontières aux propriétés intéressantes et d'effectuer la croissance de régions sur un volume de données réduit. Il est tout à fait possible d'appliquer cette méthode aux ZQP.

Brunner et Soille [BS07, Soi10] proposent une méthode itérative de filtrage d'aire. Il s'agit à l'instar des méthodes précédentes de filtrer les ZQP dont l'aire est inférieure à un seuil. Cependant, plutôt que de filtrer directement toutes les ZQP inférieures au seuil d'aire, la suppression est faite itérativement en augmentant le seuil d'aire jusqu'au seuil voulu. D'abord on produit les ZQP. On initialise le seuil d'aire à 2. Ensuite, on supprime les ZQP dont l'aire est inférieure au seuil. Puis, on étend les ZQP restantes dans l'espace libéré par les ZQP supprimées en utilisant un algorithme de SRG ayant pour graines les ZQP restantes. On obtient ainsi une nouvelle partition de ZQP. Si le seuil d'aire courant n'est pas le seuil final, on incrémente le seuil courant et on relance le processus en utilisant la nouvelle partition de ZQP. Ainsi, on filtre petit à petit les ZQP ce qui évite lorsque le seuil d'aire est élevé d'avoir des ZQP conservées représentant peu de pixels et des ZQP supprimées représentant une grande partie de l'image ce qui nuirait à la qualité du résultat. Cette méthode permet de filtrer les ZQP de quelques pixels ainsi que les régions de transition, elle permet aussi de réduire significativement la sur-segmentation si le seuil d'aire est élevé. On observe en effet sur la figure 1.41 que la sur-segmentation est plus fortement réduite qu'avec les méthodes utilisant les extrema locaux et les ZQP de quelques pixels ont disparu. Les régions de transition n'ont pas toutes disparu, certaines se sont étendues en intégrant des pixels des ZQP supprimées devenant plus grandes et ayant alors une aire supérieure au seuil. C'est un inconvénient de la méthode itérative qui peut produire des régions de transition plus grandes qu'à l'origine comme la méthode proposée par Angulo. De plus, à l'instar de la méthode précédente, le filtrage d'aire itératif effectue sa croissance de régions sur les pixels, il aurait là aussi été intéressant de l'effectuer sur les ZQP surtout pour des valeurs de seuil élevées.

Les approches existantes de filtrage de ZQP permettent de réduire l'importante sur-segmentation générée par une partition en ZQP. Cependant, pour une précision élevée, la sur-segmentation reste encore importante, surtout pour les méthodes sans paramètre qui s'attachent surtout à filtrer les régions de transition. Ce filtrage n'est pas suffisant, des méthodes basées sur un seuil d'aire minimale ont donc été développées pour réduire de façon plus importante la sur-segmentation. Si elles réduisent effectivement (selon la valeur du seuil) la sur-segmentation de façon plus importante, notamment en supprimant efficacement les ZQP de quelques pixels, elles souffrent également de problèmes divers. Certaines peuvent produire des régions de transition plus grandes en agglomérant les petites régions de transition. D'autres filtrent les ZQP trop petites mais perdent leurs informations en étendant les ZQP conservées sur les pixels et non sur ces ZQP supprimées. De plus, l'utilisation d'un seuil d'aire pose le problème du réglage de ce seuil. Le seuil idéal dépend de l'image à traiter. En effet, sur une image présentant des objets d'intérêt de grande dimension on peut utiliser un seuil élevé car l'information n'est pas présente au niveau pixel tandis que sur une image satellite qui présente de petits objets d'intérêt, utiliser un seuil élevé donne des ZQP inadéquates car l'information portée par un pixel est importante. Ce problème est illustré dans la figure 1.42. On constate, comme souligné précédement, que le filtrage d'aire permet de réduire la sur-segmentation de façon plus importante que les autres méthodes. Mais, on remarque également que, dès que la valeur du seuil d'aire augmente, on réduit également la précision maximale de la partition en ZQP après filtrage. Le réglage du seuil est donc un réel problème qu'il faut résoudre, régler une valeur d'aire n'étant pas très intuitif. Nous proposons dans la section suivante une approche qui résout ces différents problèmes.

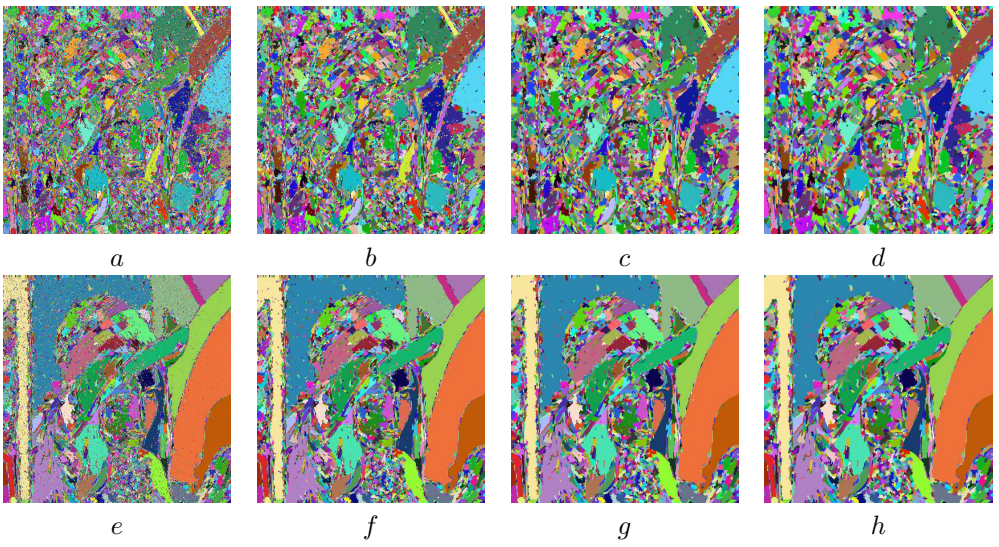


FIGURE 1.41 – Filtrage d'aire itératif avec la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$: (haut) $\alpha = \omega = 50$ a) ZQP initiales (58219 $\mathcal{Z}$) b) seuil=5 (11423 $\mathcal{Z}$),c) seuil=10 (6335 $\mathcal{Z}$),d) seuil=15 (4528 $\mathcal{Z}$), (bas) $\alpha = \omega = 100$ e) ZQP initiales (34651 $\mathcal{Z}$) f) seuil=5 (6381 $\mathcal{Z}$),g) seuil=10 (3386 $\mathcal{Z}$),h) seuil=15 (2411 $\mathcal{Z}$).

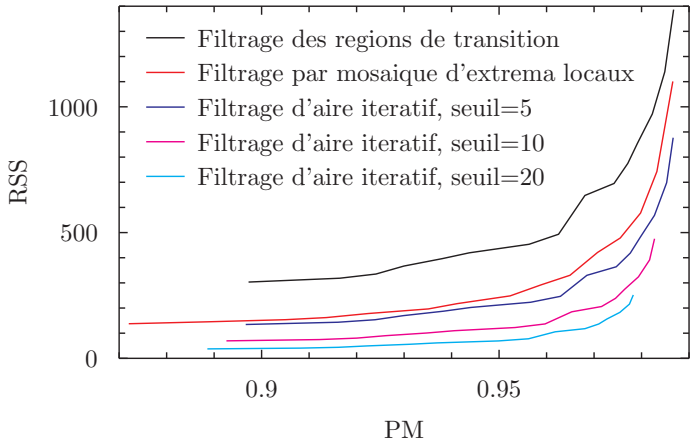


FIGURE 1.42 – Comparaison des différentes approches de filtrage existantes avec la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ sur l'ensemble des images de la base de Berkeley.

1.6.3 Le filtrage d'aire par fusion

Dans cette section, nous présentons une approche originale pour le filtrage, c'est-à-dire le filtrage d'aire par fusion. Dans un premier temps nous la définissons puis la comparons avec les méthodes existantes. Puis, nous proposons une méthode pour le réglage du seuil d'aire.

1.6.3.1 Définition

Nous proposons une méthode de filtrage de ZQP proche des méthodes de filtrage d'aire proposées par Zanoguera et Soille et de l'utilisation des ZP par Crespo. Pour l'instant, nous conservons la nécessité de régler un seuil d'aire, nous résoudrons ce problème dans la section 1.6.3.2. Le processus de filtrage consiste en deux étapes :

(1) Suppression des ZQP dont l'aire est inférieure au seuil ;

(2) Extension des ZQP conservées par un SRG sur les ZQP supprimées.

Cependant, au lieu d'appliquer le SRG sur les pixels, nous étendons les ZQP conservées sur les ZQP supprimées. Ainsi, le SRG ne travaille pas directement sur les pixels mais sur la réduction de données que représentent les ZQP.

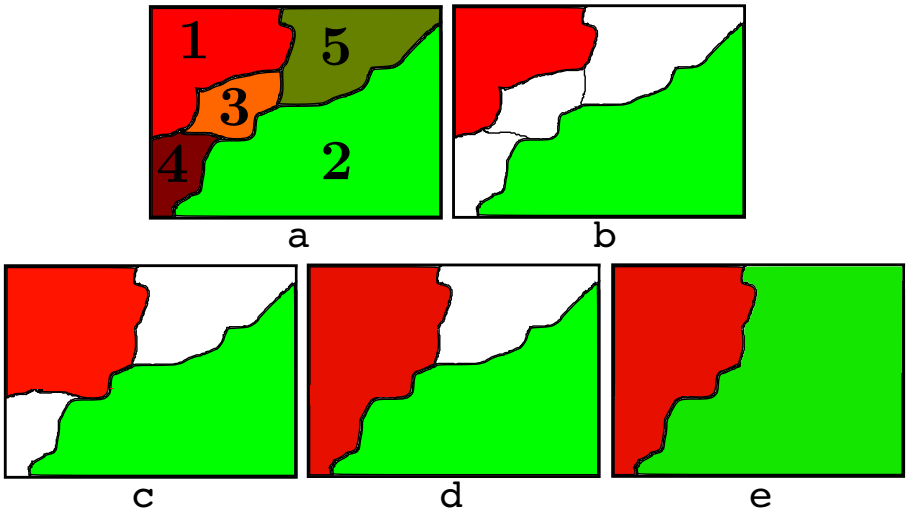


FIGURE 1.43 – Le filtrage d'aire par fusion : a) ZQP originales, b) Suppression des ZQP dont l'aire est inférieure au seuil, c) 1ère itération du SRG : ZQP 1 ⊃ ZQP 3, d) 2ème itération du SRG : ZQP 1 ⊃ ZQP 4, e) Dernière itération du SRG : ZQP 2 ⊃ ZQP 5.

La figure 1.43 illustre le processus de filtrage. Nous représentons les ZQP par la couleur moyenne des pixels qui les composent (figure 1.43.a). Nous supprimons les ZQP dont l'aire est inférieure au seuil d'aire minimale mais nous conservons leurs définitions spatiales, c'est-à-dire les pixels qu'elles recouvrent, et leur couleur moyenne (cf figure 1.43.b, suppression des régions 3, 4 et 5). On applique ensuite un SRG en utilisant les ZQP conservées comme graines. Mais au lieu d'appliquer le SRG sur les pixels nous l'appliquons sur les ZQP supprimées. Ceci donne en premier lieu sur la figure l'extension de la ZQP 1 qui incorpore la ZQP 3 (cf. figure 1.43.c) car c'est entre la ZQP 1 et la ZQP 3 que la différence de couleur moyenne est la plus faible. Cette extension de la ZQP 1 modifie sa couleur moyenne. Puis, c'est entre la ZQP 4 et la ZQP 1 étendue que la différence de couleur moyenne est la plus faible, la ZQP 1 est alors étendue en englobant la ZQP 4, ce qui modifie encore sa couleur moyenne (cf. figure 1.43.d). Enfin, la ZQP 2 et la ZQP 5 présentent la différence de couleur la plus faible et donc la ZQP 2 est étendue pour recouvrir la ZQP 5, ce qui comme pour la ZQP 1 modifie sa couleur moyenne (cf. figure 1.43.e). Les ZQP occupent à nouveau tout l'espace de définition mais la sur-segmentation a été réduite puisqu'il y avait 5 ZQP à l'origine et plus que 2 à présent.

La figure 1.44 présente les résultats obtenus avec notre méthode de filtrage sur des ZQP produites par $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$. La sur-segmentation est fortement réduite même avec une valeur de seuil d'aire faible. Si l'on compare les résultats de notre approche avec ceux du filtrage d'aire itératif présentés dans la figure 1.41, on observe qu'à seuil égal, notre méthode réduit plus fortement la sur-segmentation. Cela est dû au fait qu'augmenter itérativement la valeur du seuil peut conduire à l'agglomération de ZQP plus petites que le seuil d'aire final. Cette agglomération peut ainsi leur permettre d'atteindre ou de dépasser l'aire critique représentée par le seuil d'aire final. De telles ZQP sont supprimées par notre méthode. Dans le cas de grandes valeurs du seuil d'aire notre méthode peut alors entrainer une sous-segmentation de certaines parties de l'image mais le filtrage itératif tend à conserver trop de régions et éventuellement à étendre des régions perturbatrices telles que les régions de transition.

Cependant, si l'on compare les résultats obtenus par le filtrage d'aire itératif et notre approche sur la base de Berkeley (cf figure 1.45), on observe qu'à précision égale, notre approche génère une

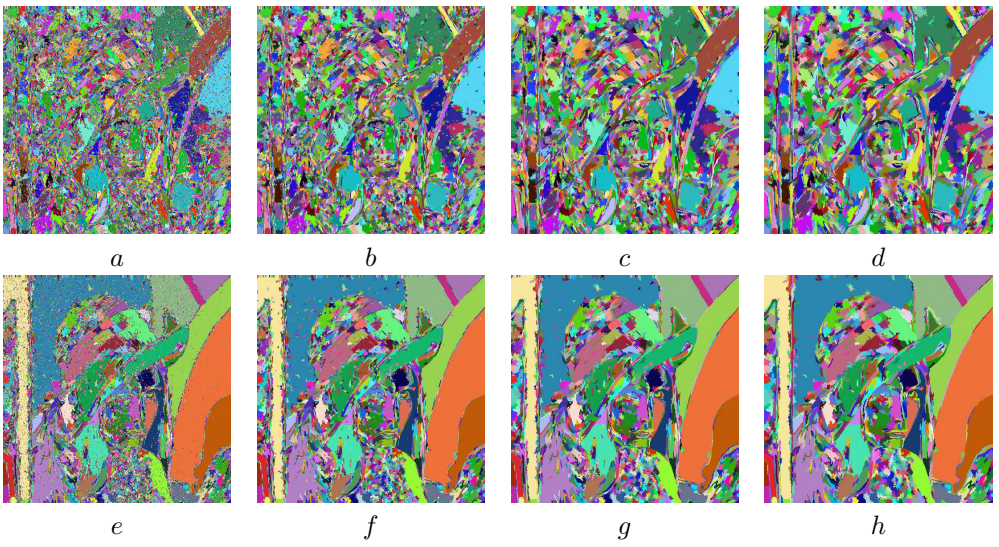


FIGURE 1.44 – Filtrage par aire et reconstruction par SRG sur les ZQP sur $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$: (haut) $\alpha = \omega = 50$ a) ZQP initiales (58219 $\mathcal{Z}$) b) seuil=5 (8014 $\mathcal{Z}$),c) seuil=10 (3572 $\mathcal{Z}$),d) seuil=15 (2219 $\mathcal{Z}$), (bas) $\alpha = \omega = 100$ e) ZQP initiales (34651 $\mathcal{Z}$) f) seuil=5 (4600 $\mathcal{Z}$),g) seuil=10 (2051 $\mathcal{Z}$),h) seuil=15 (1252 $\mathcal{Z}$).

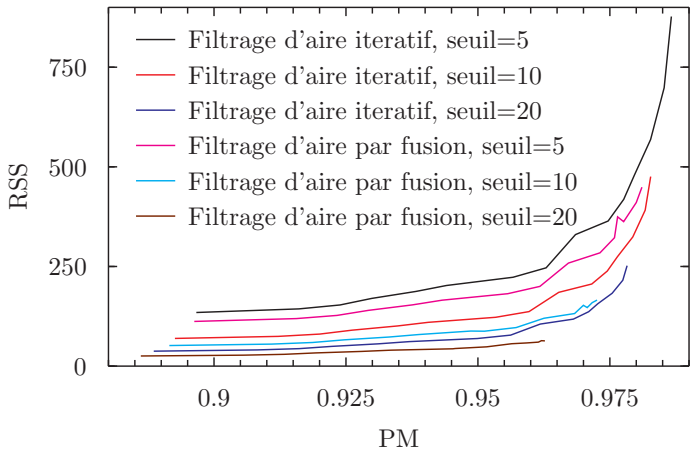


FIGURE 1.45 – Comparaison de notre approche et du filtrage d'aire itératif sur la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ sur l'ensemble des images de la base de Berkeley.

sur-segmentation inférieure à celle produite par le filtrage d'aire itératif. On ne peut cependant pas conclure que notre approche est dans l'absolu plus efficace que le filtrage d'aire itératif. En effet, s'il est indéniable que notre approche produit moins de ZQP à précision égale, en revanche les valeurs maximales de précision maximale qu'elle produit sont plus faibles, effet direct de la suppression des ZQP inférieures au seuil. En effet, il y a un risque (d'autant plus grand que le seuil est élevé) de supprimer des ZQP représentant des objets d'intérêt ou des parties homogènes d'objet d'intérêt et non une zone sur-segmentée. Ce risque est atténué dans le filtrage d'aire itératif par l'augmentation itérative de la valeur du seuil mais cette augmentation itérative, en sauvegardant des ZQP, augmente aussi de façon importante le nombre final de ZQP et donc la sur-segmentation. Cette différence de valeur maximale de la précision maximale pourrait aussi s'expliquer par la différence de mode de propagation du SRG, notre approche utilise les ZQP supprimées alors que le filtrage itératif étend les ZQP conservées sur les pixels. L'avantage de notre approche est que l'on garde l'information spatiale des ZQP, et notamment leurs frontières qui, comme nous l'avons vu précédemment, produisent généralement des mesures de précision maximale intéressantes (pour

des valeurs α et ω pas trop élevées). Cette approche nous permet de travailler sur un espace de données réduit et donc d'avoir un coût calculatoire plus restreint qu'une propagation sur les pixels. Cependant, si le filtrage itératif d'aire a un coût calculatoire plus élevé, sa propagation du SRG sur les pixels peut permettre de corriger certaines erreurs de frontières présentes originellement dans les ZQP ou d'en conserver qui auraient été supprimées par la propagation via ZQP. Mais il est aussi possible qu'en perdant l'information des ZQP supprimées et en propageant les ZQP conservées sur les pixels, le filtrage d'aire itératif produise des erreurs.

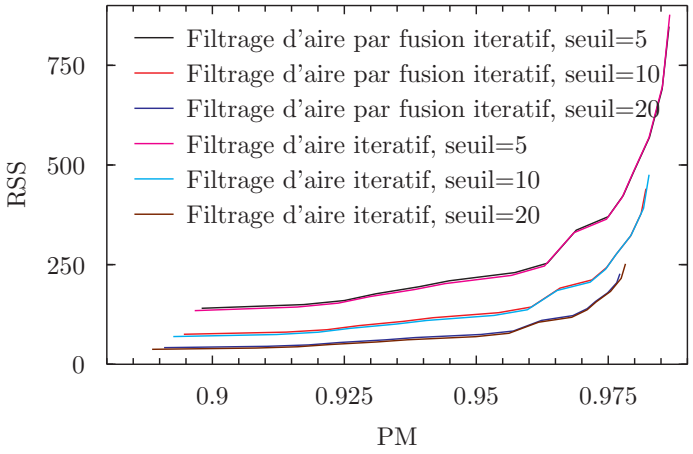


FIGURE 1.46 – Comparaison de notre approche en version itérative et du filtrage d'aire itératif sur la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ sur l'ensemble des images de la base de Berkeley.

Afin de s'affranchir de l'aspect itératif et de comparer les deux méthodes de propagation utilisées par les deux approches, nous avons développé une version itérative de notre méthode. Nous avons ensuite comparé le filtrage d'aire itératif et notre méthode en version itérative sur la base de Berkeley. Les résultats sont présentés dans la figure 1.46. On constate que les deux méthodes présentent des résultats très proches même si notre méthode en version itérative produit, à précision égale, une sur-segmentation légèrement plus élevée au filtrage d'aire itératif. Concernant la valeur maximale de la précision, il existe toujours un écart mais beaucoup plus faible qu'avec la version classique de notre approche. Ainsi, le fait de faire croître les ZQP conservées sur les ZQP supprimées au lieu des pixels des ZQP supprimées n'a pas une grande influence sur la qualité du résultat produit. La différence constatée précédemment vient plutôt de l'utilisation ou non d'une augmentation itérative du seuil.

En conclusion, il est préférable d'utiliser notre approche pour le SRG, car si propager les ZQP conservées sur les ZQP supprimées au lieu de le faire sur leurs pixels ne change pas la qualité du résultat, cette approche nécessite un temps de calcul inférieur car elle travaille sur une réduction des données et non sur les données elles-mêmes. L'augmentation itérative du seuil permet d'obtenir un meilleur maximum de précision maximale mais produit une sur-segmentation plus importante. Dans la suite, nous utilisons notre approche de filtrage en version non-itérative car elle produit à précision maximale égale une sur-segmentation inférieure.

1.6.3.2 Régler le seuil, une tâche complexe

Notre méthode de filtrage ne nécessite qu'un seul paramètre, l'aire minimale des ZQP. Si ce paramètre est simple et représente une mesure familière pour l'utilisateur, fixer cette aire minimale n'a rien d'intuitif. Pourtant, ce paramètre doit être fixé empiriquement, car selon les images, l'aire minimale idéale d'une ZQP varie. De plus, choisir une aire minimale trop élevée peut supprimer des ZQP significatives et mener à une sous-segmentation de certaines parties de l'image. A l'inverse, choisir une aire minimale trop faible ne réduira pas suffisamment la sur-segmentation. En résumé, ce seuil va dépendre du type de l'image, de son contenu, de sa taille et de ce que désire en faire

l'utilisateur. En effet, si l'utilisateur désire obtenir de petits détails sur une image, il désirera une forte sur-segmentation afin d'être sûr qu'aucun des détails qu'il souhaite extraire ne soit sous-segmenté. A l'inverse s'il désire extraire de grands objets d'intérêt, il désirera une sur-segmentation faible pour qu'il ait à assembler le minimum de ZQP pour y parvenir. Dans ce contexte, fixer une aire minimale est très complexe et absolument pas intuitif.

Afin de résoudre ce problème de réglage d'aire minimale, nous proposons deux paramètres plus intuitifs qui permettront de déterminer le seuil d'aire minimale :

(1) Le nombre de ZQP à obtenir après filtrage (proche de l'approche de [CSS+97] pour les ZP) ;

(2) Le pourcentage de ZQP à conserver après filtrage.

On peut noter que ces deux choix sont liés, en effet si on fixe le pourcentage de ZQP à conserver après filtrage, on peut obtenir le nombre de ZQP à obtenir après filtrage et inversement :

$$\text{Nombre de ZQP à conserver} = \text{Pourcentage de ZQP à conserver} \times \text{Nombre de ZQP} \tag{1.39}$$

$$\text{Pourcentage de ZQP à conserver} = \frac{\text{Nombre de ZQP à conserver}}{\text{Nombre de ZQP}} \tag{1.40}$$

Pour obtenir l'aire minimale, il suffit d'ordonner les ZQP par leur aire, de partir de l'aire la plus élevée et de descendre jusqu'à ce que le nombre (respectivement le pourcentage) de ZQP dont l'aire est égale ou supérieure à l'aire actuelle soit supérieure ou égale au nombre (respectivement au pourcentage) de ZQP à conserver. Ainsi on obtient l'aire minimale des ZQP à conserver. Par contre, il est possible que le nombre (respectivement le pourcentage) de ZQP après filtrage ne soit pas exactement le nombre (respectivement le pourcentage) de ZQP fixé car plusieurs ZQP peuvent avoir la même aire et donc en diminuant le seuil d'aire de 1, il est possible que le nombre de ZQP conservées augmente de plus que 1.

Le nombre de ZQP après filtrage ou le pourcentage de ZQP à conserver après filtrage sont des paramètres plus intuitifs que l'aire minimale et seront donc plus faciles à régler par un utilisateur. Cependant, s'ils sont plus intuitifs, leur réglage n'en est par pour autant trivial. En effet, il est difficile de déterminer un nombre de ZQP qui garantisse une sur-segmentation minimale tout en limitant voire empêchant la sous-segmentation. Mais, si régler le seuil à une valeur optimale est complexe, le régler à une valeur garantissant la limitation ou l'absence de sous-segmentation tout en réduisant de manière importante la sur-segmentation est possible. Dans cette optique, le seuillage percentile du nombre de régions est plus adapté car il fixe de façon relative la réduction de données. En effet, un seuil de 1% consiste à diviser par 100 le nombre de régions. On ne décide donc pas du nombre de ZQP à conserver mais de l'ampleur de la réduction de données que l'on désire. Ce choix a plus de sens que de fixer un nombre de ZQP à garder. En effet, prenons deux images de taille identique et sur-segmentons les par la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ en utilisant des paramètres identiques. Si une image produit 100 ZQP et l'autre 10 000 ZQP, cela induit que leur contenu est très différent, la première contient de vastes régions homogènes et l'autre de petites régions très différentes les unes des autres. Utiliser dans cette configuration un seuil absolu identique d'un certain nombre de régions n'a pas de sens. Soit ce seuil est supérieur au nombre de ZQP obtenues dans la première image et il n'y aura aucun filtrage pour cette image. Soit ce seuil est inférieur au nombre de ZQP obtenues dans la première image, et le filtrage de la seconde sera très important provoquant à coup sûr une sous-segmentation. L'utilisation d'un seuil percentile permet de résoudre ce problème en effectuant un filtrage relatif au nombre de ZQP et donc au contenu de l'image.

Dans la suite du document, nous utilisons le seuil percentile et le seuil d'aire. En effet, même s'il est difficile à régler, le seuil d'aire représente une tolérance à l'erreur acceptée par l'utilisateur et permet, lorsqu'il est bien réglé, d'éliminer efficacement les régions de transition.

1.6.4 Filtrage vidéo

Les séquences vidéo représentant un volume de données plus important que les images, elles génèrent une sur-segmentation plus importante. Il est donc nécessaire de disposer d'une méthode de

filtrage adaptée aux séquences vidéo. On pourrait étendre trivialement le filtrage d'aire par fusion et ne plus considérer une aire minimale mais un volume minimal des ZQP. Cependant, si considérer un volume minimal serait efficace dans le contexte d'images réellement tri-dimensionnelles, ce n'est pas adapté aux séquences vidéo qui sont spatio-temporelles et non purement spatiales. En effet, en considérant un volume minimal, une ZQP n'ayant spatialement que quelques pixels d'aire mais étant présente sur de nombreuses trames serait gardée alors qu'elle ne représente sans doute pas un objet mais une partie d'un objet très sur-segmenté. De même, un objet ayant une aire importante mais présent uniquement dans quelques trames ne serait pas conservé si le volume minimal était trop élevé. De plus, si fixer une aire minimale n'est pas intuitif, fixer un volume minimal l'est encore moins. Pour contourner ce problème, nous utilisons un seuil d'aire moyenne minimum, l'aire moyenne se calculant ainsi :

$$\mathcal{A}_m = \frac{\text{nombre de pixels de la ZQP}}{\text{nombre de trames où la ZQP est présente}} \tag{1.41}$$

L'utilisation de l'aire moyenne d'une ZQP permet de conserver des ZQP spatialement étendues même si elle ne sont pas présentes sur un grand nombre de trames et de supprimer des ZQP présentes sur de nombreuses trames mais spatialement peu étendues. Le filtrage par aire moyenne réduit donc fortement la sur-segmentation spatiale. Mais, il n'est pas efficace contre la sur-segmentation temporelle car le critère de filtrage est purement spatial. Nous pourrions ajouter un critère temporel pour le filtrage, par exemple un nombre minimum de trames dans lesquelles une ZQP doit être présente. Cependant, le réglage d'un tel seuil pose d'importants problèmes. En effet, dans le cas de l'utilisation de paramètres ou d'approches de ZQP produisant une importante sur-segmentation temporelle, il y a un risque élevé que cette sur-segmentation soit homogène. Autrement dit, que les ZQP spatialement étendues représentant une sur-segmentation temporelle auront des durées en terme de nombre de trames comparables. Dès lors, fixer un seuil devient délicat car l'on risque pour un seuil faible de n'enlever que peu de ZQP par rapport au seul filtrage spatial et pour un seuil plus élevé de supprimer un nombre trop important de ZQP ce qui peut induire une sous-segmentation. Nous utilisons donc un filtrage basé sur l'aire moyenne qui est efficace pour réduire la sur-segmentation spatiale mais moins pertinent pour la réduction de la sur-segmentation temporelle.

La figure 1.47 présente les résultats du filtrage par aire moyenne pour les différentes approches vidéo sur les séquences *carphone* et *foreman*. Si l'on compare les résultats du filtrage avec ceux des approches vidéo sans filtrage (présentés dans la figure 1.36), on constate que le filtrage par aire moyenne réduit très fortement la sur-segmentation à précision égale, à l'instar de ce que nous observions pour les images fixes. Nous avons utilisé l'aire moyenne car il n'était pas intuitif et peu pertinent a priori de fixer un volume minimal. Cependant l'utilisation de seuils relatifs au nombre de régions ou au pourcentage de régions pour déterminer le volume minimal rend plus intuitif ce type de seuil. De plus, cela nous permet de comparer le filtrage par aire moyenne minimale et par volume minimal en fixant un seuil de nombre de régions ou de pourcentage de régions qui déterminera l'aire moyenne minimale et le volume minimal. Nous avons comparé les résultats obtenus en utilisant un seuil de pourcentage de région de 10% sur la séquence *carphone* pour les différentes approches vidéo. Le seuil percentile de régions a été utilisé pour déterminer le seuil d'aire moyenne minimale et de volume minimal. Le résultat de l'expérience est présenté dans la table 1.2.

α/ω	3D				2D+t				t+2D			
	$\mathcal{A}_m$		Vol		$\mathcal{A}_m$		Vol		$\mathcal{A}_m$		Vol	
	PM	RSS	PM	RSS	PM	RSS	PM	RSS	PM	RSS	PM	RSS
50	0,994	4910	0,994	3956	0,992	1435	0,992	1346	0,991	817	0,992	709
100	0,993	3192	0,994	3204	0,934	703	0,934	720	0,977	258	0,977	218
150	0,824	1839	0,825	2120	0,842	701	0,842	562	0,874	106	0,874	111

TABLE 1.2 – Comparaison du filtrage par aire moyenne minimale et par volume minimal de $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ sur la séquence *carphone* pour un seuil percentile de régions de 10%.

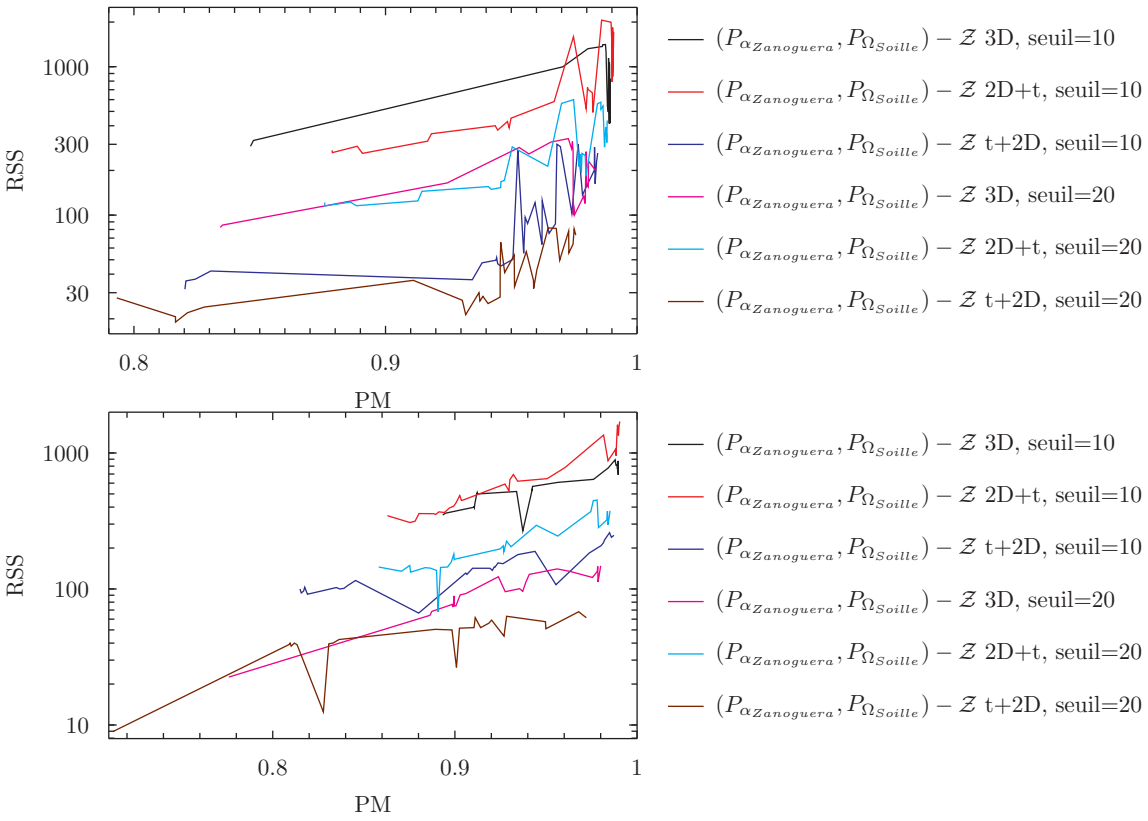


FIGURE 1.47 – Filtrage par aire moyenne sur les différentes extensions vidéo de la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ sur les séquences vidéo *carphone* (en haut) et *foreman* (en bas).

On observe que le filtrage par volume minimal produit une précision maximale comparable au filtrage par aire moyenne minimale. Concernant la sur-segmentation, le filtrage par aire moyenne minimale tend plus souvent à produire plus de ZQP que le filtrage par volume minimal. Ce phénomène est dû à l'utilisation du seuillage percentile de régions et de la possibilité de ne pas avoir exactement le bon pourcentage de régions (cf. section 1.6.3.2). La méthode utilisant l'aire moyenne produit plus de régions car il y a plus de ZQP à aire moyenne égale que de ZQP à volume égal, la plage de valeurs de l'aire moyenne étant plus faible que celle du volume.

Intuitivement, le filtrage par volume minimal ne semblait pas être pertinent. Cependant devant la difficulté du filtrage par aire moyenne à diminuer la sur-segmentation temporelle et la possibilité offerte par le seuillage percentile de régions de déterminer le volume minimal, le filtrage par volume minimal obtient des résultats équivalents au filtrage par aire moyenne en terme de précision maximale et produit une sur-segmentation plus réduite dans la plupart des cas. Ces bonnes performances s'explique par la réduction des sur-segmentations spatiales et temporelles là où l'aire moyenne réduit surtout la sur-segmentation spatiale. Or si le filtrage par aire moyenne minimale réduit bien la sur-segmentation spatiale et moins bien la sur-segmentation temporelle, le filtrage par volume minimal est moins performant pour la réduction de sur-segmentation spatiale mais plus performant pour la réduction de sur-segmentation temporelle que le filtrage par aire moyenne. Nous notons toutefois que dans le cas d'objets d'intérêt présents dans un nombre restreint de trames, le filtrage par volume minimal risquerait de supprimer les ZQP composant de tels objets et donc de rendre impossible l'obtention de ces objets par assemblage de ZQP.

1.6.5 Discussion

La sur-segmentation des ZQP dépend de plusieurs facteurs tels que les régions de transition, les pixels isolés de valeurs très différentes de celles des pixels de leur voisinage et des objets d'in-

térêt composés de régions hétérogènes. Les méthodes se concentrant sur les régions de transition réduisent la sur-segmentation mais ne règlent pas les problèmes induits par les autres facteurs. Les méthodes de filtrage basées sur l'aire des ZQP réduisent efficacement la sur-segmentation mais induisent différents problèmes selon les méthodes tels que l'accroissement des régions de transition et la perte de l'information des ZQP supprimées en propageant les régions conservées sur les pixels. De plus, elles posent le problème du réglage du seuil d'aire.

Nous avons proposé une méthode qui résout ces différentes problèmes ainsi qu'une approche de paramétrisation plus intuitive que le réglage d'un seuil d'aire par l'utilisation d'un seuil de nombre de ZQP à conserver ou d'un pourcentage de ZQP à conserver. De plus, notre approche faisant croître les régions conservées sur les ZQP supprimées et non leurs pixels, elle est moins coûteuse calculatoirement car effectuée sur une réduction des données.

Nous avons ensuite étendu cette approche aux séquences vidéo en proposant deux méthodes de filtrage, le filtrage par aire moyenne minimale d'une ZQP spatio-temporelle et le filtrage par volume minimal d'une ZQP spatio-temporelle. Ces deux méthodes de filtrage produisent des résultats de qualité équivalente et réduisent fortement la sur-segmentation des ZQP vidéo.

En combinant les définitions de ZQP adaptées aux séquences vidéo et les méthodes de filtrage, nous obtenons une méthode efficace de pré-segmentation. Cette pré-segmentation pourra être utilisée par des méthodes de fusion de ZQP en vue d'obtenir la segmentation désirée par un utilisateur.

1.7 Implantations

Nous n'avons pas encore abordé le coût calculatoire de la production de ZQP. Il s'agit pourtant d'un aspect capital surtout dans le cadre d'application des ZQP sur des données vidéo, qui sont par nature très volumineuses. Nous proposons donc dans cette section des algorithmes efficaces pour la production de ZQP. Nous discutons ensuite de l'utilisation de structure de données qui permettent un traitement rapide des données. Puis nous étudions l'utilisation de tables de correspondance pour accélérer les calculs de distance sur les ZQP couleur. Enfin, nous expérimentons une méthode de décrémentation supérieure pour la $(P_\alpha, P_{\Omega_{Soille}})$-$\mathcal{Z}$ avant de dresser un bilan des différents apports de cette section sur l'amélioration du coût calculatoire.

1.7.1 Algorithmes de construction de ZQP

Pour chaque définition de ZQP, il existe plusieurs algorithmes possibles de production de ces ZQP. Nous allons dans cette section distinguer les constructions de ZQP selon la définition de l'α-$\mathcal{Z}$ ou de la $(P_\alpha, P_1, ..., P_n)$-$\mathcal{Z}$. Nous présentons, pour chaque type de ZQP, l'algorithme naïf et un algorithme optimisé dont nous comparons le coût calculatoire.

L'approche naïve pour l'α-$\mathcal{Z}$ consiste, pour chaque pixel non affecté à une α-$\mathcal{Z}$, à construire l'α-$\mathcal{Z}$ à laquelle il appartient en explorant son voisinage. On ajoute à l'α-$\mathcal{Z}$ courante les pixels du voisinage qui vérifient la condition de variation locale (α), et on explore ensuite leur voisinage pour continuer à étendre l'α-$\mathcal{Z}$.

L'algorithme 1 est celui de l'approche naïve. Il est relativement trivial et possède une complexité assez faible. En effet, pour chaque pixel, on ne parcourt qu'une fois son voisinage. lors de l'ajout du pixel à une α-$\mathcal{Z}$. Si nous avons n pixels dans l'image et un voisinage de N_n pixels, nous obtenons une complexité en $O(n * (N_n + 1))$ réduite après élimination des facteurs constants à $O(n)$. La complexité de l'algorithme naïf est donc linéaire.

Bien que d'une complexité relativement faible, l'approche naive n'est cependant pas la plus efficace. En effet, nous pouvons nous inspirer des algorithmes d'étiquetage en composantes connexes pour construire les α-$\mathcal{Z}$. L'étiquetage en composantes connexes [SS01], défini pour les images binaires, consiste à donner la même étiquette aux pixels voisins n'étant pas du fond. On peut aisément l'étendre aux images en niveaux de gris et aux images multibandes, les composantes connexes sont

Entrées : f : image, α : seuil de variation locale
Données : $\mathcal{Z}_c$: ZQP courante, S_{FIFO} : file FIFO de pixels à traiter, N : voisinage utilisé, p : pixel courant
Résultat : $\mathcal{Z}$: ensemble des ZQP couvrant f

```
1  Z ⟵ ∅
2  pour chaque p ∈ f faire
3      si p ∉ Z alors
4          Z_c ⟵ ∅
5          S_FIFO.empile(p)
6          tant que S_FIFO ≠ ∅ faire
7              q ⟵ S_FIFO.depile()
8              Z_c ⟵ Z_c ∪ q
9              pour chaque q' ∈ N(q) faire
10                 si d(f(q), f(q')) ⩽ α et q' ∉ Z_c et q' ∉ S_FIFO alors
11                     S_FIFO.empile(q')
12                 finsi
13             finpourchaque
14         fintantque
15         Z ⟵ Z ∪ Z_c
16     finsi
17 finpourchaque
```

Algorithme 1 : Algorithme naïf de production d'α-$\mathcal{Z}$.

alors des zones plates. L'étiquetage en composantes connexes donne alors la même étiquette aux pixels voisins ayant la même valeur dans l'image. Nous pouvons adapter cet algorithme à la production d'α-$\mathcal{Z}$. En effet, l'α-$\mathcal{Z}$ consiste à donner la même étiquette aux pixels voisins ayant une différence de valeur dans l'image inférieure à α.

L'approche efficace pour la production de α-$\mathcal{Z}$ est basée sur l'algorithme d'étiquetage en composantes connexes proposé par Rosenfeld et Pflatz [RP66]. Cet algorithme consiste à parcourur deux fois les pixels de l'image. Une première fois en étudiant pour chaque pixel son semi-voisinage afin de lui donner une étiquette, une nouvelle ou une de ses semi-voisins, et éventuellement de noter une équivalence entre les étiquettes de ses semi-voisins. Un semi-voisinage est un voisinage classique amputé de la moitié de ses voisins. Les voisins conservés sont les voisins antérieurs au pixel. Nous nous plaçons ici dans le contexte d'un parcours croissant dans l'ordre suivant des dimensions abcisse puis ordonnée puis temps. Dans ce contexte, les voisins antérieurs sont les voisins du pixel précédement parcouru, c'est à dire ceux dont au moins une des coordonnées est inférieure à celle du pixel courant. La figure 1.48 présente les 4-semi-voisinage et 8-semi-voisinage.

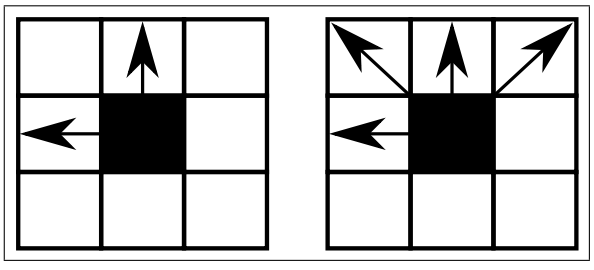


FIGURE 1.48 – Le 4-semi-voisinage et le 8-semi-voisinage.

Le second parcours des pixels consiste à réétiqueter les pixels en fonction des équivalences notées lors du premier parcours et simplifiées (c'est-à-dire que si $a \Leftrightarrow b$ et $b \Leftrightarrow c$ on simplifie par $a \Leftrightarrow b$ et $a \Leftrightarrow c$). Il s'agit donc de parcourir une fois les n pixels de l'image et les $n_{\frac{1}{2}N}$ pixels de leurs semi-voisinages puis une nouvelle fois les n pixels ce qui donne une complexité en $O(n*(n_{\frac{1}{2}N}+2))$ réduite après élimination des facteurs constants à $O(n)$. La complexité de l'approche efficace est également linéaire. Cependant, si l'on étudie les deux complexités avant réduction des facteurs constants, on constate que la complexité de l'approche efficace est plus faible que celle de l'approche naïve. Le gain théorique de l'approche efficace par rapport à l'approche naïve est de l'ordre de 50 %.

Entrées : f : image, α : seuil de variation locale
Données : p : pixel courant
Résultat : Image d'étiquettes représentant les α-$\mathcal{Z}$
Premier parcours de l'image
pour chaque $p \in f$ **faire**
  si *un ou plusieurs semi-voisins de p vérifient la condition de variation locale* **alors**
    On affecte à p la plus petite étiquette des semi-voisins la vérifiant
    si *plusieurs semi-voisins d'étiquette différente vérifiaient la condition de variation locale* α **alors**
      On ajoute l'équivalence entre les étiquettes à la table d'équivalence
    finsi
  sinon
    On affecte à p une nouvelle étiquette
  finsi
finpourchaque
Simplification de la table d'équivalence
Second parcours de l'image
Réétiqueter les pixels selon la table d'équivalence des étiquettes
Les pixels ayant la même étiquette appartiennent à la même ZQP

Algorithme 2 : Algorithme efficace de production d'α-$\mathcal{Z}$.

La $(P_\alpha, P_1, ..., P_n)$-$\mathcal{Z}$ que nous avons définie comme l'extension de la connexité des prédicats logiques aux espaces multivariés et plus particulièrement pour nos applications aux espaces couleurs peut aussi produire des ZQP selon différents algorithmes. En premier lieu, il y a l'algorithme naïf (cf. Algorithme 3) qui consiste, pour chaque pixel n'appartenant pas encore à une zone plate, à produire l'α-(Z) de ce pixel avec la valeur α. Une fois l'α-$\mathcal{Z}$ produite, on vérifie si elle satisfait tous les prédicats. Si ce n'est pas le cas, on détruit l'α-$\mathcal{Z}$ produite puis on en construit une nouvelle avec $\alpha \leftarrow \alpha - 1$ et on vérifie à nouveau si tous les prédicats sont satisfaits. Dans le cas contraire, on décrémente à nouveau α et on recommence le procesus. Par contre, si tous les prédictas sont vérifiés, on prend un autre pixel n'appartenant à aucune ZQP et on le traite de la même manière, jusqu'à ce que tous les pixels appartiennent à une ZQP. Cette méthode, bien qu'adaptée à tous les types de prédicats, n'est pas très efficace car elle nécessite de construire entièrement chaque α-$\mathcal{Z}$ avant de vérifier si elle satisfait les prédicats. Dans le cas de prédicats difficiles à satisfaire et de valeur α élevée, on risque de calculer de nombreuses α-$\mathcal{Z}$ inutilement. Cet algorithme a une complexité plus élevée que ceux de l'α-$\mathcal{Z}$. Nous prenons comme opération élémentaire l'ajout d'un pixel à une α-$\mathcal{Z}$ en construction. L'algorithme consiste à construire l'α-$\mathcal{Z}$ pour chaque pixel, de vérifier si elle satisfait tous les prédicats et si ce n'est pas le cas de recommencer cette construction en décrémentant α. Dans le pire des cas, la création d'une α-$\mathcal{Z}$ satisfaisant tous les prédicats débouche sur une zone plate, ce qui aura nécessité α décrémentations. Et donc, $\alpha + 1$ constructions de ZQP, qui dans le pire des cas auront à chaque fois conduit à la construction d'α-$\mathcal{Z}$ incluant tous les pixels non inclus dans une ZQP. Toujours dans le pire des cas, la zone plate finale ne fait qu'un pixel, si ce schéma est appliqué sur toute l'image (ou séquence vidéo) à traiter, nous obtenons dans le pire des cas $n(\alpha)+1$ ajouts de pixel à une ZQP pour la première ZQP, $(n-1)(\alpha)+1$ ajouts pour la deuxième, $(n-2)(\alpha)+1$ ajouts pour la troisième, etc. Nous obtenons dans ce cas extrême, une complexité en $O(\frac{1}{2}(n^2+n)(\alpha+1)+n)$, soit après simplification une complexité en $O(n^2)$ (car en général $n \ll \alpha$). L'approche naïve pour la $(P_\alpha, P_1, ..., P_n)$-$\mathcal{Z}$ présente une complexité quadratique.

On peut également s'inspirer de l'algorithme proposé par Soille [Soi08] pour l'(α, ω)-$\mathcal{ZS}$ (cf Algorithme 4) et vérifier si les prédicats sont satisfaits à chaque fois qu'on ajoute un pixel à l'α-$\mathcal{Z}$ en cours de construction. On évite ainsi de construire entièrement les α-$\mathcal{Z}$ à chaque itération, ce qui permet un gain non négligeable de temps de calcul. Cependant, tous les prédicats ne peuvent être vérifiés à chaque ajout de pixel : c'est le cas du prédicat basé sur l'indice de connexité. L'ajout d'un pixel peut rendre l'indice de connexité inférieur au seuil β, mais il est possible que l'ajout d'autres pixels puisse faire remonter l'indice au-dessus du seuil. La vérification à chaque pixel n'est donc pas adaptée au prédicat de l'indice de connexité qu'il faut vérifier une fois l'α-$\mathcal{Z}$ entièrement produite.

Nous avons développé un algorithme mêlant les deux idées précédentes (cf. Algorithme 5). Nous classons les prédicats en deux classes, les prédicats "locaux" (P^{local}) et "globaux" (P^{global}). Les

Entrées : f : image, α : seuil de variation locale, P : ensemble des prédicats logiques, P_α : prédicat d'α-connexité
Données : $\mathcal{Z}_c$: ZQP courante, α_c : α courant, S_{FIFO} : file FIFO de pixels à traiter, N : voisinage utilisé p : pixel courant, ZQPValide : booléen indiquant si la ZQP est valide
Résultat : $\mathcal{Z}$: ensemble des $(P_\alpha, P_1, ..., P_n)$-$\mathcal{Z}$ couvrant f

```
1  Z ⟵ ∅
2  pour chaque p ∈ f faire
3      si p ∉ Z alors
4          α_c ⟵ α
5          ZQPValide ⟵ faux
6          tant que ZQPValide=faux faire
7              Z_c ⟵ ∅
8              S_FIFO.empile(p)
9              tant que S_FIFO ≠ ∅ faire
10                 q ⟵ S_FIFO.depile()
11                 Z_c ⟵ Z_c ∪ q
12                 pour chaque q' ∈ N(q) faire
13                     si d_Pα(f(q), f(q')) ⩽ α_c et q' ∉ Z_c et q' ∉ S_FIFO alors
14                         S_FIFO.empile(q')
15                     finsi
16                 finpourchaque
17             fintantque
18             si ∀P_i ∈ P, P_i(Z_c) = vrai alors
19                 ZQPValide ⟵ vrai
20             sinon
21                 α_c ⟵ α_c − 1
22             finsi
23         fintantque
24         Z ⟵ Z ∪ Z_c
25     finsi
26 finpourchaque
```

Algorithme 3 : Algorithme naïf pour la $(P_\alpha, P_1, ..., P_n)$-$\mathcal{Z}$.

prédicats "locaux" sont ceux dont on peut vérifier la satisfaction à chaque ajout de pixel lors de la construction des α-$\mathcal{Z}$ (par exemple ω). A l'inverse, les prédicats "globaux" sont ceux dont on ne peut vérifier la satisfaction qu'une fois l'α-$\mathcal{Z}$ totalement construite. Parmi les prédicats vus précédemment, la variation globale est un prédicat local tandis que l'indice de connexité est un prédicat global. Pour gagner en efficacité, dans le cas de ZQP complexes nécessitant un nombre important de prédicats, il est utile de trier les prédicats afin de vérifier les plus contraignants, c'est-à-dire les plus difficiles à satisfaire, en premier. Afin d'obtenir dans le pire des cas la construction des zones plates de l'image, nous posons que les prédicats sont toujours vérifiés pour les zones plates, c'est-à-dire pour les α-$\mathcal{Z}$ avec $\alpha = 0$. La complexité théorique de l'algorithme efficace est la même que celle de l'algorithme naïf. En effet, dans le pire des cas, le ou les prédicats ne seraient violés que lorsque l'α-$\mathcal{Z}$ courante contiendrait tous les pixels non présents dans une ZQP, rendant inefficace la distinction entre prédicats locaux et globaux. Cependant, si en théorie la complexité de l'approche efficace est en $O(n^2)$ comme celle de l'approche naïve, dans la pratique l'approche efficace est moins coûteuse. Il est tout à fait possible qu'un prédicat "local" ne soit violé qu'à l'ajout du dernier pixel de l'α-$\mathcal{Z}$ courante pour un α spécifique. Toutefois, dans la plupart des cas un prédicat "local" violé ne sera pas violé lors du dernier ajout, ce qui permet à l'algorithme d'économiser, à chaque construction de ZQP relancée, un certain nombre d'ajouts de pixels. Et donc d'être moins coûteux que la version naïve. Nous disposons ainsi d'un algorithme simple et efficace, adapté à la $(P_\alpha, P_1, ..., P_n)$-$\mathcal{Z}$.

1.7.2 Utilisation de structures efficaces

La production de ZQP et leur filtrage nécessitent de gérer une grande quantité de données surtout pour les séquences vidéo où le nombre de pixels est très important. Il est donc nécessaire de disposer de structures efficaces pour traiter ce grand volume de données et ce afin de garantir un temps de calcul raisonnable.

```
Entrées : f : image, α : seuil de variation locale, ω : seuil de variation globale
Données : Z_c : ZQP courante, α_c : α courant, S_FIFO : file FIFO, fp : file de priorité, N : voisinage utilisé, p : pixel courant
rl : image des variations locales initialisées à ∞, rlval : variation locale courante, prio_c : priorité courante,
mincc : valeur minimum de la ZQP courante, maxcc : valeur maximum de la ZQP courante
Résultat : Z : ensemble des (α, ω)-ZS de f
pour chaque p ∈ f faire
    si p ∉ Z alors
        Z_c ⟵ {p}
        mincc ⟵ maxcc ⟵ f(p)
        α_c ⟵ α
        pour chaque q ∈ N(p) faire
            rlval ⟵ |f(p) − f(q)|
            si q ∈ Z alors
                si α_c ≥ rlval alors
                    α_c ⟵ rlval − 1
                finsi
                passer à l'itération suivante
            finsi
            si rlval ≤ α_c alors
                rl(q) ⟵ rlval
                fp.insère(rlval, q)
            finsi
        finpourchaque
        prio_c ⟵ fp.prioritéMax()
        tant que fp ≠ ∅ faire
            q ⟵ fp.dépile()
            si (q ∈ Z) ou (q ∈ Z_c) alors
                passer à l'itération suivante
            finsi
            si q.prio > prio_c alors
                tant que S_FIFO ≠ ∅ faire
                    Z_c ⟵ Z_c ∪ S_FIFO.dépile()
                fintantque
                prio_c ⟵ q.prio
                si (q ∈ Z) ou (q ∈ Z_c) alors
                    passer à l'itération suivante
                finsi
            finsi
            S_FIFO.empile(q)
            si f(q)<mincc alors
                mincc ⟵ f(q)
            finsi
            si f(q)>maxcc alors
                maxcc ⟵ f(q)
            finsi
            si (ω < maxcc − mincc) ou (prio_c > α_c) alors
                tant que S_FIFO ≠ ∅ faire
                    rl(S_FIFO.dépile()) ⟵ ∞
                fintantque
                tant que fp ≠ ∅ faire
                    rl(fp.dépile()) ⟵ ∞
                fintantque
                sortir de la boucle
            finsi
            pour chaque q′ ∈ N(q) faire
                rlval ⟵ |f(q) − f(q′)|
                si (q′ ∈ Z) et (α_c ≥ rlval) alors
                    α_c ⟵ rlval − 1
                    si prio_c > α_c alors
                        tant que S_FIFO ≠ ∅ faire
                            rl(S_FIFO.dépile()) ⟵ ∞
                        fintantque
                        tant que fp ≠ ∅ faire
                            rl(fp.dépile()) ⟵ ∞
                        fintantque
                        sortir de la boucle
                    finsi
                    passer à l'itération suivante
                finsi
                si (rlval > α_c) ou (rlval ≥ rl(q′)) alors
                    passer à l'itération suivante
                sinon
                    si rlval < rl(q′) alors
                        rl(q′) ⟵ rlval
                        fp.insère(rlval, q′)
                    finsi
                finsi
            finpourchaque
        fintantque
        tant que S_FIFO ≠ ∅ faire
            Z_c ⟵ Z_c ∪ S_FIFO.dépile()
        fintantque
        Z ⟵ Z ∪ Z_c
    finsi
finpourchaque
```

Algorithme 4 : Algorithme de Soille pour l'(α, ω)-$\mathcal{ZS}$.

Entrées : f : image, α : seuil de variation locale , P_α : prédicat d'α-connexité, P^{local} : ensemble des prédicats locaux, P^{global} : ensemble des prédicats globaux
Données : $\mathcal{Z}_c$: ZQP courante, α_c : α courant, S_{FIFO} : file FIFO de pixels à traiter, N : voisinage utilisé, p : pixel courant
Résultat : $\mathcal{Z}$: ensemble des $(P_\alpha, P_1, ..., P_n)$-$\mathcal{Z}$ couvrant f

```
1  Z ⟵ ∅
2  pour chaque p ∈ f faire
3      si p ∉ Z alors
4          α_c ⟵ α
5          Z_c ⟵ {p}
6          ajouterVoisins(p)
7          ZQPValide ⟵ faux
8          répéter
9              tant que S_FIFO ≠ ∅ faire
10                 q ⟵ S_FIFO.depile()
11                 si ∀P_i ∈ P^local, P_i(Z_c ∪ {q}) = vrai alors
12                     Z_c ⟵ Z_c ∪ {q}
13                     ajouterVoisins(q)
14                 sinon
15                     α_c ⟵ α_c − 1
16                     Z_c ⟵ {p}
17                     S_FIFO ⟵ ∅
18                     ajouterVoisins(p)
19                 finsi
20             fintantque
21             si ∀P_i ∈ P^global, P_i(Z_c) = vrai alors
22                 ZQPValide ⟵ vrai
23             sinon
24                 α_c ⟵ α_c − 1
25                 Z_c ⟵ {p}
26                 ajouterVoisins(p)
27             finsi
28         jusqu'à ZQPValide = vrai ;
29         Z ⟵ Z ∪ Z_c
30     finsi
31 finpourchaque

32 ajouterVoisins(pixel q) :
33 pour chaque q' ∈ N(q) faire
34     si d_{P_α}(f(q), f(q')) ⩽ α_c et q' ∉ Z_c et q' ∉ S_FIFO alors
35         si q' ∉ Z alors
36             S_FIFO.empile(q')
37         sinon
38             α_c ⟵ α_c − 1
39             Z_c ⟵ {q}
40             S_FIFO ⟵ ∅
41             ajouterVoisins(q')
42             sortir de la boucle
43         finsi
44     finsi
45 finpourchaque
```

Algorithme 5 : Algorithme efficace pour la connexité des prédicats logiques.

Les graphes offrent une structure de données efficace particulièrement adaptée pour représenter des pixels ou des régions et leurs relations d'adjacence. Nous les utilisons pour deux types de tâches dans nos implantations : la construction des ZQP dans le cadre de notre approche incrémentale et, l'application du SRG dans le processus de filtrage de ZQP.

Pour la construction des ZQP, nous transformons les pixels ou les ZQP produites précédemment en graphe d'adjacence de régions. Chaque nœud représente un pixel ou une ZQP et est décrit selon les prédicats utilisés (conformément à la figure 1.31). Pour l'adjacence, nous relions les nœuds selon l'adjacence considéré. Chaque arête est valuée de la différence entre les deux nœuds qu'elle relie selon le prédicat d'α-connexité utilisé. Si la valeur de l'arête est supérieure à la valeur du paramètre α, elle n'est pas ajoutée au graphe, ce qui allége la structure de données et accélère la future production de ZQP. Nous construisons ensuite les ZQP sur ce graphe selon la $(P_\alpha, P_1, ..., P_n)$-$\mathcal{Z}$.

Pour le filtrage de ZQP, nous transformons la partition de l'image ou de la vidéo en graphe pour y appliquer le SRG. Chaque ZQP est un nœud du graphe valué de la couleur moyenne de la ZQP. Nous ajoutons ensuite les liens d'adjacence selon le voisinage considéré. Nous ne créons pas d'arête reliant les ZQP graines adjacentes. Seules les arêtes reliant des ZQP graines à des ZQP supprimées sont valuées. Elles sont valuées de la différence entre les valeurs des deux nœuds qu'elles relient. L'ordre de fusion des ZQP est guidé par la valeur des arêtes : ce sont les ZQP reliées par l'arête ayant la plus petite valeur qui sont fusionnées. Lorsqu'une ZQP graine croit en englobant une ZQP supprimée, la valeur du nœud la représentant est mise à jour avec la nouvelle couleur moyenne, le nœud représentant la ZQP englobée est supprimé, l'arête les reliant est supprimée, les arêtes de la ZQP supprimée sont rattachées à la ZQP graine et leur valeur est mise à jour. Si une arête relie deux ZQP graines, elle est supprimée. Lorsqu'il n'y a plus d'arête, le filtrage est terminé : il n'y a plus que des ZQP graines.

Lorsque que les nœuds du graphe représentent des ZQP, cela permet d'appliquer efficacement différents traitements. En effet, les liens d'adjacence sont inclus dans le graphe via les arêtes, il n'est plus nécessaire de parcourir le voisinage d'un pixel ou d'une ZQP pour obtenir ses voisins, ce qui réduit le coût calculatoire.

1.7.3 Utilisation de tables de correspondance

Les tables de correspondance (LUT) sont fréquement utilisées lorsqu'un calcul est récurrent et coûteux afin d'accélérer son traitement. Il s'agit de stocker dans une table le résultat de ce calcul. Dans le cas de l'extension proposée par Zanoguera pour étendre les ZQP à la couleur, l'utilisation de tables de correspondance est adaptée. En effet, le calcul de distance peut s'avérer coûteux alors que l'utilisation d'une table transforme le coût du calcul en coût de lecture d'une case d'un tableau.

Algorithme	Sans LUT	Avec LUT
α-$\mathcal{Z}$ naïf	1	0.98
α-$\mathcal{Z}$ optimisé	1	0.89
$(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ efficace	1	0.95

TABLE 1.3 – Comparaison normalisée des temps de calcul pour la production de ZQP couleur en fonction de l'utilisation ou non des tables de correspondance sur la séquence vidéo *foreman*.

La tableau 1.3 compare les temps de calculs normalisés de l'α-$\mathcal{Z}$ couleur selon Zanoguera et de la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ avec la distance L_2. Le gain pour l'algorithme naïf y est très léger tandis que le gain pour l'algorithme optimisé est plus conséquent ($> 10\%$). Concernant la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$, le gain est plus modeste que pour l'algorithme optimisé de l'α-$\mathcal{Z}$ mais reste tout de même intéressant car de l'ordre de 5%. La différence de gain selon les méthodes s'explique par la part du calcul de la distance dans leur coût calculatoire.

L'utilisation de tables de correspondance permet de réduire le temps de calcul dans le cadre de l'utilisation du prédicat d'α-connexité de Zanoguera. Cependant, une telle table pour une image ou une vidéo couleur standard prend une place importante en mémoire (16 Mo). Toutefois, cette taille s'avère négligeable lorsque comparée à la place mémoire que peut prendre une séquence vidéo.

1.7.4 Approximation par décrémentation supérieure

L'implantation algorithmique de la définition de la $(P_\alpha, P_{\Omega_{Soille}})$-$\mathcal{Z}$ repose sur la décrémentation de la valeur de α lorsque tous les prédicats ne sont pas vérifiés. Ceci permet de garantir l'obtention de l'α-$\mathcal{Z}$ maximale vérifiant tous les prédicats. Dans le cas de grande valeur de α, il est possible qu'il y ait un grand nombre de décrémentations notamment si les prédicats sont très contraignants pour l'image ou la vidéo en cours de traitement. Ces décrémentations sont responsables de la majeure partie du coût calculatoire de la production de $(P_\alpha, P_{\Omega_{Soille}})$-$\mathcal{Z}$. Une solution pour diminuer ce coût serait d'utiliser une décrémentation supérieure à 1, ainsi on diminuerait le nombre de décrémentations nécessaires pour satisfaire tous les prédicats, ce qui aurait pour effet d'accélérer le traitement.

α/ω	Décrémentation de α				
	Classique	$\alpha = \alpha - 2$	$\alpha = \alpha - 3$	$\alpha = \alpha - 4$	$\alpha = \alpha - 5$
50	1	0,58	0,45	0,38	0,34
100	1	0,51	0,37	0,28	0,23
150	1	0,51	0,34	0,26	0,22

TABLE 1.4 – Comparaison normalisée des temps de calcul pour la production de $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ selon différentes décrémentations d'α sur l'ensemble des images de la base de Berkeley.

La table 1.4 présente une comparaison normalisée des temps de calcul obtenus pour différentes valeurs de décrémentation de α. On constate qu'une décrémentation supérieure à 1 entraine une diminution significative du coût calculatoire de la production de $(P_\alpha, P_{\Omega_{Soille}})$-$\mathcal{Z}$. Même en fixant le pas de décrémentation à 2 (ce qui double le pas de décrémentation), nous obtenons une diminution de 42% du temps de calcul pour $\alpha = \omega = 50$ et 49% pour $\alpha = \omega = 100$ et $\alpha = \omega = 150$. Si l'utilisation d'un pas de décrémentation supérieur à 1 réduit efficacement le coût calculatoire de la production de $(P_\alpha, P_{\Omega_{Soille}})$-$\mathcal{Z}$, le gain calculatoire se paie par une baisse de la qualité des ZQP obtenues. En effet, si l'on a $\alpha = 10$, que pour la valeur 10 un prédicat n'est pas vérifié, que nous utilisons un pas de 2, nous testerons donc ensuite les prédicats sur une α-$\mathcal{Z}$ produite avec une valeur de 8. Si pour cette valeur de α, tous les prédicats sont vérifiés nous garderons cette α-$\mathcal{Z}$. Cependant, si dans cet exemple, pour une valeur de α de 9 aucun prédicat n'était violé nous aurions pu produire une α-$\mathcal{Z}$ avec la valeur 9. Ce qui aurait, de plus, été cohérent avec la définition de la $(P_\alpha, , P_1, ..., P_n)$-$\mathcal{Z}$, puisque dans le cadre de la connexité des prédicats logiques, nous cherchons l'α-$\mathcal{Z}$ la plus grande qui satisfait tous les prédicats.

La figure 1.49 présente les résultats obtenus pour la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ selon différents pas de décrémentation. Nous constatons que l'utilisation d'un pas de décrémentation supérieur à 1 a une influence sur la qualité des résultats. À précision maximale égale, les pas de décrémentation supérieurs à 1 produisent une plus grande sur-segmentation que le pas de décrémentation de 1. L'utilisation d'un pas de décrémentation supérieur à 1 diminue le coût calculatoire mais diminue également la qualité des ZQP produites. Cependant, si l'utilisation d'un pas de décrémentation de 2 diminue la qualité des ZQP, on observe que les résultats obtenus sont relativement proches de ceux obtenus pour un pas de décrémentation de 1 alors que le gain en temps de calcul est important ($> 40\%$).

L'utilisation de pas de décrémentation supérieur à 1 ne permettant pas d'obtenir des ZQP respectant exactement la définition de la $(P_\alpha, P_1, ..., P_n)$-$\mathcal{Z}$, nous en déconseillons l'utilisation. Cependant, dans le cas d'applications comportant des contraintes fortes sur le temps de calcul,

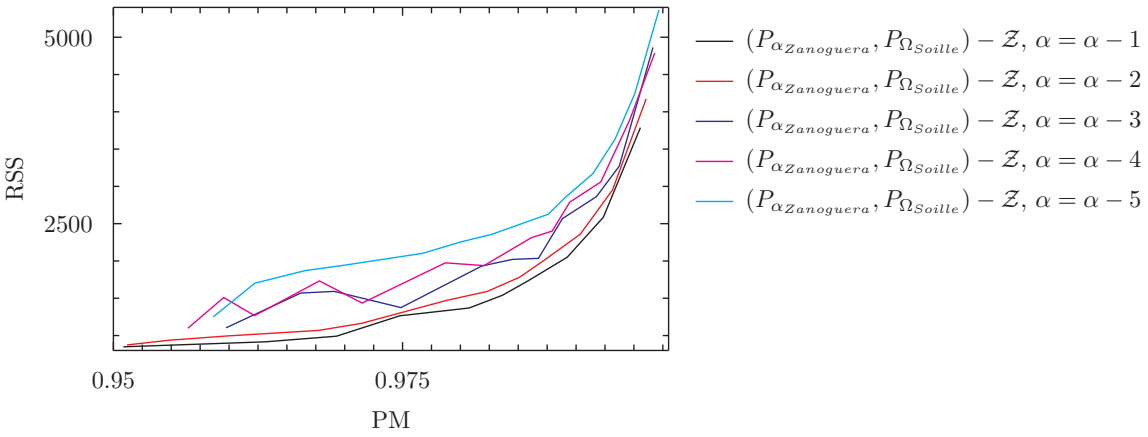


FIGURE 1.49 – Comparaison de la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$ selon différents pas de décrémentation sur l'ensemble des images de la base de Berkeley.

nous préconisons l'utilisation d'un pas de décrémentation de 2. En effet, la diminution de qualité que ce choix entraîne est légère tandis que le gain en terme de temps de calcul est très important.

1.7.5 Discussion

Nous avons abordé dans cette section l'aspect algorithmique de la production de ZQP et proposé un algorithme efficace pour la $(P_\alpha, P_1, ..., P_n)$-$\mathcal{Z}$ capable de prendre en compte tous les prédicats logiques. Nous avons également présenté la façon dont nous utilisions les structures de données de type graphe sur certains de nos traitements dans le but de les rendre les plus rapides possibles. Nous avons ensuite montré l'intérêt de l'utilisation de tables de correspondance dans le cadre de l'utilisation du prédicat d'α-connexité de Zanoguera. Enfin, nous avons évalué la réduction du temps de calcul apportée par l'utilisation d'un pas de décrémentation de α supérieur à 1 dans le cadre de la $(P_\alpha, P_1, ..., P_n)$-$\mathcal{Z}$ ainsi que son incidence sur la qualité des ZQP produites.

La combinaison de ces différentes contributions à l'amélioration du coût calculatoire de la production et du filtrage des ZQP permet d'obtenir un outil de présegmentation non seulement intéressant du point de vue de la qualité de la présegmentation mais également du point de vue de son efficacité computationnelle.

1.8 Conclusion

Dans ce chapitre, nous avons étudié les zones quasi-plates définies par Soille [Soi08] comme étant des pièces de puzzle qu'il s'agit d'assembler en vue d'obtenir des objets d'intérêt.

Nous avons présenté les différentes définitions existantes des ZQP, l'unification théorique des différentes méthodes via la connexité des prédicats logiques ainsi que quelques applications existantes des ZQP. Nous avons ensuite abordé le problème de l'évaluation des ZQP. Les ZQP étant une méthode de segmentation produisant une importante sur-segmentation, nous avons retenu deux critères : le ratio de sur-segmentation représentant le ratio de ZQP obtenues par rapport au nombre de régions de la vérité-terrain, et la précision maximale qui représente la précision en termes de pixel que l'on obtiendrait si l'on assemblait les ZQP du mieux possible par rapport à la vérité-terrain. Nous avons également étudié les extensions existantes des ZQP à la couleur. Nous avons déterminé que l'espace couleur et la distance avaient peu d'influence sur le prédicat d'α-connexité de Zanoguera, que la couleur n'apportait rien à l'α-$\mathcal{Z}$ et enfin qu'il était préférable de traiter l'image en niveaux de gris si l'on se restreignait à l'utilisation de l'α-$\mathcal{Z}$. Cependant, nous avons montré que dans le cas de la $(P_\alpha, P_1, ..., P_n)$-$\mathcal{Z}$, l'utilisation de la couleur permettait d'obtenir de meilleurs résultats qu'en niveaux de gris.

L'extension des ZQP aux données vidéo n'avait pas encore été étudiée. Nous avons montré les lacunes de l'extension triviale 3D et proposé un nouveau cadre générique pour la construction incrémentale de ZQP. Ce cadre a été utilisé pour proposer deux approches différenciant le traitement des dimensions spatiales et temporelle : les approches 2D+t et t+2D. Ces deux approches permettent d'obtenir des résultats nettement meilleurs que l'approche 3D. Les ZQP produisant une importante sur-segmentation, une étape de filtrage est nécessaire. Nous avons étudié les approches existantes et nous nous en sommes inspirés pour élaborer une nouvelle méthode permettant de combler leurs lacunes. Nous avons proposé deux solutions plus intuitives pour régler le seuil d'aire minimale nécessaire à notre méthode de filtrage. Puis nous avons développé et comparé deux méthodes équivalentes en termes de résultats pour le filtrage vidéo des ZQP. Enfin, nous avons présenté et évalué nos différentes contributions à l'amélioration calculatoire de la production de ZQP.

Nous disposons à présent d'un outil efficace pour produire puis filtrer des ZQP couleur sur des séquences vidéo. Il nous faut à présent développer des méthodes pour permettre l'assemblage de ces pièces de puzzle afin d'obtenir les objets d'intérêt que désire l'utilisateur. C'est le sujet que nous allons aborder dans le chapitre suivant.

Chapitre 2

Segmentation vidéo interactive

Sommaire

2.1 Introduction

Les méthodes de segmentation de séquences vidéo, qu'elles soient automatiques ou nécessitant le réglage de paramètres, permettent d'obtenir des régions homogènes au sens de certains critères prédéfinis. Ces régions ne correspondent que rarement à des objets vidéo, et sont souvent peu pertinentes par rapport aux désirs d'un utilisateur donné. En effet, la notion d'objets réels représentés dans une séquence vidéo est subjective et dépend du contexte, ce dernier étant propre à chaque utilisateur. De plus, les méthodes automatiques se heurtent au problème de la grande variété d'objets réels pouvant être observés dans une séquence, ce qui à notre avis limite fortement leur développement. Les approches actuelles s'orientent donc vers des méthodes intégrant l'utilisateur aux différentes phases du processus de segmentation. Ces méthodes qualifiées d'« interactives » ou « guidées » autorisent l'utilisateur à agir à différents moments :

- avant la segmentation, par l'introduction d'informations/connaissances qui seront utilisées comme paramètres de la méthode de segmentation ;
- pendant la segmentation, par la modification des informations/connaissances précédemment introduites et/ou une action directe sur les régions produites (fusion, division, correction, etc.) ;
- après la segmentation, par la correction des régions produites.

Pendant ces périodes d'interaction, l'utilisateur peut effectuer des :

- ajouts de connaissances structurées externes (avant ou pendant) ;
- marquages d'objets-vidéo (avant ou pendant) ;
- fusions des régions pour former des objets-vidéo (pendant ou après) ;
- ...

Un des points critiques d'une approche interactive est le temps de réponse de la méthode après la ou les actions de l'utilisateur. La majorité des méthodes opèrent directement au niveau pixel, ce qui entraine un temps de réponse élevé au vu du volume de données à traiter. Pour contourner ce problème, d'autres méthodes, appelées approches « superpixels », opèrent à partir d'une sur-segmentation initiale. Cette sur-segmentation est intéressante car elle est composée de régions homogènes qui sont ensuite assemblées comme des pièces de puzzle pour obtenir des objets-vidéo. Autrement dit, au lieu d'effectuer la segmentation sur les pixels, on l'effectue sur ces régions homogènes. On réduit ainsi le coût calculatoire car le processus de segmentation est effectué sur un volume de données plus restreint. L'utilisateur effectue alors ces interactions sur les pixels de la séquence vidéo mais elles sont utilisées sur la sur-segmentation. Bien que la segmentation de séquences vidéo en superpixels soit un domaine de recherche d'actualité [DM09, SZR11], nous pouvons noter que les méthodes proposées s'appuyent généralement sur des superpixels construits par des méthodes classiques, telles que la ligne de partage des eaux [VS91] ou l'algorithme du mean-shift [CM02]. Puisque les zones quasi-plates peuvent également être considérées comme des superpixels, nous proposons ici une méthode pour la création guidée d'objets-vidéo s'appuyant sur une sur-segmentation initiale des données en zones quasi-plates.

Dans ce chapitre nous présentons un état de l'art sur les méthodes de segmentation guidée de séquences vidéo. Puis, nous étudions comment évaluer les méthodes de segmentation, lorsqu'elles sont définies dans un tel cadre « guidé ». Ensuite, nous proposons une méthode de segmentation guidée basée sur les zones quasi-plates. Finalement nous proposons de rendre cette méthode interactive avant de conclure le chapitre.

2.2 État de l'art en segmentation guidée

La segmentation vidéo guidée/interactive est un domaine de recherche très actif. Cette section n'a pas pour but de recenser toutes les méthodes existantes de segmentation vidéo interactive. Nous allons étudier et distinguer trois types de méthode de segmentation interactive/guidée : les méthodes morphologiques qui reposent sur les fondements de la morphologie mathématique, les méthodes basées sur des graphes d'adjacence de régions qui reposent sur une sur-segmentation initiale affinée par des fusions de régions, et les méthodes basées sur des coupes de graphe qui ont été très utilisées ces dernières années.

2.2.1 Approches morphologiques

La morphologie mathématique a produit de nombreuses méthodes de segmentation interactive. Dans un premier temps ces méthodes ont été développées pour des images statiques, puis des méthodes ont été développées pour les séquences vidéo.

La méthode de segmentation morphologique interactive la plus connue est la ligne de partage des eaux guidée par marqueurs [RBD92] qui est une version interactive (sauf si l'on défninit automatiquement les marqueurs) de la ligne de partage des eaux classique [VS91]. Elle consiste à faire dessiner des marqueurs sur les objets d'intérêt et sur le fond. Ces marqueurs sont ensuite utilisés comme minima dans une image que l'on considère comme un relief. On fait ensuite monter le niveau de l'eau à partir des minima définis par les marqueurs. On augmente ainsi la surface des bassins (et implicitement la taille des objets d'intérêt recherchés par l'utilisateur). Lorsque le niveau de l'eau a suffisamment augmenté pour que deux régions se joignent, on construit une digue pour éviter que les eaux se mélangent. Cette digue représente la ligne de partage des eaux et correspond à la frontière des objets d'intérêt. Pour l'appliquer trivialement aux séquences vidéo, il suffit de traiter indépendamment chaque trame de la séquence vidéo en demandant à l'utilisateur de dessiner des marqueurs sur chacune d'entre elles. On obtient ainsi une séquence temporelle de segmentations. Bien que simple, cette solution souffre d'un défaut évident. L'utilisateur doit en effet dessiner des marqueurs pour chacune des trames, ce qui s'avère fastidieux dès lors que la durée de la séquence vidéo dépasse quelques secondes. Une autre solution est de ne plus considérer le gradient comme une information de relief mais comme un volume 3D représentant la porosité de chaque voxel. On

introduit alors l'eau dans chaque minimum et on augmente non plus le niveau de l'eau mais la pression pour agrandir les régions. Ainsi, l'information apportée par les marqueurs est non plus seulement propagée spatialement mais également temporellement, ne nécessitant ainsi pas de dessiner des marqueurs sur chacune des trames. Cependant, si la justesse de la segmentation est très forte sur les trames marquées, elle s'estompe au fur et à mesure que l'on s'éloigne d'une trame marquée. Il est donc nécessaire de marquer un nombre important de trames pour pouvoir guider efficacement la segmentation.

Flores et Lotufo [FL10] ont proposé une autre approche de segmentation interactive de séquences vidéo utilisant la ligne de partage des eaux guidée par marqueurs. L'utilisateur dessine des marqueurs pour segmenter la première trame de la séquence vidéo. La première trame est segmentée par la ligne de partage des eaux guidée par marqueurs. De nouveaux marqueurs sont extraits de cette segmentation : ils correspondent aux contours obtenus après érosion et dilatation des objets segmentés. Nous obtenons donc un marqueur interne et un marqueur externe pour chaque objet. Ces marqueurs sont ensuite fractionnés en petites régions et liés : un marqueur interne est lié au marqueur externe qui lui est symétrique par rapport au contour de l'objet. Chaque couple de marqueurs est propagé à la trame suivante par une estimation de mouvement. Les marqueurs propagés sont utilisés pour obtenir la segmentation de la trame suivante. L'utilisateur peut éditer les marqueurs (correction/suppression/ajout) afin d'affiner la segmentation. Ce processus est répété jusqu'à ce que toutes les trames soient segmentées. Cette méthode nécessite une implication importante de l'utilisateur qui doit vérifier la segmentation de toutes les trames au fur et à mesure de l'avancement du processus, afin de pouvoir corriger les éventuelles erreurs de propagation de marqueurs. Cette méthode ne permet de segmenter qu'un seul objet d'intérêt dans une séquence vidéo.

Marcotegui *et al.* [MZC+99] utilisent la ligne de partage des eaux pour produire une segmentation multi-échelle. Le niveau le plus bas correspond aux régions obtenues par une ligne de partage des eaux. Le niveau le plus haut correspond à la région contenant toute l'image. Les différentes échelles sont obtenues en fusionnant les régions voisines de la segmentation par ligne de partage des eaux selon le critère de la valeur d'extinction [VM95] combinant taille des régions et contraste inter-régions. L'utilisateur dispose ensuite de plusieurs outils pour obtenir la segmentation désirée, il peut ainsi sélectionner le nombre de régions, raffiner une région, fusionner des régions, dessiner des marqueurs ou ajuster les contours. Une fois la trame initiale segmentée de cette façon, la segmentation est propagée à la trame suivante en utilisant une méthode de suivi d'objets [ML98]. Cette méthode permet également de détecter l'apparition de nouvelles régions et décide de les affecter ou non à l'objet suivi. L'utilisateur peut intervenir dans ce processus de deux façons : en modifiant les affectations de nouvelles régions effectuées par la méthode et/ou en utilisant les mêmes outils que pour la segmentation de la trame initiale. Le système proposé par les auteurs dispose également d'une méthode de segmentation des objets en mouvement basée sur une détection de changements et une analyse de mouvement. Cette méthode permet la segmentation automatique d'objets en mouvement mais permet également à l'utilisateur d'intervenir pour améliorer le résultat. L'interactivité est possible de plusieurs manières : soit en segmentant manuellement la première trame pour indiquer les régions en mouvement à suivre soit en fournissant l'image du fond sans les objets (ce qui améliorera le processus d'extraction des objets). Cette méthode nécessite une implication lourde de l'utilisateur, qui doit vérifier le résultat de segmentation dans chaque trame.

Gu et Lee [GL98] ont également utilisé la ligne de partage des eaux guidée par marqueurs. La méthode consiste à demander à l'utilisateur de dessiner les frontières de l'objet qui l'intéresse. Fournir les frontières exactes de l'objet étant un procédé coûteux en temps, on demande à l'utilisateur un tracé approximatif. Afin d'obtenir ensuite les frontières réelles et précises de l'objet, la méthode définit un marqueur extérieur et un marqueur intérieur obtenus respectivement par une dilatation et une érosion de la forme définie par le contour dessiné par l'utilisateur. Ces deux marqueurs sont utilisés par un algorithme de ligne de partage des eaux pour segmenter l'objet d'intérêt dans la trame initiale. Cette segmentation initiale est ensuite propagée à la trame suivante par une méthode de suivi d'objet basée sur l'estimation du mouvement. Ce processus est répété jusqu'à ce que l'utilisateur resegmente manuellement une trame en dessinant les contours de l'objet d'intérêt.

Cette approche ne permet l'extraction que d'un seul objet d'intérêt et travaille au niveau du pixel (le traitement pourrait être accéléré en travaillant sur une sur-segmentation).

Mise à part l'extension directe de la ligne de partage des eaux guidée par marqueurs à la vidéo, les méthodes morphologiques résolvent le problème de la segmentation interactive de vidéo en propageant une segmentation initiale dans les trames suivantes.

2.2.2 Approches basées sur des graphes

L'utilisation de structures de données de type graphe pour la segmentation de séquences vidéo est de plus en plus fréquente. Cet engouement s'explique par la possibilité de représenter facilement des régions et leurs relations d'adjacence. La description et la comparaison de régions adjacentes ainsi que leur fusion sont des opérations simples et rapides au sein d'un graphe. Nous allons étudier deux types d'utilisation de graphe dans le cadre de la segmentation interactive de séquences vidéo : les méthodes utilisant des graphes d'adjacence de régions et les méthodes par coupe de graphe.

2.2.2.1 Graphe d'adjacence de régions

Liu *et al.* [LYP05] ont proposé une méthode interactive de segmentation de séquences vidéo basée sur la ligne de partage des eaux classique [VS91] et l'algorithme *seeded region growing* [AB94]. Il s'agit dans un premier temps de produire une sur-segmentation de la trame initiale grâce à la ligne de partage des eaux. On demande ensuite à l'utilisateur de dessiner des marqueurs sur l'objet d'intérêt et sur le fond. Les régions de la sur-segmentation sont utilisées pour créer un graphe d'adjacence de régions. Une région est décrite par sa couleur moyenne et son aire. Les arêtes du graphe représentant les liens d'adjacence entre les régions sont valuées de la différence entre les couleurs moyennes des deux régions qu'elles relient, pondérée par l'aire de la plus petite des deux régions. C'est sur sur les régions de ce graphe que va être appliqué le *seeded region growing*. On affecte l'étiquette *objet* aux régions qui sont seulement marquées par des marqueurs de l'objet, *fond* aux régions qui sont seulement marquées par des marqueurs du fond et *inconnu* pour celles qui ne sont pas marquées ou marquées par les deux types de marqueurs. L'algorithme va alors fusionner les régions *inconnu* aux régions *objet* et *fond* par priorité de valeurs minimales des arêtes, privilégiant aussi la fusion des petites régions. La segmentation obtenue est présentée à l'utilisateur qui peut alors la corriger en agissant sur l'affectation des régions de la sur-segmentation initiale à l'objet ou au fond. Cette segmentation de la trame initiale est ensuite projetée sur la trame suivante pour y être adaptée en tenant compte du mouvement et des modifications morphologiques de l'objet. Ce processus de projection est répété jusqu'à ce que l'intégralité de la séquence vidéo soit segmentée. L'application du *seeded region growing* aux régions issues d'une sur-segmentation améliore l'efficacité calculatoire de la méthode. Néanmoins, la méthode ne prévoit pas de processus permettant de corriger cette sur-segmentation, ce qui rend impossible la correction de régions, ne permettant donc pas à l'utilisateur d'obtenir les objets d'intérêt qu'il recherche. A l'instar des autres méthodes $2D + t$ précédentes, l'utilisateur fait face à une charge de travail importante car il doit vérifier chaque trame afin de pouvoir corriger la segmentation dans le but de ne pas propager d'erreurs aux trames suivantes.

Grundmann *et al.* [GKHE10] ont développé une méthode de segmentation utilisant une adaptation aux séquences vidéo de [FH04] comme sur-segmentation initiale. Ils commencent par produire une sur-segmentation de la vidéo originale. Cette sur-segmentation est obtenue en considérant la séquence vidéo comme un graphe d'adjacence de régions représentant la séquence vidéo comme un cube spatio-temporel et où chaque pixel est un nœud, les liens d'adjacence sont spatio-temporels et tiennent compte du flot optique. Un pixel n'est pas adjacent aux pixels d'une trame adjacente selon sa position dans la trame courante mais selon sa position estimée par le flot optique dans la trame adjacente. Ce graphe représente une sur-segmentation extrême, et est réduit en fusionnant les pixels adjacents similaires pour créer une sur-segmentation initiale moins extrême. La granularité de cette segmentation est déterminée par un paramètre et une taille minimale de région est fixée. La sur-segmentation initiale obtenue est représentée par un graphe d'adjacence de régions dont les nœuds sont décrits par un histogramme des valeurs des pixels de la région dans l'espace

Lab et le flot optique. Les arêtes de ce graphe sont valuées par une combinaison des distances χ^2 entre les histogrammes et entre les valeurs de flot optique des régions qu'elles relient. Cette sur-segmentation est utilisée pour construire une segmentation hierarchique obtenue en augmentant itérativement le paramètre de granularité et le nombre minimum de pixels par région. Pour obtenir une segmentation personnalisée, les auteurs proposent que l'utilisateur choisisse l'échelle et sélectionne les régions représentant les objets d'intérêt. Cependant le choix de l'échelle à considérer sur l'intégralité de la séquence vidéo n'est pas trivial et cliquer sur toutes les régions représentant un objet peut s'avérer fastidieux.

2.2.2.2 Coupes de graphe

Les coupes de graphe sont depuis une dizaine d'années utilisées dans le cadre de la segmentation interactive de séquences vidéo. Une coupe de graphe consiste à supprimer des arêtes selon certains critères prédéfinis afin d'obtenir plusieurs composantes connexes de nœuds du graphe.

Boykov et Jolly [BJ01] ont proposé une méthode interactive de coupe de graphes permettant d'effectuer des segmentations dans les images à n dimensions et donc dans les séquences vidéo. Ils considèrent la séquence vidéo comme un cube et créent un graphe la représentant, où chaque pixel de la séquence vidéo est un nœud et où les arêtes représentent les liens d'adjacence entre les pixels selon la 26-connexité (chaque pixel a 8 voisins dans sa trame et 9 dans chaque trame adjacente). Les arêtes sont également valuées d'une mesure de similarité entre les pixels qu'elles relient. L'utilisateur dessine des marqueurs de deux types dans la séquence vidéo : *fond* et *objet*. Tous les pixels sous un marqueur sont marqués comme étant du même type que le marqueur. Le principe de la segmentation va consister à trouver la coupe de valeur minimale qui séparera les nœuds *objet* des nœuds *fond*. Il s'agit donc de supprimer des arêtes afin de constituer des composantes connexes de nœuds ne comportant soit que des nœuds objet et des nœuds non étiquetés, soit que des nœuds fond et des nœuds non étiquetés. L'algorithme utilisé est un algorithme de flot maximum présenté dans [BK04]. Cette méthode s'avère efficace mais non applicable à des séquences vidéo de grande taille. En effet, tous les pixels sont traités, ce qui nécessite un volume important de mémoire. En outre, cette méthode ne permet de segmenter qu'un seul objet d'intérêt dans la séquence vidéo.

Price *et al.* [PMC09] proposent une méthode de segmentation interactive par propagation de segmentation. Le principe est de segmenter chaque trame en utilisant des coupes de graphe selon plusieurs critères et en utilisant la segmentation de la trame précédente. Afin d'accélérer le traitement, les coupes de graphes ne sont pas effectuées sur les pixels mais sur les régions d'une sur-segmentation par ligne de partage des eaux. L'utilisateur dessine des marqueurs sur la première trame afin de marquer l'objet d'intérêt. La propagation de la segmentation de la trame courante à la trame suivante va guider la coupe de graphe de cette trame. Les informations de mouvement du fond et de la forme sont prises en compte de même que le gradient, la couleur, les couleurs adjacentes, la cohérence spatio-temporelle et les informations de forme. Une méthode de suivi de points particuliers est également utilisée. Ces différents critères sont pondérés au fur et à mesure de la propagation de trames via un processus d'apprentissage prenant en compte les éventuelles corrections de l'utilisateur. La pondération des différents critères en fonction de la validation implicite d'un résultat ou des corrections de l'utilisateur semble donner de bons résultats. Cependant, elle implique une réelle vérification de la segmentation par l'utilisateur sur chacune des trames afin de bien guider la pondération est de ne pas introduire d'erreurs. L'utilisation d'une sur-segmentation accélère le traitement mais cette sur-segmentation ne peut être corrigée, ce qui peut s'avérer facheux en cas d'erreurs dans la sur-segmentation.

Les méthodes que nous avons présentées incluent principalement trois types d'interactions : dessin de marqueurs, action sur des régions existantes (fusion/raffinement/division/affectation...) et choix d'une échelle dans une segmentation multi-échelle. Ces méthodes souffrent de différents inconvénients. L'inconvénient principal concerne les méthodes ne pouvant segmenter qu'un seul objet d'intérêt dans une séquence vidéo. Cela implique que, dans le cas où un utilisateur est intéressé par plusieurs objets d'intérêt, il devra traiter plusieurs fois la séquence vidéo. Ensuite, de nombreux

traitements s'effectuent directement sur les pixels de la séquence vidéo, ce qui impose un coût important en mémoire. Même si l'on utilise une approche trame par trame, traiter tous ces pixels dans un processus de segmentation est long et coûteux alors qu'effectuer les traitements sur une pré-segmentation accélère le traitement car il est effectué sur un volume de données plus restreint. Les approches par propagation de trames nécessitent un investissement lourd de l'utilisateur qui doit vérifier et corriger si nécessaire les segmentations de chaque trame pour éviter que les erreurs de segmentation ne se propagent aux trames suivantes. Enfin, certaines méthodes utilisant une sur-segmentation ne permettent pas de corriger les régions de cette sur-segmentation, ce qui peut être bloquant si l'assemblage de ces régions ne fournit pas les objets d'intérêt voulus par l'utilisateur.

2.3 Évaluation de segmentation vidéo interactive

Dans cette section, nous présentons les méthodes que nous utilisons pour l'évaluation de la segmentation interactive. Dans un premier temps, nous expliquons pourquoi nous ne pouvons utiliser les métriques présentées dans la section 1.2.2, puis nous présentons les méthodes utilisées pour l'évaluation du résultat de la segmentation et celles utilisées pour évaluer le processus interactif.

2.3.1 Pourquoi des métriques différentes de celles utilisées pour les ZQP ?

Les métriques d'évaluation définies dans la section 1.2.2 ne sont pas adaptées à l'évaluation de segmentation guidée. En effet, le ratio de sur-segmentation et la précision maximale sont des mesures adaptées à l'évaluation de sur-segmentation. Or, les segmentations obtenues par un processus interactif peuvent être considérées comme des segmentations finales et adaptées (à terme) aux désirs de l'utilisateur. L'évaluation par ratio de sur-segmentation est donc inadaptée car il serait de 1 dans un tel cas, sauf si l'utilisateur ne désire pas le même nombre d'objets que la segmentation de référence ce qui, dans ce cas, poserait le problème de l'inadéquation de la référence. Dans l'absolu, la précision maximale est applicable à l'évaluation du résultat d'une segmentation interactive où elle représenterait le pourcentage de pixel appartenant à des régions identiques dans la segmentation et dans la référence. Or, s'il était cohérent d'évaluer la précision maximale d'une sur-segmentation par une méthode calculant le pourcentage global de pixels bien classés, cela n'est plus cohérent pour une segmentation finale en objets. En effet, cette métrique mesure la précision globale de la segmentation et ne permet pas de distinguer les résultats de chaque région ou objet dans notre cas. On pourrait l'appliquer pour chaque région mais cette métrique mesurerait la précision au sein de la région à évaluer et ne prendrait pas en compte les pixels appartenant à la référence mais ne se trouvant pas dans la région à évaluer. La précision maximale n'est donc pas la métrique idéale pour évaluer la précision de la segmentation finale en objets. Si l'évaluation du résultat de la segmentation est important, il faut également disposer d'un moyen permettant d'évaluer le processus interactif permettant à l'utilisateur de personnaliser la segmentation. Dans les sections suivantes, nous présentons les métriques que nous avons retenues pour l'évaluation du résultat de la segmentation et du processus interactif.

2.3.2 Évaluation de la segmentation

La première chose que l'on souhaite évaluer au niveau du résultat de la segmentation est la précision des objets obtenus par rapport à une segmentation de référence. Nous avons vu dans la section précédente que la précision maximale utilisée pour évaluer une sur-segmentation n'était pas adaptée à l'évaluation d'une segmentation finale en objets car elle calculait une mesure de précision globale de la segmentation et non une précision de chaque objet. Pour évaluer chaque objet, on utilise l'indice de Jaccard [Jac01], conçu à l'origine pour mesurer la diversité des espèces dans deux échantillons de population. Il est également utilisé pour évaluer des segmentations d'image [GWL07, MO10]. Il est défini comme le ratio entre le nombre de pixels correctement segmentés d'un objet à évaluer par rapport à une segmentation de référence et le nombre de pixels de l'union de l'objet à évaluer et de l'objet dans la segmentation de référence. Formellement, on note l'indice de Jaccard (IJ) tel que :

$$IJ = \frac{\text{card}(obj \cap ref)}{\text{card}(obj \cup ref)} \tag{2.1}$$

où obj est l'ensemble des pixels de l'objet à évaluer. Cette mesure nous permet d'évaluer indépendamment la qualité de la segmentation d'un objet. Si nous désirons obtenir une mesure globale pour la segmentation, il suffit de combiner les différentes mesures obtenues indépendamment en calculant un indice de Jaccard moyen (IJM) selon la moyenne pondérée suivante :

$$IJM = \frac{\sum (IJ_i \times Pmt_i)}{\sum Pmt_i} \tag{2.2}$$

où IJ_i est l'indice de Jaccard de l'objet i et Pmt_i le nombre moyen de pixels par trame de l'objet i. On aurait pu normaliser cette moyenne en utilisant le nombre de pixels de chaque objet. Cependant, un objet spatialement étendu mais n'apparaissant que dans quelques trames de la séquence vidéo aurait été peu pris en compte car son nombre de pixels pourrait être plus restreint que celui d'un petit objet présent tout au long de la séquence. Une mauvaise segmentation d'un tel objet aurait peu pénalisé l'indice de Jaccard moyen. Une séquence vidéo n'étant pas réellement $3D$ mais étant plutôt une représentation $2D$ accompagnée de son évolution au cours du temps, nous préférons privilégier la dimension spatiale dans la normalisation de la moyenne.

Dans le contexte de la segmentation interactive de séquences vidéo, il est également pertinent d'étudier l'évolution de la précision de la segmentation au cours du temps [VM04, FL10]. En effet, l'indice de Jaccard nous donne une évaluation globale de la qualité de l'objet par rapport à la segmentation de référence mais ne nous donne aucune information sur sa constance ou inconstance au cours du temps. Il est pourtant intéressant d'avoir ces informations pour mesurer l'impact de l'interactivité de l'utilisateur dans les trames où il n'est pas intervenu. Ces informations permettent également de savoir si une mauvaise mesure de qualité d'un objet est constante dans le temps, ce qui induirait soit les mauvaises performances de l'algorithme de segmentation ou l'insuffisance voire l'inadéquation de l'interaction de l'utilisateur. Dans le cas où la mesure de qualité n'est pas constante, cela traduit une mauvaise segmentation de l'objet sur certaines trames qui peut être réglée par une intervention de l'utilisateur sur ces trames. Pour obtenir cette qualité par trame, il suffit de modifier l'équation 2.1 pour obtenir l'équation :

$$IJ_t = \frac{\text{card}(obj_t \cap ref_t)}{\text{card}(obj_t \cup ref_t)} \tag{2.3}$$

où IJ_t, obj_t et ref_t sont respectivement l'indice de Jaccard, l'ensemble des pixels de l'objet à évaluer et l'ensemble des pixels de l'objet de référence dans la trame t.

Un dernier critère très important en segmentation interactive est le temps de calcul nécessaire pour obtenir la segmentation en objets après intervention de l'utilisateur. Ce critère est important dans tous les types de segmentation, mais il l'est encore plus en segmentation interactive. En effet, l'utilisateur va probablement vouloir attendre le résultat de son intervention afin de pouvoir éventuellement la corriger si le résultat ne lui convient pas. De plus, les séquences vidéo étant par nature volumineuses, leur traitement va nécessiter un temps important. L'évaluation du temps de calcul devient dès lors primordiale. Un utilisateur privilégiera une méthode produisant des résultats rapides à une méthode produisant de meilleurs résultats mais nécessitant beaucoup plus de temps de calcul. Nous distinguerons deux types de temps de calcul : le temps de calcul hors-ligne correspondant au pré-traitement de la séquence vidéo pouvant être effectué sans l'utilisateur, et le temps de calcul en-ligne correspondant au temps nécessaire pour segmenter la séquence vidéo après intervention de l'utilisateur. Nous rappelons que l'idéal pour une méthode est d'effectuer le maximum de calculs hors-ligne afin d'être le plus rapide possible en-ligne.

2.3.3 Évaluation de l'interactivité

Outre la qualité des objets obtenus par le processus de segmentation interactive, il est important d'évaluer le processus en lui-même. Le premier élément que l'on va chercher à évaluer est

l'implication de l'utilisateur dans le processus interactif. Pour cela, on peut utiliser plusieurs métriques [FL10] telles que le nombre d'interventions de l'utilisateur (par trame ou global) ou encore le temps passé par l'utilisateur dans le processus interactif. Il s'agit d'une évaluation simple mais pertinente du coût représenté par l'interaction pour l'utilisateur.

Parallèlement, une propriété importante en segmentation interactive de séquences vidéo est l'influence temporelle d'une interaction. Il s'agit de mesurer, lorsqu'on interagit sur une trame, l'évolution temporelle de l'apport de cette interaction en terme de qualité. Une méthode serait idéale si elle ne nécessitait d'interagir que sur une seule trame pour segmenter correctement toute une séquence vidéo. On tend donc à développer des méthodes nécessitant le minimum d'interactions, c'est-à-dire que les interactions dans une trame se propagent durablement dans les trames suivantes. Nous étudierons cette capacité en interagissant sur une trame et en observant comment évolue la qualité des objets obtenus lorsque l'on s'éloigne temporellement de cette trame.

2.4 Construction de ZQP guidée par marqueurs

Dans le chapitre 1, nous avons développé une méthode permettant de produire des pièces de puzzle spatio-temporelles. Ces pièces de puzzle doivent permettre, par leur assemblage, de créer les objets que l'utilisateur souhaite obtenir. Il est donc nécessaire de disposer d'une méthode permettant leur assemblage.

Afin d'assembler les ZQP, nous utilisons une méthode proche de celle que nous avons employée pour le filtrage des ZQP (cf. section 1.6). Pour le filtrage, nous avions utilisé les ZQP ayant une taille (ou aire moyenne) supérieure à un seuil comme graines pour un algorithme de *seeded region growing*. Nous reprenons cette idée, et l'utilisateur designera certaines ZQP comme graines ; nous ferons ensuite croitre ces graines par fusion avec les régions voisines les plus similaires selon un certain critère, par exemple la couleur moyenne. Nous utilisons le même algorithme de SRG que pour le filtrage.

Nous ne demandons pas à l'utilisateur de sélectionner une à une les ZQP qui seront les graines du SRG, puisque ce serait fastidieux et surtout peu intuitif. Nous allons plutôt demander à l'utilisateur de dessiner des marqueurs sous forme de gribouillis. La figure 2.1 illustre ce mode d'interaction : l'utilisateur dessine des gribouillis de différentes couleurs (une couleur par objet) afin de marquer les objets d'intérêt et le fond de la séquence vidéo. Nous utilisons ensuite la position de ces marqueurs pour déterminer les ZQP graines.

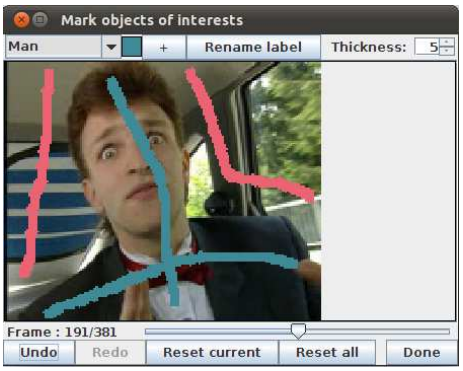


FIGURE 2.1 – Exemple de marqueurs sous forme de gribouillis.

La figure 2.2 présente la méthode de segmentation de séquences vidéo par ZQP guidée par marqueurs. Cette méthode comporte de deux étapes :

1. La première étape se compose de deux processus effectués en parallèle, l'un automatique, l'autre incluant l'utilisateur :

- Automatique : La séquence vidéo est segmentée en ZQP spatio-temporelles qui sont ensuite filtrées afin de fournir une sur-segmentation raisonnable permettant par fusion de ZQP de constituer les objets voulus par l'utilisateur. Cette sur-segmentation est ensuite transformée en graphe d'adjacence de ZQP. Les ZQP sont transformées en nœuds et sont décrites par un attribut, par exemple leur couleur moyenne. Les arêtes représentent les liens d'adjacence spatio-temporelle entre les ZQP.
- utilisateur : L'utilisateur dessine des gribouillis sur les objets d'intérêt. Ces gribouillis sont ensuite extraits et utilisés comme marqueurs dans la suite de la méthode.

2. La seconde étape consiste en premier lieu en la fusion des informations issues de la première étape. Pour cela, on plonge les marqueurs entrés par l'utilisateur dans le graphe d'adjacence des ZQP. Les nœuds représentant des ZQP sous les marqueurs sont étiquetés comme « graines ». L'algorithme de SRG utilisé est le même que pour le filtrage, seule la façon dont les nœuds graines sont déterminés est différente. Nous avons autant de nœuds graines qu'il y a de ZQP graines sous les marqueurs. Nous aurions pu fusionner tous les nœuds représentant le même marqueur et mettre à jour les liens d'adjacence du graphe. Cependant, dans le cas d'objets d'intérêt texturés, une telle stratégie aurait produit une couleur moyenne du nœud fusionné non représentative. En utilisant indépendamment chaque nœud graine, nous disposons d'un ensemble de couleurs moyennes qui représente mieux un objet composé de couleurs hétérogènes et permet donc de mieux extraire l'objet marqué par l'utilisateur. À la fin du SRG, les étiquettes des nœuds sont plongées dans la séquence vidéo afin d'obtenir la segmentation de la séquence.

La partie automatique et la partie utilisateur de la première étape, bien que parallèles, peuvent être effectuées séquentiellement de façon à obtenir une partie hors-ligne et une partie en-ligne. Nous effectuons ainsi préalablement la partie automatique consistant à produire les ZQP, à les filtrer et à créer le graphe d'adjacence. Ces traitements sont les plus coûteux du point de vue calculatoire et la présence de l'utilisateur n'est pas nécessaire. En effectuant tous ces traitements préalablement, on fait gagner du temps à l'utilisateur qui n'a alors plus qu'à dessiner les marqueurs. Le SRG opéré à partir des marqueurs de l'utilisateur est rapide car effectué sur une réduction de données importante et dans une structure qui gère efficacement l'adjacence, un graphe. L'utilisateur obtient rapidement la segmentation correspondant à son guidage par marqueurs. Nous avons mesuré le temps d'éxécution sur la séquence *carphone* (176x144 pixels sur 381 trames soit 9 656 064 pixels couleur). Ces tests ont été effectués sur un ordinateur portable avec un processeur Intel Core i7 Q720 cadencé à 1,6 GHz et disposant de 4 Go de mémoire vive. Nous avons mesuré et comparé les temps de calculs obtenus pour la segmentation en zones quasi-plates guidée par marqueurs pour différentes valeurs de (α, ω) en utilisant l'approche 2D+t et l'approche t+2D. Les résultats sont présentés dans la table 2.1. Les tests confirment que, si la partie hors-ligne est coûteuse en temps de calcul, la partie en ligne qui correspond à la seconde étape de la méthode est rapide et ce, quels que soient les paramètres (α, ω) et l'approche utilisée. En effet, ce temps de calcul est largement inférieur à la seconde pour la séquence *carphone* qui dure pourtant 31 secondes (avec une cadence égale à 12 images/seconde). Ces résultats valident l'intérêt d'utiliser une réduction de l'espace de données initial que représente la séquence vidéo. En effet, en appliquant la segmentation guidée sur le graphe d'adjacence des ZQP filtrées, le traitement est très rapide et l'utilisateur peut ainsi obtenir la segmentation qu'il désire quasi-immédiatement. Cette méthode peut donc être utilisée dans un contexte de segmentation réellement interactive, comme nous le verrons dans la section suivante.

Nous avons également comparer le coût calculatoire de notre approche à ceux obtenus par la ligne de partage des eaux guidée par marqueurs et au seeded region growing. Ces deux méthodes n'utilisent, à l'instar de notre approche, aucune information de mouvement et ne se basent que sur l'information chromatique et la connexité pour effectuer la segmentation. La ligne de partage des eaux guidée par marqueurs (LPEGM) est la méthode de segmentation guidée par l'utilisateur la plus utilisée en Morphologie Mathématique. Le Seeded Region Growing (SRG) est utilisé par notre méthode d'une façon originale puisque sur les ZQP filtrées. Il est intéressant de voir si nous obtenons de meilleures performances qu'en appliquant le SRG directement sur les pixels afin de valider l'intérêt de notre approche. La méthode de SRG opère entièrement en-ligne et ne possède pas de pré-traitement des données. Elle nécessite un temps beaucoup plus important que notre

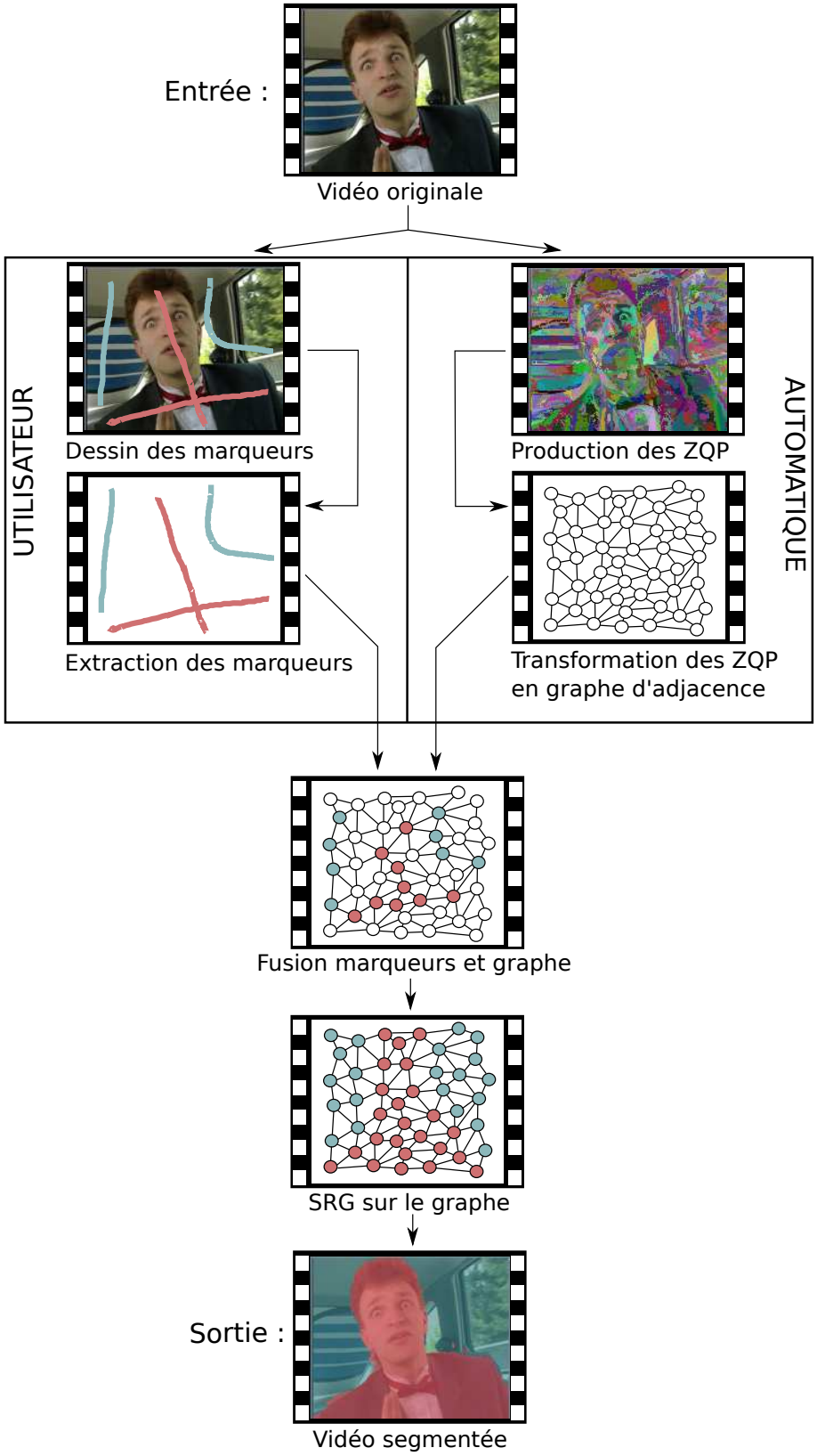


FIGURE 2.2 – Processus de segmentation vidéo par assemblage de ZQP guidé avec des marqueurs.

approche, rendant pertinent du point de vue de l'efficacité notre parti pris de travailler sur une réduction de l'espace de données. De plus, non seulement la partie en ligne de notre approche est moins coûteuse que le SRG mais son temps de calcul total (parties en ligne et hors ligne) est inférieur au temps requis par le SRG. Notre approche est donc bien plus efficace. La LPEGM, par contre, comprend une partie hors ligne et une partie en ligne. La partie hors ligne correspond à la transformation de l'espace de données original, qui dans le cadre de séquences vidéo couleurs peut également s'apparenter à une réduction puisque nous passons d'une séquence comprenant trois bandes chromatiques à une séquence ne comportant qu'une seule bande de gradient. Les temps de calcul hors-ligne de la LPEGM sont nettement inférieurs à ceux de notre approche. Les temps en ligne sont par contre très supérieurs : notre approche est donc beaucoup plus rapide que la LPEGM. En effet, alors que notre approche nécessite moins d'une seconde, la partie en ligne de la LPEGM prend plus d'une quinzaine de secondes. Notre approche se relève donc pertinente d'un point de vue du temps calculatoire. Elle est plus efficace en termes de temps de calcul que le SRG et la LPEGM, deux méthodes guidées par marqueurs qui lui sont comparables. À titre de comparaison à l'état de l'art, nous avons également étudié le temps requis par une méthode récente, celle de Price *et al.* [PMC09], également basée sur une sur-segmentation intiale, qui pour une séquence vidéo de taille 640x480 pixels annonce un temps de traitement de 1880 ms par trame (segmentation + propagation). Les auteurs ne précisent malheureusement pas sur quel matériel, ils ont obtenu ce temps de traitement.

Méthode	α, ω	# ZQP	Temps de calcul en **ms**		
			hors-ligne	en ligne (par trame)	
ZQPGM 2D+t	10	28 612	44 390	528	(1,39)
	20	30 671	35 510	550	(1,44)
	30	27 713	38 762	508	(1,33)
	40	22 202	43 280	364	(0,96)
	50	18 501	46 343	326	(0,86)
ZQPGM t+2D	10	3 772	44 781	108	(0,28)
	20	4 713	32 080	123	(0,32)
	30	4 649	26 957	116	(0,30)
	40	3 842	26 128	107	(0,28)
	50	3 147	25 133	98	(0,26)
SRG	–	–	0	56 636	(148,65)
LPEGM	–	–	3 354	17 312	(45,44)

TABLE 2.1 – Comparaison des temps de calcul hors-ligne et en ligne de l'assemblage de ZQP guidé avec des marqueurs (ZQPGM) selon différentes valeurs de (α, ω) et $\mathcal{A}_{moyenne} = 10$, du Seeded Region Growing (SRG) et de la ligne de partage des eaux guidée par marqueurs (LPEGM) sur la séquence *carphone*.

Bien que performante d'un point de vue calculatoire, la méthode que nous proposons présente encore un problème important : la gestion des ZQP couvertes par de multiples marqueurs. Les ZQP représentent une réduction de l'espace de données mais les marqueurs sont produits par l'utilisateur de façon indépendante sur les pixels. Il est donc possible qu'une ZQP soit couverte par plusieurs marqueurs. Or, une même ZQP ne peut appartenir à plusieurs objets. Instinctivement, nous percevons deux solutions (cf. figure 2.3) : supprimer les marqueurs problématiques ou segmenter la ZQP d'après les marqueurs. Ces deux approches reviennent à choisir l'information en laquelle nous avons le plus confiance. Faisons-nous confiance à la méthode automatique qui a créé les ZQP ou à l'utilisateur qui a dessiné les marqueurs ? Nous donnons la préférence à l'utilisateur qui, dans le cadre d'une segmentation personnalisée, est le seul à même de juger de la pertinence de ses marqueurs. Nous optons donc pour l'approche consistant à resegmenter les ZQP sous multiples marqueurs selon ces marqueurs. Cette approche présente également un autre avantage : elle permet d'améliorer la sur-segmentation initiale en améliorant la qualité des frontières en fonction des besoins de l'utilisateur.

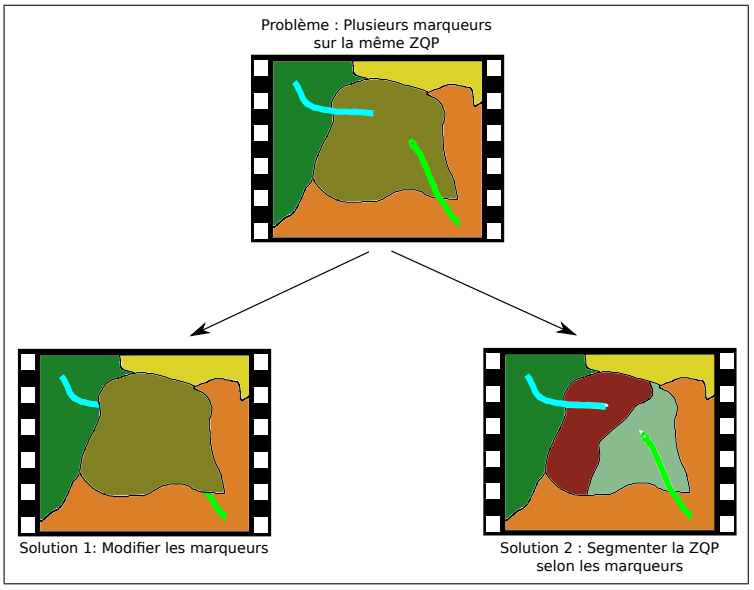


FIGURE 2.3 – Les deux approches pour régler le problème des marqueurs multiples sur une même ZQP.

Afin de segmenter les ZQP remises en cause par les marqueurs, il est nécessaire de choisir une méthode de segmentation guidée par marqueurs. La méthode choisie doit être simple pour que son coût calculatoire reste modéré. Nous proposons d'utiliser la LPEGM (basée sur un gradient spatio-temporel euclidien) ou le SRG, ces deux méthodes étant relativement efficaces d'un point de vue calculatoire. Afin de déterminer la méthode la plus adaptée, nous avons comparé les résultats obtenus sur un extrait de *carphone* de 80 trames. Nous avons dessiné des marqueurs sur les trames 15, 40 et 65. Les ZQP de la sur-segmentation initiale ont été obtenues en utilisant des valeurs α et ω comprises entre 10 et 50, ainsi qu'une aire moyenne de 10 pour le filtrage. Les indices de Jaccard moyen obtenus sont présentés dans la table 2.2.

Méthode	α, ω	Indice de Jaccard Moyen		
		Modifier marqueurs	SRG	LPEGM
ZQPGM 2D+t	10	0,956	0,963	**0,971**
	20	0,972	0,973	**0,974**
	30	0,965	0,966	**0,968**
	40	0,954	0,966	**0,973**
	50	0,955	0,965	**0,973**
ZQPGM t+2D	10	0,915	0,890	**0,971**
	20	0,936	0,938	**0,969**
	30	0,958	0,934	**0,974**
	40	0,947	0,925	**0,976**
	50	0,916	0,915	**0,969**

TABLE 2.2 – Comparaison des indices de Jaccard Moyen obtenus par les différentes méthodes de correction de ZQP sur l'extrait de 80 trames de la séquence *carphone* avec les trames 15,40 et 65 marquées.

Les mesures obtenues montrent que la meilleure méthode de correction des ZQP est la LPEGM. En effet, les résultats obtenus par le SRG et par la modification de marqueurs sont inférieurs en qualité à ceux de la LPEGM. Nous utiliserons donc cette méthode pour resegmenter les ZQP remises en cause par les marqueurs.

Nous avons évalué la performance de la segmentation en ZQP guidée par marqueurs d'un point

de vue calculatoire. Il faut également évaluer sa performance en terme de qualité de résultat et la comparer à celle de méthodes proches. Dans ce contexte, nous utilisons, une nouvelle fois, la LPEGM et le SRG à des fins de comparaison. Pour cette comparaison, nous avons utilisé l'extrait de 80 trames de la séquence *carphone* et l'extrait de 40 trames de *foreman*. Sur ces extraits nous avons dessiné 2 jeux distincts de marqueurs présentés dans la figure 2.4. Les marqueurs des deux jeux ne sont définis que sur la trame médiane de l'extrait : le premier jeu ne comprend que quelques carrés, le second jeu est composé de gribouillis et comporte plus de pixels marqués. L'utilisation de deux jeux distincts permettra d'évaluer la robustesse de notre approche quant au choix des marqueurs.

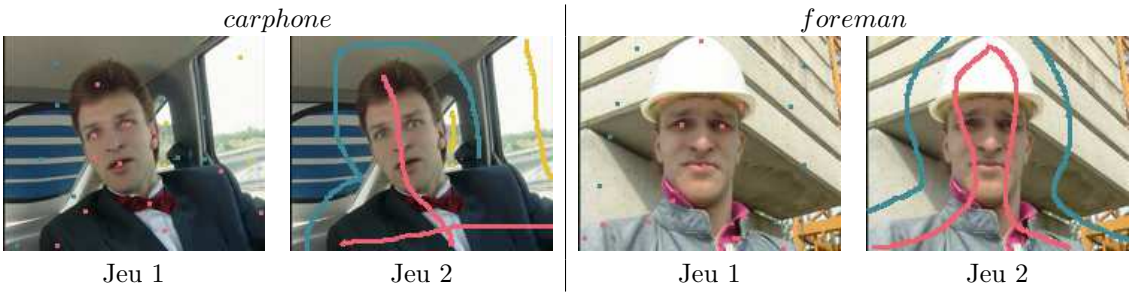


FIGURE 2.4 – 4 jeux de marqueurs utilisés sur un extrait de la séquence *carphone* et de la séquence *foreman* pour la comparaison des ZQPGM avec le SRG et la LPEGM (seule la trame médiane est marquée).

Concernant la comparaison en elle-même, nous avons utilisé plusieurs jeux de paramètres α,ω et $\mathcal{A}_m$. Nous avons fixé $\alpha = \omega$ puis les avons fait varier de 10 à 100 avec un pas de 10, de même pour $\mathcal{A}_m$ que nous avons également fait varier de 10 à 100 avec un pas de 10. Ces deux variations donnent 100 combinaisons de paramètres pour l'approche 2D+t et pour l'approche t+2D soit 200 combinaisons en tout pour chaque jeu de marqueurs. Cette grande variété de combinaisons nous permettra d'évaluer la robustesse de notre méthode aux paramètres. La table 2.3 présente les indices de Jaccard moyen obtenus pour certaines combinaisons de paramètres et ceux obtenus par le SRG et la LPEGM. L'intégralité des indices est présentée dans la figure 2.5.

Méthode	α, ω	$\mathcal{A}_m$	Indice de Jaccard Moyen			
			carphone		*foreman*	
			Jeu 1	Jeu 2	Jeu 1	Jeu 2
ZQPGM 2D+t	30	10	0,782	0,905	0,710	**0,952**
	50	50	**0,825**	0,910	0,674	0,884
	90	50	0,793	0,908	**0,791**	0,859
ZQPGM t+2D	20	60	0,767	**0,928**	0,695	0,944
	40	100	0,749	0,925	0,656	0,940
	100	70	0,781	0,919	0,637	0,935
SRG	-	-	0,641	0,548	0,529	0,400
LPEGM	-	-	0,749	0,897	0,634	0,946

TABLE 2.3 – Échantillon des résultats des ZQPGM, du SRG et de la LPEGM obtenus sur l'extrait de 80 trames de *carphone* et sur l'extrait de 40 trames de *foreman* avec deux jeux de marqueurs distincts pour chaque séquence.

Les indices de Jaccard moyens obtenus montrent que les ZQPGM sont plus performantes que le SRG pour les paramètres choisis (à l'exception de certaines combinaisons de paramètres pour le jeu 1 et la séquence *carphone*, figure 2.5.a). Concernant la comparaison avec la LPEGM, le bilan est plus mitigé. Nous avons utilisé 200 combinaisons de paramètres pour chaque jeu de marqueurs. Pour le jeu 1 de la séquence *carphone*, 74 des 200 combinaisons de paramètres permettent d'obtenir

de meilleurs résultats que la LPEGM. Pour le jeu 2, il y a 110 combinaisons de paramètres donnant de meilleurs résultats que la LPEGM. Conernant la séquence *foreman*, pour le jeu 1, 184 des 200 combinaisons de paramètres donnent de meilleurs résultats que la LPEGM mais, pour le jeu 2, seulement 37 des 200 combinaisons de paramètres permettent d'obtenir de meilleurs résultats que la LPEGM. Les ZQPGM peuvent donc être plus performantes que la LPEGM mais la qualité des résultats qu'elles produisent est liée aux valeurs des paramètres α, ω et $\mathcal{A}_m$ utilisés. Notre approche est donc peu robuste aux paramètres. Pour la séquence *carphone* et le jeu de marqueurs 1, l'indice de Jaccard moyen varie entre 0,580 et 0,897, tandis que pour le jeu 2, les valeurs oscillent entre 0,860 et 0,938. Concernant la séquence *foreman*, les valeurs oscillent entre 0,570 et 0,894 pour le jeu 1 et entre 0,827 et 0,960 pour le jeu 2. Ce manque de robustesse peut donc être compensé par les marqueurs. En effet, les jeux 2 qui comportent plus de pixels donnent de meilleurs résultats que les jeux 1, de plus les résultats obtenus pour les jeux 2 sont plus robustes aux paramètres que ceux obtenus par les jeux 1. Fournir des marqueurs fiables permet donc de contrer la relative non robustesse aux paramètres de notre approche.

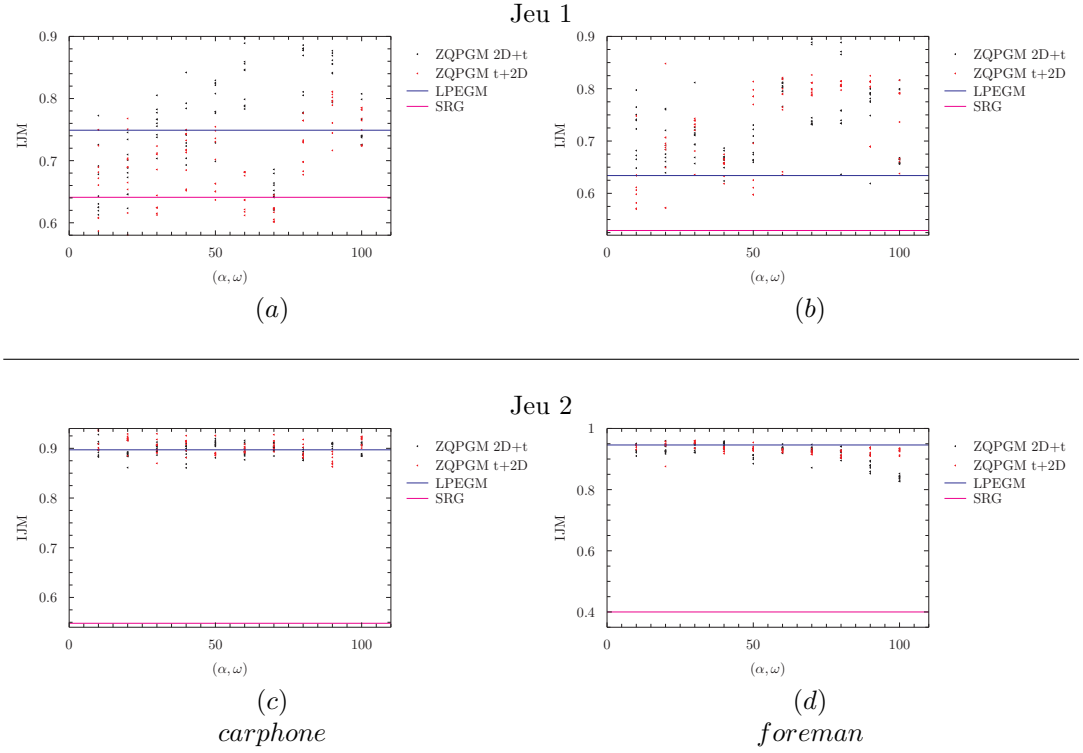


FIGURE 2.5 – Comparaison des indices de Jaccard moyen obtenus sur les séquences *carphone* et *foreman* selon les différentes méthodes et les différents jeux de marqueurs.

Outre l'indice de Jaccard moyen de l'ensemble de la séquence vidéo, il est également intéressant d'étudier l'évolution de l'indice de Jaccard moyen au cours du temps, c'est-à-dire la valeur de l'indice pour chaque trame. Cette étude nous permet d'évaluer la stabilité temporelle de notre approche et de la comparer avec celle des autres méthodes utilisées dans cette section. Les variations de l'indice de Jaccard au cours du temps sont données dans la figure 2.6. Pour les paramètres des deux approches ZQPGM, nous avons utilisé ceux de la table 2.3 donnant les meilleurs résultats. La pire configuration serait une valeur élevée d'indice de Jaccard pour la trame médiane (qui est la seule trame présentant des marqueurs) et des valeurs en baisse au fur et à mesure que l'on s'éloigne temporellement de la trame médiane. Dans cette configuration, la courbe des indices ressemblerait à une pyramide ou à un fort pic sur la trame médiane, comme pour le SRG avec le jeu 2 (2.6.h). Les ZQPGM ne sont pas dans cette configuration, et la valeur maximale de l'indice de Jaccard n'est d'ailleurs jamais atteinte sur la trame médiane. On remarque que c'est la seconde partie de la segmentation qui obtient les meilleures valeurs d'indice (à l'exception de 2.6.a) et ce même pour la

LPEGM et le SRG : ceci indique que la segmentation de la seconde partie de la séquence vidéo est sans doute plus facile. Les ZQPGM, bien qu'ayant des valeurs d'indice de Jaccard variant au cours du temps, présentent une relative stabilité temporelle. En effet, les valeurs d'indice ne s'effondrent pas en s'éloignant de la trame contenant des informations. De plus, elles peuvent même s'améliorer en s'éloignant de la trame marquée montrant que la propagation de l'information est effective.

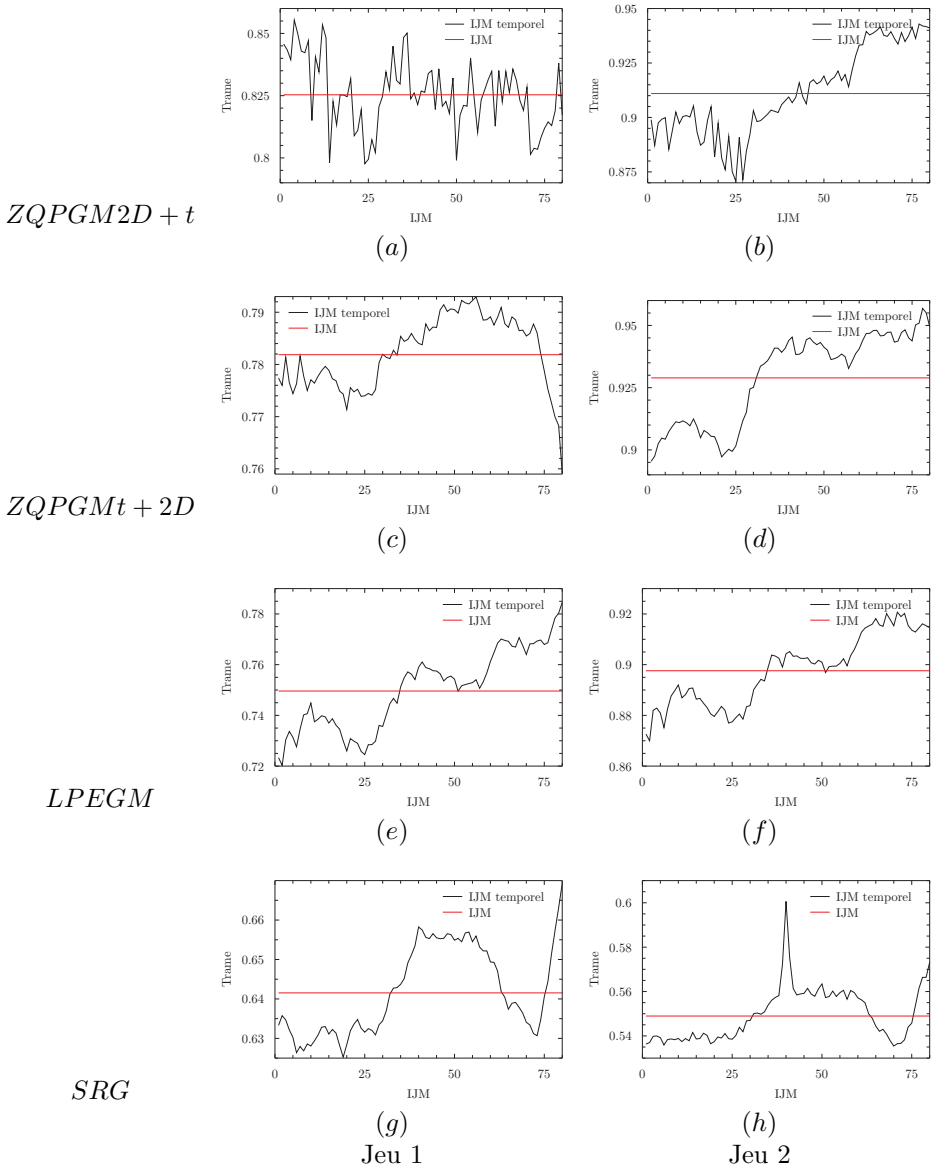


FIGURE 2.6 – Évolution temporelle de l'indice de Jaccard moyen sur la segmentation de l'extrait de *carphone* selon les différentes méthodes et les différents jeux de marqueurs.

La figure 2.7 montre la segmentation finale obtenue en utilisant les ZQPGM avec $\alpha = \omega = 50$ et $\mathcal{A}_m = 50$ en utilisant l'approche 2D+t. On observe que, bien que les marqueurs soient bien placés, la segmentation résultante présente encore des défauts. Cependant, le processus actuel de notre méthode (cf. fig 2.2) ne permet pas la modification des marqueurs pour affiner la segmentation. Il est donc nécessaire que les marqueurs soient optimaux dès le départ. Dès lors, cette approche est difficilement utilisable en l'état et ce, surtout pour segmenter des séquences vidéo de longue durée dont la qualité de la segmentation va dépendre de l'aptitude de la méthode à propager l'information des marqueurs. Néanmoins, notre approche s'affranchit de plusieurs des inconvénients recensés dans les approches existantes : elle permet la segmentation de plusieurs objets d'intérêt, elle ne

travaille pas sur les pixels mais sur une pré-segmentation ce qui lui garantit un coût calculatoire faible, et elle permet la correction de certaines régions de la pré-segmentation si les marqueurs de l'utilisateur les remettent en cause.

FIGURE 2.7 – Résultat de la ZQPGM sur les extraits de *carphone* et *foreman* avec $\alpha = \omega = 50$ et $\mathcal{A}_m = 50$ avec l'approche 2D+t et les jeux de marqueurs 2.

Dans cette section, nous avons présenté une méthode de segmentation vidéo guidée qui s'appuie sur les ZQP. Bien qu'elle dispose d'une partie en-ligne efficace calculatoirement et donne de bons résultats, elle ne permet pas encore à l'utilisateur d'évaluer la segmentation finale et de corriger ses marqueurs pour l'améliorer. Nous étudions cette possibilité dans la section suivante.

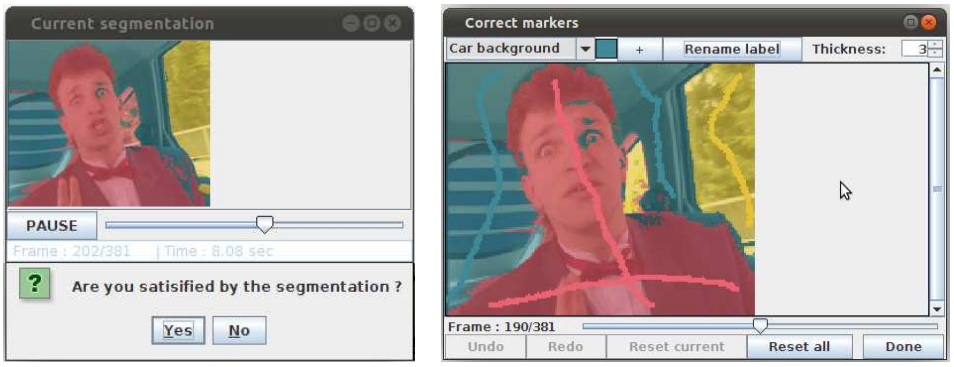
2.5 Évaluation de zones quasi-plates et correction guidée par marqueurs

La méthode de segmentation guidée proposée dans la section précédente n'est pas interactive. Dans cette section, nous rendons cette méthode interactive et évaluons ses performances.

Rendre interactive la méthode proposée dans la section précédente consiste principalement à rajouter deux étapes au processus de segmentation guidée par marqueurs :

1. Évaluation de la segmentation par l'utilisateur : l'utilisateur visualise la segmentation basée sur les marqueurs actuels, et décide de la valider ou d'initier un processus de correction (interface présentée dans la figure 2.8.a) ;
2. Correction des marqueurs : la segmentation actuelle ne convient pas à l'utilisateur, il peut éditer les marqueurs actuels afin de mieux guider et personnaliser la segmentation (interface présentée dans la figure 2.8.b).

Le processus interactif de segmentation guidée par marqueurs est présenté dans la figure 2.9. Il intègre les deux nouvelles étapes citées précédemment ainsi que la correction des ZQP remises en cause qui a été présentée dans la section précédente. Ce processus interactif est répété jusqu'à ce que l'utilisateur soit satisfait de la segmentation finale en objets-vidéo. L'utilisation de cette méthode est rendue possible principalement par la rapidité avec laquelle une segmentation est proposée une fois que l'utilisateur a validé les marqueurs. En effet, le temps de calcul faible permet



(*a*) (*b*)

FIGURE 2.8 – Interface : (a) Évaluation de la segmentation, (b) Correction des marqueurs.

une réelle interactivité et de nombreuses modifications de marqueurs dans des temps restreints pour l'utilisateur. De plus, la correction des ZQP remises en cause par les marqueurs améliore la sur-segmentation initiale, ce qui peut être particulièrement intéressant dans un contexte multi-utilisateurs. En effet, dans un tel contexte, la sur-segmentation initiale en ZQP est commune à tous les utilisateurs. La personnalisation s'effectuant au travers des marqueurs, un utilisateur dont les marqueurs remettent en cause certaines ZQP va améliorer la sur-segmentation initiale et contribue directement à la qualité des segmentations des autres utilisateurs. Nous allons détailler ces deux points dans la suite.

Le principal avantage de notre approche interactive est le faible coût computationnel de la fusion de régions effectuée par un SRG sur notre graphe d'adjacence de régions. L'efficacité calculatoire de cette étape repose sur trois facteurs : elle est effectuée sur une importante réduction de l'espace de données, chaque région possède un descripteur simple qui permet une fusion peu coûteuse en termes d'opérations de calcul, et la correction des ZQP remises en cause est effectuée sur une petite partie de la séquence vidéo. La réduction de données permet de travailler sur un nombre d'éléments nettement inférieur au nombre de pixels de la vidéo. L'utilisation de la couleur moyenne de la région comme descripteur est relativement restrictive. Cependant, cela permet au calcul du nouveau descripteur, lors de la fusion de deux régions, de correspondre à quatre additions (on additionne, pour chaque bande, la somme des valeurs de pixels des deux régions et on additionne le nombre de pixels des deux régions) et trois divisions (on divise chaque somme obtenue par le nombre total de pixels dans la nouvelle région). Chaque fusion ne coûte donc que 6 opérations. L'utilisation de descripteurs plus discriminants mais plus complexes aurait un impact important sur le coût computationnel de notre approche. La correction des ZQP remises en cause par les marqueurs est nécessaire, mais nécessite également un coût calculatoire qui s'avère ici restreint. En effet, l'opération de resegmentation ne s'applique qu'à quelques ZQP à chaque itération. Nous n'effectuons donc ces calculs uniquement sur une petite partie de la séquence vidéo. La combinaison de ces différents facteurs induit un temps de calcul presque négligeable par rapport au temps utilisateur (temps passé par l'utilisateur à éditer les marqueurs). La figure 2.10 présente une comparaison du temps de calcul et du temps utilisateur nécessaire pour l'obtention d'une même valeur d'indice de Jaccard. On observe clairement que, comme nous l'indiquions précédemment, le temps de calcul est négligeable par rapport au temps passé par l'utilisateur sur l'édition des marqueurs. Ainsi, dans notre approche l'utilisateur est toujours actif et n'attend pas les résultats des calculs qui sont pour lui quasi-instantanés.

Le second avantage de notre approche est l'amélioration de la sur-segmentation initiale au fur et à mesure des interactions de l'utilisateur. En effet, dans le cas où les marqueurs de l'utilisateur remettent en cause une ZQP, celle-ci est resegmentée. Cette resegmentation impacte directement la sur-segmentation initiale et améliore sa qualité en terme de précision maximale mais augmente légèrement son ratio de sur-segmentation. Une ZQP remise en cause est, en général, segmentée en deux régions (car remise en cause par deux marqueurs), ce qui n'augmente que d'un le nombre

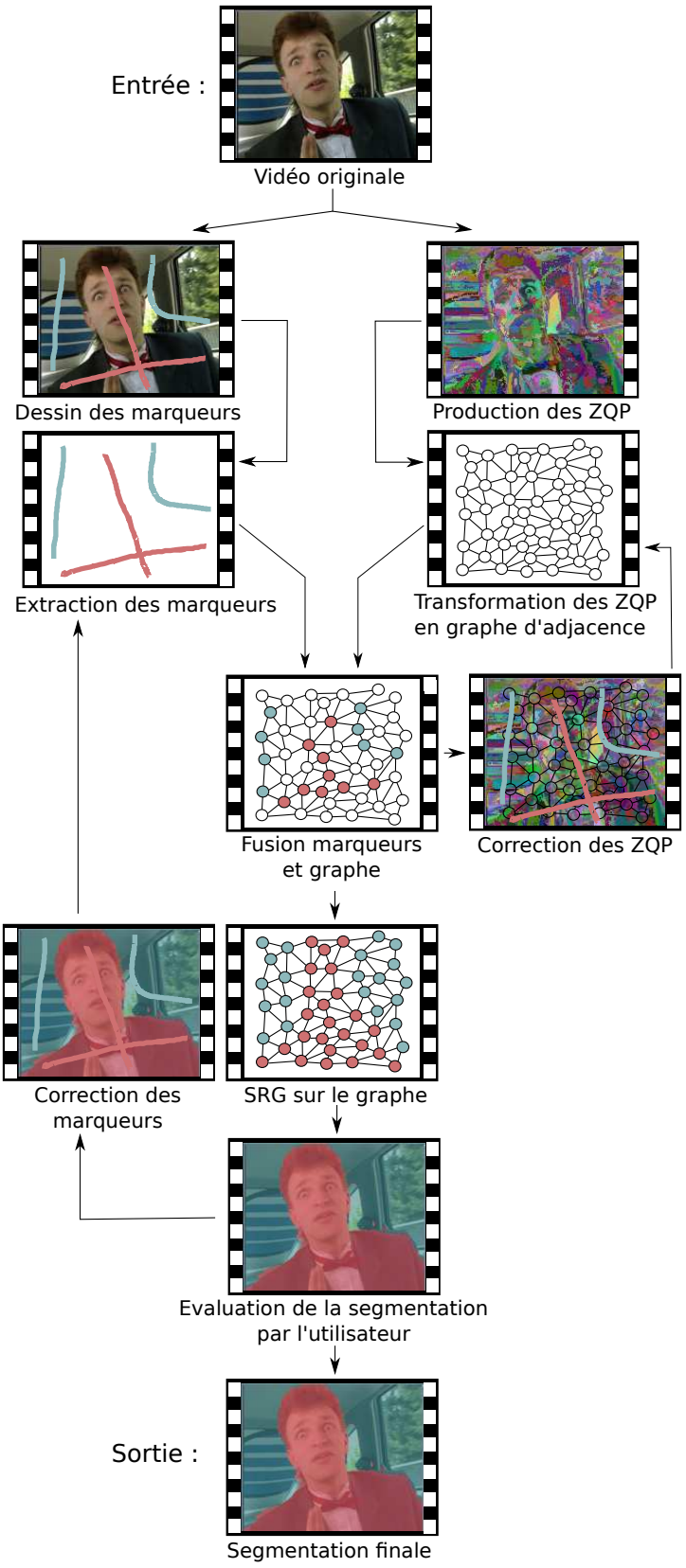


FIGURE 2.9 – Processus de segmentation vidéo interactive par les ZQP guidée par marqueurs.

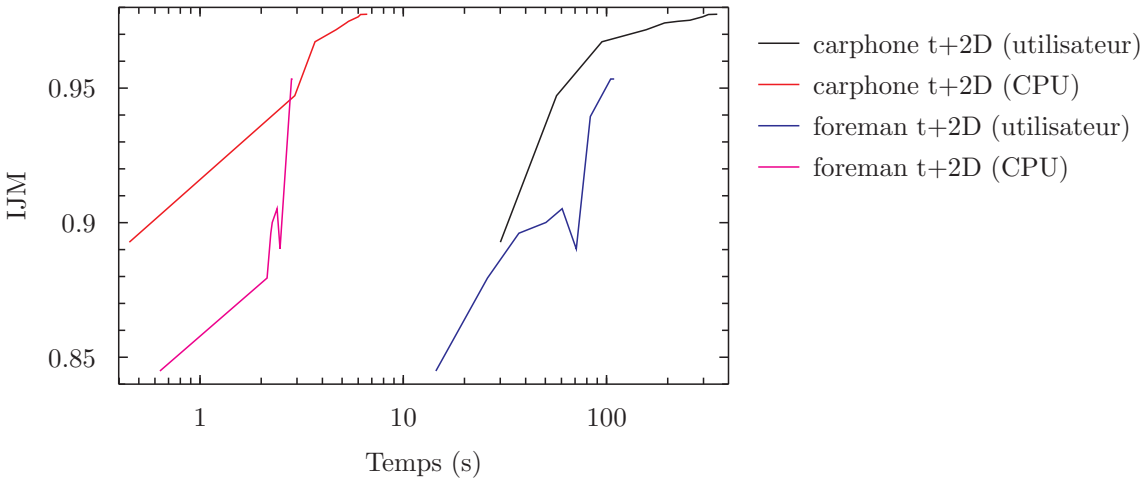


FIGURE 2.10 – Comparaison des temps CPU et utilisateur de notre approche interactive.

de régions de la sur-segmentation et augmente de façon limitée le ratio de sur-segmentation. La figure 2.11 présente l'évolution de la précision maximale de la sur-segmentation en fonction des interactions successives effectuées par l'utilisateur. Nous considérons ici comme interaction, une itération du processus. On observe effectivement que la précision maximale augmente au fur et à mesure que l'utilisateur interagit avec notre méthode. L'utilisateur va donc, au cours des différentes itérations, corriger les imperfections de la sur-segmentation initiale. Cette correction rend naturellement plus juste les segmentations de l'utilisateur et, comme nous l'évoquions précédement, améliore le résultat de segmentation d'autres utilisateurs, dans un contexte multi-utilisateurs.

Nous avons, dans cette section, présenté une évolution de notre méthode de segmentation de zones quasi-plates guidée par marqueurs. Cette évolution permet à l'utilisateur d'évaluer la segmentation obtenue et de l'améliorer interactivement lors d'un processus à faible coût calculatoire. De plus, la méthode de correction des ZQP remises en cause par les marqueurs permet d'améliorer la sur-segmentation initiale et donc intrinséquement celle de la segmentation finale.

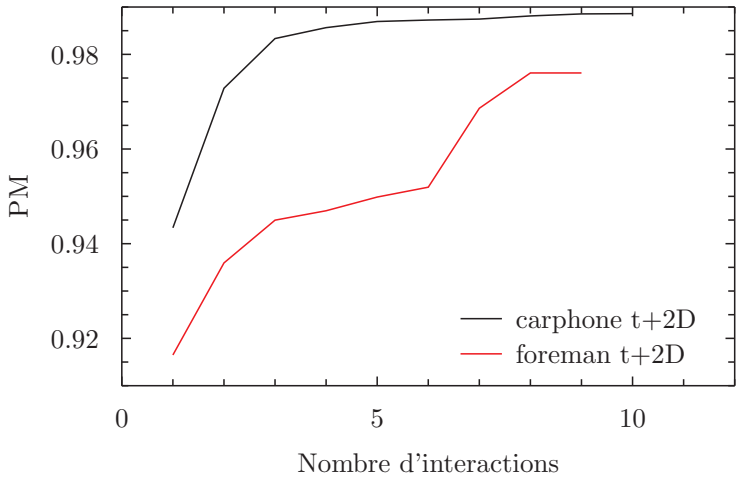


FIGURE 2.11 – Évolution de la précision maximale de la sur-segmentation au fur et à mesure des interactions.

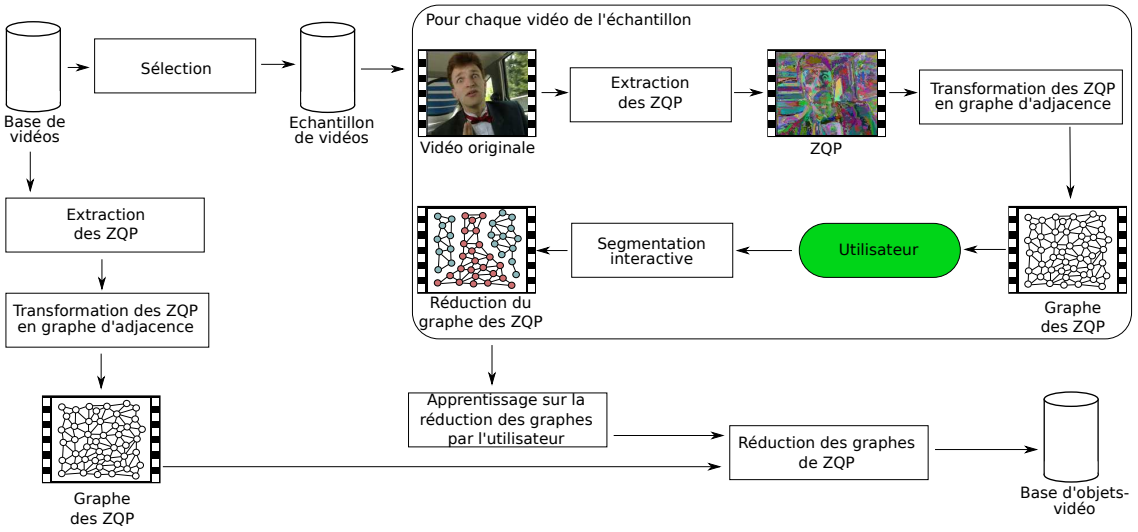


FIGURE 2.12 – Schéma de co-segmentation basée sur les ZQP.

2.6 Vers la généralisation de la segmentation

Si notre méthode est efficace d'un point de vue calculatoire, le fait qu'elle implique l'utilisateur dans la segmentation de chaque séquence vidéo ne la rend pas applicable à une grande base de séquences vidéo. Cependant nous pouvons, sans intervention de l'utilisateur, disposer pour chaque séquence vidéo d'une sur-segmentation en ZQP que nous transformons en graphe pour l'étape interactive de notre processus. Notre méthode peut être assimilée à une réduction de ce graphe d'adjacence des ZQP. Il serait donc possible d'apprendre des réductions de graphe effectuées sur les séquences vidéo segmentées par l'utilisateur, afin de guider la réduction de graphes issus d'autres séquences vidéo. Cet apprentissage peut prendre plusieurs formes. Intuitivement, nous pouvons imaginer deux possibilités pour l'apprentissage : un apprentissage relatif aux nœuds ou aux arêtes. Le premier consisterait à déterminer, via les marqueurs des séquences vidéo segmentées, les caractéristiques qui font qu'un nœud est un marqueur ou non. Le second consisterait à apprendre via les segmentations finales de séquences vidéo ce qui caractérise une arête comme une arête de frontières ou une arête de régions. La problématique d'apprentissage de segmentation aussi appelée co-segmentation a été abordée dans le cadre de la segmentation d'images. Nous pouvons citer iCoseg [BKP+11] parmi les travaux les plus récents. Cette méthode permet de co-segmenter différentes vues d'un même objet, et la méthode de segmentation est une extension de [BJ01] que nous avons rappelée précédemment. Les auteurs appliquent leur coupe de graphe sur une sur-segmentation initiale (ici par Mean-Shift). De plus, la méthode dispose d'un système de proposition d'endroits à marquer basée sur les marqueurs initiaux et le contenu des images. Ce système détermine les endroits des images où la segmentation est la plus incertaine et demande à l'utilisateur d'y dessiner des marqueurs afin de guider efficacement la co-segmentation. Cette méthode pourrait être adaptée aux séquences vidéo surtout s'il s'agit de séquences montrant le ou les mêmes objets. Cependant, cette méthode n'est sans doute pas applicable au traitement conjoint de plusieurs séquences. Dans ce cas, si l'on ne segmente qu'une seule séquence vidéo et que l'on souhaite utiliser les informations issues de cette vidéo pour en segmenter d'autres, les travaux des mêmes auteurs sur la sélection de la meilleure image graine pour la co-segmentation [BPK+09] sont également intéressants et extensibles aux séquences vidéo. D'autres travaux récents [JBP10, VKR10, CCC11] montrent l'importance du problème de la co-segmentation, pour l'instant limitée aux images fixes. Ce problème sera certainement abordé prochainement dans le contexte des séquences vidéo. Nous pensons que notre approche de segmentation interactive pourrait s'intégrer dans une méthode de co-segmentation de séquences vidéo (cf. figure 2.12). Cependant, il s'agit d'un domaine de recherche récent, les méthodes actuelles co-segmentent principalement le même objet ou des objets très similaires mais depuis des points de vue ou sur des fonds différents. Même pour les images fixes, nous sommes encore loin de la co-segmentation sur des ensembles hétérogènes d'images.

2.7 Conclusion

L'approche de segmentation personnalisée de séquences vidéo que nous avons proposée est une solution fiable aux problèmes posés par la segmentation de séquences vidéo. Elle ne présente pas les inconvénients observés dans les systèmes existants. En effet, elle permet la segmentation de plusieurs objets d'intérêt simultanément, elle a un coût calculatoire faible, puisqu'elle s'appuie sur une réduction des données, et elle permet à l'utilisateur de corriger la segmentation sans relancer tout le processus.

On pourrait envisager d'utiliser un descripteur plus riche que la couleur moyenne pour décrire les régions de la sur-segmentation initiale. En effet, l'utilisation d'attributs plus discriminants permettrait d'améliorer les résultats obtenus par le processus de fusion de régions guidée par marqueurs. Cependant, le point fort de notre méthode est son faible coût calculatoire qui permet de réelles interactions avec l'utilisateur. Utiliser des descripteurs plus complexes impliquerait une augmentation du coût calculatoire, qui réduirait de fait l'interactivité de notre méthode. Le point faible de notre méthode est la nécessité de définir des marqueurs pour chaque séquence vidéo d'une base, et nous avons évoqué la co-segmentation qui permettrait de résoudre ce problème. Mais, il s'agit d'un domaine de recherche récent et, à la connaissance de l'auteur, il n'existe pas encore de solution pour co-segmenter des séquences vidéo hétérogènes.

Dans ce chapitre, nous avons résolu le problème de l'extraction des objets-vidéo. Nous allons maintenant étudier comment ces objets-vidéo peuvent être manipulés, par exemple au travers d'un processus de fouille.

Chapitre 3

Fouille vidéo

Sommaire

3.1 Introduction

La fouille de données [CPSK07] est un processus d'extraction d'informations et de connaissances d'une masse de données. La fouille vidéo [RDD02] est l'application de ce processus à des données vidéo. Selon ces définitions, un Système de Fouille Vidéo (SFV) est un système capable d'extraire de l'information à partir d'une masse de séquences vidéo, c'est-à-dire des séquences temporelles d'images, éventuellement couplées à des données audio. Dans ce mémoire, nous considérons uniquement les données visuelles (pas de données sonores).

Dans ce chapitre, nous présentons un court état de l'art sur la fouille vidéo et proposons une taxonomie permettant de caractériser les méthodes qui ont été proposées dans la littérature. Nous étudions ensuite comment la notion d'objet est prise en compte dans la fouille vidéo et l'impact de l'utilisation des objets sur la taxonomie que nous proposons. Puis, nous proposons un cadre générique adapté à la fouille vidéo orientée-objet. Nous appliquons ensuite le cadre proposé au contexte du regroupement d'objets-vidéo issus de la méthode de segmentation interactive proposée dans le chapitre 2. Enfin, nous discutons des limites actuelles de notre système et des perspectives qu'il offre.

3.2 Etat de l'art

Dans cette section nous présentons quelques études existantes sur les méthodes et systèmes de fouille vidéo afin d'avoir un aperçu des recherches actuelles. Puis, nous proposons une taxonomie permettant de caractériser les SFV qui nous sera utile dans la suite de ce chapitre.

3.2.1 Études existantes

De nombreux auteurs ont proposé des états de l'art relatifs à la fouille vidéo. Cependant, aucun de ces travaux ne présente l'ensemble du domaine de la fouille vidéo, chaque contribution concernant en général un sous-domaine : indexation, classification, résumé, recherche, etc. Nous pouvons noter que ces sous-domaines ne sont pas spécifiques à la vidéo. Parmi les premiers travaux, Idris et Panchanathan [IP97] présentent quelques méthodes d'indexation vidéo et précisent que le niveau naturel pour analyser le contenu visuel devrait être l'objet. Brunelli *et al.* [BMM99] présentent également des systèmes d'indexation vidéo et notent, en 1999, que la détection des objets génériques ne peut pas être réalisée avec les méthodes actuelles. Dai *et al.* [DZL06] soulignent la difficulté d'extraire des descripteurs adaptés à tous les types de séquences vidéo et donc la nécessité de faire des hypothèses sur la nature des séquences vidéo traitées. Ceci rend impossible, pour le moment, l'existence de systèmes génériques. Ils notent que la fouille vidéo est encore à un stade préliminaire mais connait un développement important. Plus récemment, Brezeale et Cook [BC08] étudient le niveau auquel les informations font l'objet d'une classification dans le domaine vidéo : la plupart des méthodes étudiées proposent de travailler au niveau global, quelques-unes utilisent le plan ou la scène et, surtout, aucune n'utilise l'objet. Money et Agius [MA08] étudient les systèmes de résumé de vidéo. Ils proposent d'utiliser des informations propres à l'utilisateur pour produire des résumés personnalisés et plus riches sémantiquement. Ren *et al.* [RSSZ09] se concentrent sur l'utilisation d'informations spatio-temporelles pour la recherche de vidéo. Ils pointent l'importance des relations spatio-temporelles inter-objets dans le cadre de la recherche de vidéo. Snoek et Worring [SW09] présentent le domaine de la recherche de vidéo basée sur des concepts via une étude de près de 300 articles. Les auteurs insistent sur l'importance de l'efficacité calculatoire des méthodes et la nécessité de disposer d'un très large panel de détecteurs de concepts permettant d'aborder la diversité des contenus vidéo.

Contrairement à ces études, nous ne nous concentrons pas sur un objectif spécifique de fouille vidéo, mais souhaitons prendre en considération tous les objectifs possibles (exposés dans la section 3.2.2.1). Comme [IP97], nous pensons que le niveau de l'objet est le plus adapté pour la fouille vidéo. C'est pour permettre l'utilisation de l'objet-vidéo, que nous proposons dans cette thèse une méthode de segmentation vidéo interactive. Dans la suite du chapitre, nous nous concentrons sur le rôle de l'objet dans le processus de fouille de données vidéo et discutons de la position de l'utilisateur comme élément fondamental du processus visuel d'exploitation.

3.2.2 Une taxonomie pour les Systèmes de Fouille Vidéo

De nombreux aspects sont à prendre en compte pour décrire et caractériser un SFV. Dans cette section, nous allons identifier ces différents aspects et définir une taxonomie. Nous utilisons cette taxonomie dans la suite du chapitre pour analyser les SFV existants.

3.2.2.1 Objectifs des SFV

Les bases de séquences vidéo nécessitent de grandes capacités de stockage et la fouille manuelle, c'est-à-dire sans aucune automatisation, de ces bases est fastidieuse. Des SFV ont donc été développés pour accomplir de façon automatique les tâches jusqu'alors accomplies par des êtres humains. Ainsi, les objectifs possibles pour un SFV sont variés et dépendent des besoins de ses utilisateurs. Le *résumé de séquences vidéo* (Res) vise à produire de courts et représentatifs extraits de vidéo dans le but de permettre aux utilisateurs de comprendre leur sujet et leur contenu sans les visionner dans leur intégralité. L'*indexation de séquences vidéo* (Ind) est la caractérisation d'une vidéo afin d'être ultérieurement capable de la retrouver rapidement en utilisant des requêtes spécifiques

(par exemple, « Je désire une vidéo où Zidane marque un but contre le Brésil »). La *classification de séquences vidéo* (Cla) vise à regrouper les séquences vidéo dans des catégories prédéfinies afin d'identifier leur contenu. Elle peut également, dans le cas où il n'existe pas de catégorie pré-définies, regrouper les séquences similaires afin de les catégoriser ultérieurement. La *recherche basée sur le contenu* (Rec) permet aux utilisateurs de retrouver des séquences vidéo similaires à une séquence donnée en requête, on l'oppose à la recherche par mots-clés qui repose sur une annotation préalable des séquences vidéo. L'*annotation de séquences vidéo* (Ann) consiste à associer à une séquence des mots décrivant des éléments de son contenu ou en rapport avec son contenu (par exemple, « voiture » ou « salon de l'automobile » ou « Genève »). La *détection de copies* (Cop) a pour objectif de détecter si une séquence vidéo est une copie d'une autre, et ce même si elle a été modifiée. La *navigation* repose sur une structuration préalable des données et vise à permettre à un utilisateur d'explorer de façon efficace une base de séquences vidéo. Elle est assez proche de l'*indexation*.

3.2.2.2 Propriétés des SFV

Les SFV sont ainsi caractérisées par différentes propriétés relatives à la nature des données (compressées ou non, génériques ou non) et des informations à prendre en compte (séquence, scène, objet, etc.), aux descripteurs à extraire et à l'échelle à laquelle les calculer, et au rôle joué par l'utilisateur.

Dans cette section, nous introduisons et décrivons ces différentes propriétés.

Type des données

Un SFV peut avoir à traiter différents types de données vidéo. Ainsi une séquence vidéo peut être compressée (C) par différents algorithmes (par exemple, des standards anciens tels que MJPEG, MPEG-1/2 ou plus récents comme MPEG-4, H-263, etc.) ou disponible sous forme brute (B) (c'est-à-dire contenant un flux brut d'images successives). La compression permet un stockage moins coûteux en mémoire mais requiert un processus de décompression avant une visualisation et peut induire une perte d'information qui peut s'avérer critique dans certains domaines (par exemple, l'imagerie médicale). Traiter des séquences compressées est plus rapide car le volume de données à traiter est moindre, cependant l'extraction de concepts visuels est plus complexe, alors que l'analyse du contenu visuel peut être effectuée directement dans le cadre de séquences vidéo brutes (induisant néanmoins un coût calculatoire plus élevé). Les SFV peuvent également être dédiés au traitement de types de séquences spécifiques (S) (par exemple des retransmissions sportives ou des journaux d'informations) ou génériques (G). Traiter des séquences vidéo spécifiques permet d'obtenir de meilleurs résultats car l'on peut utiliser les connaissances du domaine dans le processus de fouille. Par contre, la prise en compte de séquences génériques par un SFV facilite la réutilisation et l'adaptation de ce dernier à des contextes variés.

Éléments

Quel que soit l'objectif d'un SFV, il peut être atteint en considérant différents éléments, de la séquence vidéo dans son intégralité au simple pixel. La première étape en fouille vidéo consiste en l'extraction de l'élément à traiter. La *séquence vidéo intégrale* (Vid) est l'élément classique et le plus simple à traiter car il ne nécessite aucune extraction. Néanmoins, si la séquence contient des scènes très différentes, cet élément peut ne pas être très significatif. Une *scène* (Sce) est composée de plusieurs plans dans un contexte spatio-temporel identique et peut être difficile à extraire. Un *plan* (Pla) est un segment de vidéo délimité par deux transitions, le problème de son extraction est un sujet très étudié et de nombreuses méthodes ont été proposées pour le résoudre [LHV03][SOD10]. La *trame* (Tra) est l'unité temporelle d'une séquence vidéo, une vidéo étant une séquence temporelle de trames. Un *objet* (Obj) est, selon notre définition, l'élément qui comporte le plus de sémantique mais son utilisation est limitée par la difficulté d'extraire la représentation d'un objet réel (par exemple une voiture) à un instant donné, ou au cours du temps. Une *région* (Reg) est un ensemble de pixels connexes qui (au contraire de l'objet) ne repose pas sur un concept sémantique. Enfin, le *pixel (Pix)* est le plus petit élément et, pris isolément, il n'apporte que peu voire pas d'information [1].

1. A l'exception de certains domaines d'application où sa valeur correspond à une valeur physique porteuse d'informations (par exemple en imagerie thermique).

La figure 3.1 illustre les différents éléments.

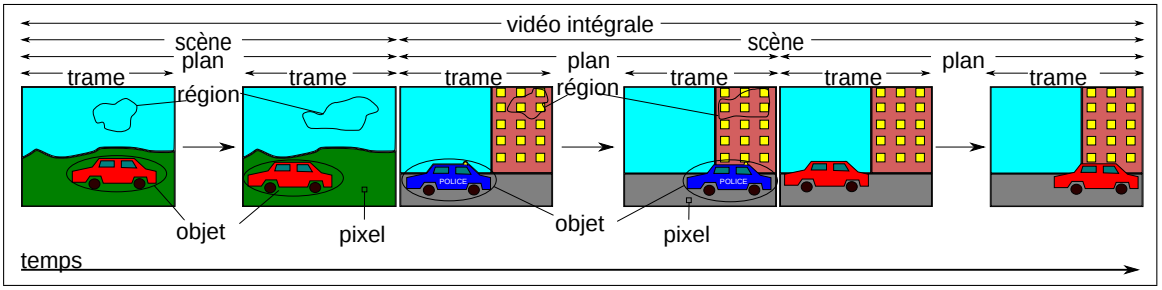


FIGURE 3.1 – Les différents éléments qu'un SFV peut traiter.

Échelles

Afin d'analyser et caractériser les éléments par différents descripteurs, il est nécessaire de choisir l'échelle à laquelle les descripteurs vont être calculés. L'échelle est liée en partie à l'élément utilisé et au descripteur choisi. À une échelle *globale* (Glo), les descripteurs vidéo sont appliqués à l'intégralité de la séquence vidéo. À l'échelle du *bloc* (Blo), la séquence est divisée en blocs suivant une grille spatiale, les descripteurs sont alors calculés dans chaque bloc indépendamment. L'échelle *région* partitionne la séquence vidéo en régions de tailles et formes variées, à l'aide d'une étape de segmentation. Les descripteurs sont ensuite associés à chaque région indépendamment. L'échelle *objet* (Obj) consiste à définir les descripteurs pour les objets réels représentés dans les éléments de la vidéo. L'échelle *point d'intérêt* (PI) consiste à calculer les descripteurs sur des points (et leur voisinage) présent dans un contexte remarquable. Enfin l'échelle *pixel* (Pix) est la plus petite possible : les descripteurs ne servent alors qu'à décrire un pixel. La figure 3.2 illustre les différentes échelles.

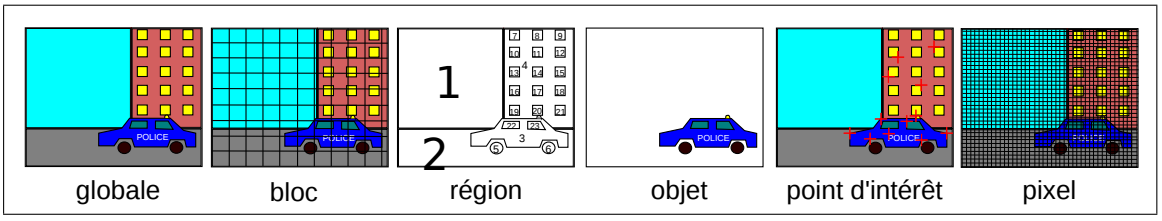


FIGURE 3.2 – Les différentes échelles de descripteurs possibles pour un SFV.

Descripteurs

Il existe de nombreux descripteurs pour décrire le contenu d'une séquence vidéo. Il est possible de décrire la *couleur* (Coul) [MOVY01] et/ou les *textures* (Tex) [MOVY01][CP05] présentes dans les données visuelles. Nous pouvons aussi abstraire les formes et les contours (For) [Bob01][ZL03] afin de décrire la morphologie des éléments présents dans la vidéo. Outre ces descripteurs classiques, il existe de nombreux autres descripteurs spécifiques (par exemple pour le *mouvement* (Mou) [JD01]) régulièrement proposés dans la littérature [RHC99]. Choisir le descripteur le plus adapté à un SFV précis n'est pas trivial car chaque descripteur vise à caractériser un contenu vidéo selon un point de vue particulier. De plus, il faut prendre en compte le coût calculatoire car l'extraction de certains descripteurs peut s'avérer très gourmande en temps de calcul.

3.2.2.3 Implication de l'utilisateur

L'implication de l'utilisateur est un point critique dans un SFV. Nous considérons quatre niveaux d'implication possibles de l'utilisateur. Celle-ci peut être *Nulle* (Nul) si le système est to-

talement automatique et que l'utilisateur n'intervient pas dans le processus [2]. Elle est *Supervisée* (Sup) lorsque l'utilisateur doit fournir un ensemble complet de données étiquetées ou une ontologie afin de configurer le système pour traiter un jeu de données spécifiques. L'implication est dite *Semi-supervisée* (S-sup) quand l'utilisateur doit fournir moins de données étiquetées que pour l'implication *supervisée* et/ou doit (in)valider certains résultats afin de guider le processus de fouille [3]. Enfin, l'implication est appelée *Paramétrique* (Param) lorsque l'utilisateur doit fixer les différents paramètres du système. La paramétrisation peut-être simple et intuitive ou fastidieuse selon le nombre de paramètres ainsi que leur complexité. Les différents niveaux d'implication de l'utilisateur sont résumés dans la figure 3.3.

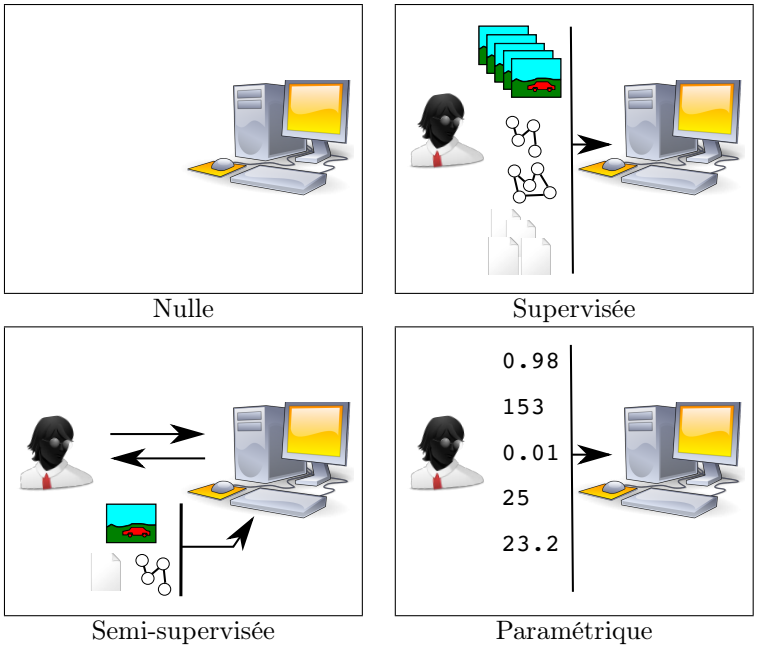


FIGURE 3.3 – Les différentes implications possibles de l'utilisateur dans un SFV.

La taxonomie que nous avons introduite dans cette section permet de décrire et caractériser un SFV. Nous allons l'utiliser à la fin de la section 3.3.1 pour caractériser les principaux SFV de la littérature.

3.3 L'objet dans la fouille vidéo

3.3.1 Les sytèmes de fouille vidéo actuels

Dans cette section, nous caractérisons les travaux récents concernant de près ou de loin la fouille vidéo orientée-objet, afin de mettre en lumière les tendances actuelles dans ce domaine et de positionner nos contributions.

Anjulan et Canagarajah [AC07b] présentent un système de recherche basé sur le contenu d'objets-vidéo. En premier lieu, ils extraient des régions à invariance locale [MTS+05] et les caractérisent avec des descripteurs SIFT [Low99]. Puis, les régions à invariance locale représentant le même objet dans des trames connexes sont liées en fonction des descripteurs, formant une suite temporelle de régions à invariance locale. Ces suites sont ensuite regroupées en classe par similarité.

2. En réalité, l'implication de l'utilisateur n'est jamais nulle. Il est toujours impliqué ne serait-ce que dans de le choix de l'algorithme, de la méthode ou du système qu'il utilise.
3. Cette approche s'apparente à l'apprentissage semi-supervisé [Zhu08] ainsi qu'à l'apprentissage actif.

Les requêtes, c'est-à-dire des régions sélectionnées par l'utilisateur, sont comparées aux centres des classes pour retrouver les objets similaires à ceux proposés par l'utilisateur.

Anjulan et Canagarajah [AC07a] ont étendu la méthode précédente permettant d'extraire des suites temporelles de régions à invariance locale. Au lieu de l'appliquer à la recherche d'objets-vidéo, ils proposent une méthode de regroupement d'objets-vidéo. Pour chaque plan, les suites représentant le même objet sont regroupées dans une classe selon le principe suivant : deux suites appartiennent au même objet si elles sont spatialement proches et si leur distance spatiale est constante dans les trames où elles sont toutes les deux présentes. Puis, les instances d'un même objet dans des plans différents sont regroupées en comparant leur similarité à un seuil.

Avila *et al.* [dALdLA11] proposent une méthode simple de création de résumé de séquences vidéo. Pour chaque trame, un histogramme couleur et des profils de lignes sont calculés. Les profils sont verticaux et horizontaux avec un intervalle spécifique entre deux profils. Les trames sont ensuite regroupées en appliquant l'algorithme K-Means sur les histogrammes et les profils. Le résumé se compose de la trame la plus représentative de chaque cluster. Il s'agit d'un résumé statique, les informations de mouvement ne sont pas prises en compte.

Basharat *et al.* [BZS08] definissent un système de recherche de séquence vidéo basée sur la correspondance de volumes spatio-temporels. Les volumes spatio-temporels sont des objets tridimensionnels représentant des régions et leur évolution au cours du temps. Dans un premier temps, les points d'intérêt sont extraits dans chaque trame et décrits par le descripteur SIFT. Les points similaires dans des trames connexes sont reliés afin de construire des trajectoires. Les volumes spatio-temporels sont obtenus à partir de ces trajectoires en se basant sur le fait qu'une région est composée de points ayant une trajectoire similaire. Les volumes ainsi obtenus sont décrits par des descripteurs de points d'intérêt, de couleur, de texture et de mouvement. La comparaison avec la requête s'effectue à l'aide d'un graphe biparti. Les arêtes du graphe sont valuées par une mesure de similarité comparant les volumes de la requête aux volumes de la séquence vidéo à analyser.

Chevalier *et al.* [CDBPD07] traitent de la recherche d'objets dans des séquences vidéo compressées. La séquence est d'abord segmentée par une ligne de partage des eaux modifiée [MBPL05] appliquée sur une trame de basse résolution (DC) [MSS02]. On obtient ainsi une partition en régions représentant des morceaux d'objets. Les objets sont représentés par un graphe d'adjacence de régions. Une mesure de similarité est ensuite calculée à l'aide d'une méthode de comparaison de graphes.

Gao *et al.* [GLFT09] présentent un système de recherche de plans basée sur l'analyse du mouvement. Le mouvement est caractérisé en plusieurs étapes. Ils calculent en premier lieu le flot optique [HS81] puis le divisent spatialement afin d'obtenir des cubes de flot optique. Ces cubes sont ensuite utilisés pour construire des tenseurs de flot optique (TFO). La dimensionnalité des tenseurs est réduite par une analyse linéaire discriminante [BG98]. La partie recherche du système est basée sur des modèles de Markov cachés.

Liu et Chen [LC09] proposent un système de recherche de séquences vidéo basé sur l'extraction automatique d'objets d'intérêt. En premier lieu, des régions sont extraites et caractérisées à l'aide de SIFT. Ces régions sont ensuite classées en régions des objets d'intérêt ou en régions du fond. Cette catégorisation s'effectue via un algorithme d'EM [DLR77]. Des boites englobantes sont construites autour des objets d'intérêt et le contenu des boites est décrit par divers descripteurs. La partie recherche est effectuée par une nouvelle méthode d'appariement basée sur les ensembles appliquée sur les descriptions.

Moxley *et al.* [MMM10] développent une méthode d'annotation et de correction d'annotation de séquences vidéo. Chaque séquence est décrite par divers descripteurs. Elles sont ensuite projetées dans un graphe, chaque séquence vidéo étant un nœud. Les arêtes sont valuées par une mesure de similarité entre les séquences associées aux nœuds qu'elles relient. Il s'agit ensuite de renforcer l'annotation en utilisant le graphe. Une annotation qui n'est pas affectée à une vidéo mais à un groupe de vidéos similaires à celle-ci sera également affectée à la séquence vidéo considérée. A l'inverse, de mauvaises annotations seront filtrées si elles n'apparaissent pas dans les vidéos similaires. Les auteurs considèrent la méthode comme automatique : nous la considérons comme supervisée car elle nécessite un nombre important de vidéos annotées pour être réellement efficace.

Poullot *et al.* [PCB08] présentent un système de détection de copies prévu pour de grands volumes de données. La première étape consiste à extraire les trames les plus représentatives de

la séquence vidéo. Ces trames sont décrites en utilisant une description « glocale » définie par les auteurs comme un résumé de descriptions locales de points d'intérêt. Ces descriptions sont ensuite indexées pour accélérer le traitement. Le système cherche ensuite à lier les trames similaires en utilisant leurs descriptions. Une fois les trames similaires liées, les séquences similaires sont extraites pour effectuer la détection de copies.

Ren et Zhu [RZ08] définissent une méthode de résumé de séquences vidéo basée sur l'apprentissage automatique. Divers descripteurs sont extraits : un ratio de similarité de pixel (RSP), un ratio de modification de contours (RMC) et le coefficient de corrélation d'histogramme (CCH). Un réseau de neurones est utilisé pour détecter les transitions entre les plans en s'appuyant sur les descripteurs précédents. Les trames représentatives sont ensuite extraites en se basant sur les changements de contours, couleur et texture. Le résumé de la séquence est composé des trames représentatives extraites.

Sivic et Zisserman [SZ08] transposent les principes de la recherche basée sur le texte aux objets-vidéo. Dans chaque trame, des régions sont extraites et décrites par des SIFT. Ces régions sont suivies à travers la séquence vidéo. Après une étape de filtrage, les régions restantes d'un sous-ensemble de la séquence sont regroupées pour créer un vocabulaire visuel. Les régions les plus communes sont rejetées et une structure d'indexation est construite. Les requêtes de l'utilisateur sont des régions d'une séquence qui sont analysées pour déterminer les mots visuels qu'elles contiennent. Les résultats sont recherchés en utilisant la fréquence des mots visuels et la consistence spatiale.

Teixeira et Corte-Real [TCR09] proposent un système de classification d'objets et l'appliquent à la vidéosurveillance. Les objets sont segmentés, puis décrits par SIFT. Les descriptions sont quantifiées en utilisant des sacs de mots visuels basés sur un arbre de vocabulaire visuel. Les objets sont alors classés en utilisant l'algorithme Learn++.MT [MTP04]. Après le classement d'un objet, le modèle visuel de cet objet est mis à jour si nécessaire.

You *et al.* [YLP10] proposent un cadre pour la classification vidéo et l'analyse d'évènements. Leur idée est d'effectuer conjointement la classification de séquences vidéo et l'analyse d'évènements. Ils décrivent les séquences vidéo MPEG à l'aide de différents descripteurs (mouvement, couleur dominante, etc.). Puis, en utilisant des modèles de mélanges de gaussiennes [GGM04] et des modèles de Markov cachés, ils effectuent la classification des séquences en genre et l'analyse d'évènements. Ce cadre nécessite donc l'implication de l'utilisateur pour fournir des données d'entrainement étiquetées afin d'effectuer un apprentissage supervisé des modèles.

Zhai *et al.* [ZLTZ07] élaborent des résumés de séquences vidéo basés sur une classification non-supervisée de graphes. Un graphe des K plus proches voisins sur les trames de la séquence est construit. Ce graphe est partitionné en composantes connexes représentant les classes des trames similaires. Si les composantes connexes comprennent plus de nœuds qu'un seuil défini, une méthode ISOMAP [TSL00] suivie d'une classification non-supervisée de modèles de mélanges [FJ02] est appliquée. Ils sélectionnent une trame par classe pour représenter la classe dans le résumé. Comme dans [dALdLA11], l'information de mouvement n'est pas prise en compte.

Le tableau 3.1 résume les différentes caractéristiques des systèmes étudiés, selon la taxonomie introduite dans la section 3.2.2. Les méthodes caractérisées dans le tableau sont discutées dans la suite de la section. Ces articles récents (de 2007 à 2011) traitent majoritairement des séquences vidéo génériques non-compressées (même si les données sont stockées sous forme de séquences vidéo compressées, elles sont décodées préalablement pour être traitées par le système). Nous observons que l'objectif le plus fréquent est la *recherche*. Cela semble relativement logique : en effet, la première chose qu'un utilisateur désire est d'obtenir les séquences vidéo dont il a besoin, surtout si celles-ci sont noyées dans une grande quantité de données. Le résumé de vidéo suscite également l'intérêt de la communauté, puisqu'il a pour but de permettre à l'utilisateur de connaître le contenu d'une vidéo sans avoir à la regarder dans son intégralité, ce qui se traduit par un gain de temps important. Á l'exception de la tâche de *résumé de vidéo*, l'élément le plus courant semble être l'objet. Cependant, l'objet est loin d'être l'échelle la plus utilisée, les échelles globale et région sont les plus communes : en effet, produire une segmentation automatique (en objets) qui ait une signification sémantique reste aujourd'hui encore un problème ouvert. Nous notons que les descripteurs utilisés sont variés et souvent combinés afin d'obtenir de meilleurs résultats. Cependant les descripteurs SIFT sont très utilisés et semblent être les descripteurs les plus efficaces actuellement. Nous notons qu'à l'exception de [BZS08],[MMM10] et [YLP10], la majorité des SFV étudiés implique l'utilisateur

dans le processus seulement au travers du réglage de paramètres. Aucun des SFV étudiés n'implique l'utilisateur de façon semi-supervisée, ce qui nous semble pourtant un moyen fiable et léger pour guider le système, si tant est que ledit système soit capable de passer à l'échelle.

méthodes	Tâches	Données	Elément	Descripteur	Echelle	Implication
[AC07b]	Rec	B,G	Obj	RIL,SIFT	Reg	Param
[AC07a]	Cla	B,G	Obj	RIL,SIFT	Reg	Param
[dALdLA11]	Res	B,G	Tra	Col,PL	Glo	Param
[BZS08]	Rec	B,G	Vid	SIFT,Col,Tex,Mot	Reg	Nul
[CDBPD07]	Rec	C,G	Obj	RAG	Reg	Param
[GLFT09]	Rec	B,G	Pla	OFT	Blo	Param
[LC09]	Rec	B,G	Vid	Divers	Obj	Param
[MMM10]	Ann	B,G	Vid	Divers	Glo	Sup
[PCB08]	Cop	B,G	Tra	Glocal	PI	Param
[RZ08]	Res	B,G	Tra	RSP,RMC,CCH	Glo	Param
[SZ08]	Rec	B,G	Obj	SIFT	Reg	Param
[TCR09]	Cla	B,S	Obj	SIFT	Obj	Sup
[YLP10]	Cla	C,G	Vid	Divers	Glo	Sup
[ZLTZ07]	Res	B,G	Vid	KNNG	Glo	Param

TABLE 3.1 – Caractérisation des approches récentes en fouille vidéo.

3.3.2 Vers une fouille vidéo orientée objet

L'étude des tendances récentes dans le domaine de la fouille vidéo montre que, si les échelles objet et région semblent être adoptées, la séquence vidéo intégrale et les trames sont toujours les éléments les plus couramment traités dans les systèmes commerciaux alors que la recherche se focalise plutôt sur des méthodes utilisant l'objet comme élément de base. En effet, dans le contexte de l'analyse vidéo, les informations sont principalement apportées par des objets et leur évolution temporelle. En exploitant l'environnement des objets, comme le fond ou les objets adjacents, il est également possible de préciser la sémantique. De la même façon, les relations spatio-temporelles entre les objets peuvent être exploitées pour enrichir le processus de fouille. Dans cette section, nous expliquons dans quelle mesure les caractéristiques d'un Système de Fouille Vidéo Orienté-Objet (SFVOO) sont différentes d'un SFV classique.

3.3.2.1 Caractéristiques d'un SFV orienté objet

Choisir l'objet comme élément d'un SFV a une influence importante sur les autres caractéristiques (sauf pour les objectifs de la fouille car ils peuvent a priori être tous accomplis en s'appuyant sur l'objet comme élément).

Données

L'impact sur le type de données est relativement faible. Même s'il n'est pas trivial d'extraire des objets-vidéo depuis un flux vidéo compressé, des solutions existent, tels les travaux de [BRS04], [TCAA05] et [HCC06] et la norme MPEG-4 qui permet un codage objet [AEH+00]. De plus, l'approche orientée-objet est adaptée à tout type de séquence vidéo : générique ou spécifique. Mais, intuitivement, il semble plus simple de traiter des séquences spécifiques puisque la variabilité des types d'objets sera plus limitée. Au contraire, le traitement de séquences génériques nécessite l'existence de méthodes d'extraction adaptées à chaque type d'objets ou l'implication de l'utilisateur pour apporter la sémantique nécessaire.

Échelle

L'approche orientée-objet nécessite de considérer deux types d'échelles complémentaires : une échelle pour les objets eux-mêmes et une échelle pour le contexte dans lequel ils évoluent (communément appelé le fond), tel qu'illustré en figure 3.4. Les deux trames présentées sont issues de la séquence vidéo *STS-53 Launch and Landing, segment 02 of 5* du projet *OpenVideo*[4] et comportent le même objet, la navette spatiale *Discovery*. Dans le cadre d'une base contenant de nombreuses séquences vidéo de cette navette spatiale, l'utilisateur peut vouloir distinguer les différentes situations dans lesquelles se trouve cette navette (par exemple celles présentées dans les deux séquences vidéo). Décrire seulement l'objet, qui est ici identique dans les deux séquences, ne permet pas de les distinguer. Il est donc nécessaire de pouvoir également décrire l'environnement propre à l'objet afin de pouvoir distinguer la navette sur sa rampe de lancement de la navette en pleine ascension. Cette observation est partagée par d'autres travaux récents [JYTK11]. Les échelles possibles pour décrire l'objet sont celles inférieures ou égales à l'échelle objet tandis que les échelles possibles pour le contexte sont celles supérieures à l'échelle objet (cf. figure 3.2).

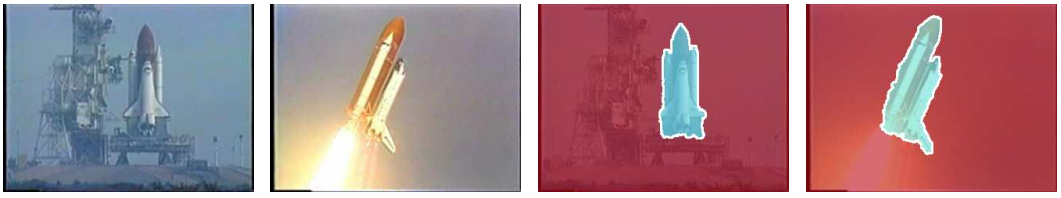


FIGURE 3.4 – Illustration des échelles *objet* et *contexte* sur deux trames d'une séquence vidéo de la navette spatiale *Discovery* (gauche) et leurs segmentations respectives (droite) en la navette *Discovery* (bleu) et son environnement (rouge).

Descripteurs

Tous les descripteurs peuvent être considérés dans un SFV orienté objet, mais leur usage est différent de celui fait par les autres types de SFV. En effet, ils doivent permettre de décrire l'objet et/ou son environnement. En outre, des descripteurs de mouvement doivent permettre de décrire le mouvement général de l'objet mais aussi son mouvement interne dans le cas d'objets complexes. De façon plus générale, les descripteurs utilisés dans une approche objet doivent bien sûr être discriminants (objet vs. objet ou objet vs. environnement), mais de plus, les différences (ou les similarités) qu'ils mettent en valeur doit avoir une signification sémantique.

Implication de l'utilisateur

Dans un SFV orienté-objet, le rôle de l'utilisateur est prédominant. La perception d'un objet est subjective et peut donc être différente d'un utilisateur à l'autre. Pour nous, l'utilisateur doit être profondément impliqué dans le processus de fouille afin de pouvoir guider efficacement ce dernier. Cependant, même si cette intervention est fondamentale pour le SFV orienté-objet, elle doit rester intuitive et légère afin d'être efficace et peu coûteuse en temps. Ces propriétés peuvent être assurées, tel que présenté dans la section 3.3.2.3, au travers de la mise en place d'un retour de pertinence mais également par des méthodes d'apprentissage actif ou semi-supervisé.

3.3.2.2 Extraction d'objets-vidéo

Afin d'extraire les objets d'une séquence vidéo, la plupart des méthodes intègrent une étape de segmentation. Seules font exception les méthodes dédiées aux séquences vidéo compressées selon un schéma orienté-objet, voire celles basées sur des points d'intérêt. Pour les SFV, la segmentation vidéo consiste la plupart du temps en un découpage en plans [LHV03]. Au contraire, pour les SFV orientés-objet, l'étape d'extraction doit produire des objets et décrire leur évolution temporelle. Comme nous l'avons souligné dans l'introduction de cette thèse, la principale difficulté rencontrée

4. The Open Video Project, http ://www.open-video.org/

ici est de combler le fossé sémantique séparant les données brutes des objets réels. Cette extraction peut être effectuée pendant la phase d'encodage de la séquence vidéo dans le cas des données compressées, ou plus généralement avec une segmentation. Il est possible de ne travailler qu'avec des séquences encodées au format MPEG-4 [AEH+00]. En effet, cette norme de compression possède un mode objet qui repose sur une segmentation de la séquence vidéo en objets-vidéo. Cependant, limiter un SFV au traitement unique de séquences vidéo au format MPEG-4 restreint son intérêt et constitue une importante contrainte. De plus, la norme MPEG-4 n'impose aucunement un codage en objets et ne spécifie aucune méthode permettant la segmentation de la séquence en objets avant l'encodage. Il est donc possible que même en nous restreignant aux séquences vidéo au format MPEG-4, nous soyons obligés d'effectuer une segmentation en objets-vidéo. Ceci a motivé le fait que, malgré les capacités de gestion d'objets-vidéo de cette norme, nous ayons choisi une autre voie de recherche (décrite dans le chapitre 2).

Puisqu'un objet est supposé posséder une sémantique, la segmentation nécessite des méthodes intégrant de telles informations. Plus généralement, l'introduction de sémantique est un point important des SFV orientés-objet que nous détaillons dans la section suivante. Il existe des méthodes de segmentation vidéo sémantique, tel que le chroma-keying et la segmentation semi-automatique [Tek05]. Le chroma-keying consiste à filmer un objet sur un fond uniforme rendant triviale la segmentation en objet-vidéo. La sémantique est apportée par l'utilisateur qui fournit l'objet à filmer et donc à extraire. Dans la segmentation semi-automatique, l'utilisateur marque les contours de chaque objet-vidéo lors de sa première apparition dans la séquence vidéo. Ces deux méthodes peuvent s'avérer très coûteuses en temps pour l'utilisateur et ne peuvent être appliquées sur des bases de séquences vidéo de grande taille ou génériques dans le cas du chroma-keying. Il est donc nécessaire de disposer de méthodes sémantiques de segmentation vidéo adaptées à ces bases de séquences vidéo. Plus généralement, l'introduction de sémantique dans un SFV orienté-objet est un point critique qui est abordé dans la section suivante.

3.3.2.3 Introduire de la sémantique

Un SFV orienté-objet nécessite d'introduire de la sémantique dans le processus de segmentation ainsi que dans le processus de fouille. Les descripteurs bas-niveau présentés dans la section 3.2.2.2 fournissent des représentations numériques mais ne sont pas capables de donner une perception sémantique de l'objet comme le ferait un être humain. Même si les descripteurs récents (SIFT [Low99], sac de mots visuels [SZ08], etc.) donnent de très bons résultats, le fossé sémantique n'est toujours pas comblé.

Pour le réduire un peu plus, une solution serait de fournir des exemples pour chaque objet potentiellement présent dans une séquence vidéo, mais cette approche n'est pas réaliste. Exploiter un mécanisme tel que le retour de pertinence [RL03] semble être une solution plus judicieuse. À l'issue du processus de fouille, l'utilisateur évalue un échantillon du résultat qu'il peut corriger si nécessaire (à l'instar d'un apprentissage par renforcement). En fonction de cette évaluation, le processus peut être itéré pour tenir compte de la connaissance introduite par l'utilisateur (via l'évaluation et la correction de l'échantillon). Ce processus itératif est moins coûteux en temps que la production d'exemples complets nécessaires dans une approche supervisée. Cela entraîne une personnalisation du résultat. Enfin, le retour de pertinence peut aussi être appliqué dès l'étape de segmentation (afin de l'améliorer). En effet, à notre avis : meilleure est la segmentation, plus aisée mais surtout plus pertinente est la fouille. Enfin, créer des descripteurs dédiés aux objets considérés est également une solution intéressante mais ce problème reste ouvert dans le contexte de séquences vidéo génériques et ne sera pas abordé ici.

Les observations de cette section sont exploitées dans la section suivante où nous proposons un schéma générique pour la fouille vidéo orientée-objet.

3.4 Notre proposition : Video Object Mining Framework

Dans cette section, nous proposons un schéma générique pour la fouille vidéo orientée-objet : Video Object Mining Framework (VOMF). Ce schéma présente les caractéristiques identifiées dans la section précédente :

- utilisation des objets-video comme éléments du processus de fouille ;
- implication de l'utilisateur dans le processus d'extraction des objets-vidéo ;
- utilisation de descripteurs sémantiques à l'échelle de l'objet ;
- implication de l'utilisateur dans le processus de fouille.

3.4.1 Le processus de fouille proposé

Le processus de fouille proposé par VOMF est illustré dans la figure 3.5. Ce processus se compose de quatre étapes :

1. Segmentation des objets-vidéo ;
2. Évaluation de ces objets-vidéo ;
3. Caractérisation et fouille de la base d'objets-vidéo ;
4. Évaluation du résultat de cette fouille.

La première étape est l'extraction automatique des objets-vidéo présents dans les séquences vidéo de la base vidéo. Un échantillon des objets obtenus est évalué par l'utilisateur via un système de retour de pertinence. Si les segmentations de l'échantillon sont approuvées par l'utilisateur, l'ensemble des objets est transmis à l'étape de fouille (indexation, classification, recherche, etc.). Dans le cas contraire, les erreurs de segmentation sont identifiées par l'utilisateur, qui introduit de la sémantique par ce biais. Une nouvelle segmentation est alors construite en s'appuyant sur les segmentations déjà effectuées et la sémantique apportée par l'utilisateur. Ce cycle est répété jusqu'à ce que l'utilisateur soit satisfait par les objets obtenus. Cependant, il faut veiller à ce que le cycle ne soit pas répété de trop nombreuses fois pour que le temps nécessaire à l'utilisateur pour générer la base d'objets-vidéo soit acceptable. Les résultats de la fouille vidéo sont également évalués par l'utilisateur en même temps, via un retour de pertinence sur un échantillon du résultat. Si l'échantillon évalué est satisfaisant, le traitement est terminé. Sinon, à l'instar de la segmentation, l'utilisateur peut corriger l'échantillon. Dans ce cas, la fouille vidéo est relancée et exploite les corrections de l'utilisateur pour améliorer le résultat. L'utilisateur est placé au centre du système. Il supervise le processus de fouille à travers le retour de pertinence et introduit de la sémantique en corrigeant les résultats inappropriés. Pour être efficace, le retour de pertinence ne doit pas être exhaustif. Il faut au contraire que seules quelques évaluations/corrections soient suffisantes pour influencer profondément les processus de segmentation et de fouille afin que le retour de pertinence ne soit pas trop coûteux en temps pour l'utilisateur.

3.4.2 Application au regroupement d'objets-vidéo

Le *regroupement* ou *classification non-supervisée*, consiste à regrouper les éléments d'un ensemble de données en différentes classes de telle manière que la similarité intra-classe soit maximale et que la similarité inter-classe soit minimale, et ce, sans disposer de données d'apprentissage. Appliqué aux objets-vidéo, cela revient à regrouper dans un même ensemble des objets-vidéo similaires et dans des ensembles différents des objets-vidéo non similaires. L'objectif d'un regroupement d'objets-vidéo peut-être de différentes natures :

- structuration : si l'on dispose d'une base d'objets-vidéo non structurée, regrouper les objets-vidéo similaires permet de proposer une structuration de la base d'objets-vidéo. De plus, si la méthode produit une hiérarchie des regroupements, on peut choisir le niveau de structuration que l'on désire ;
- annotation : plutôt que d'annoter individuellement chaque objet-vidéo et ainsi d'augmenter le risque d'erreur (faute d'orthographe, synonymie, etc.), on annote la classe et tous les objets-vidéo de la classe héritent de l'annotation ;

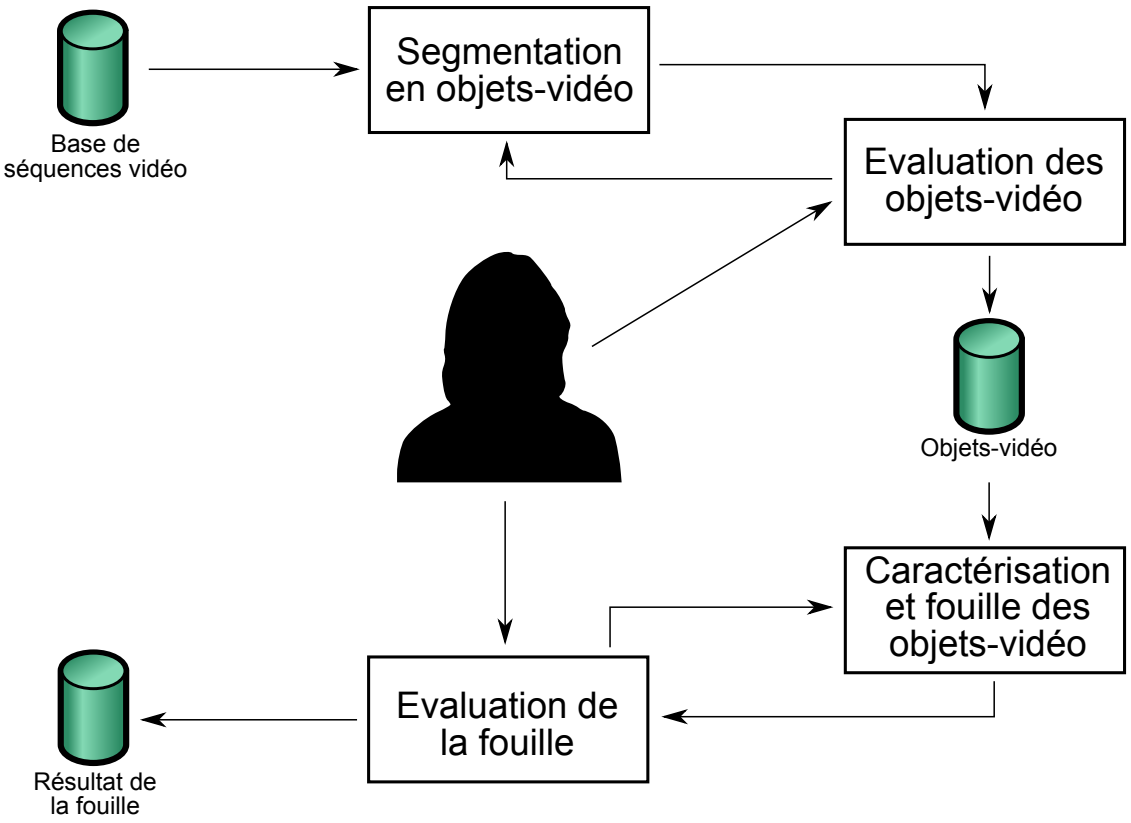


FIGURE 3.5 – VOMF : Video Object Mining Framework.

- indexation : dans un contexte de recherche par le contenu, plutôt que de comparer la requête à l'ensemble de la base d'objets-vidéo, on la compare à une représentation de chaque classe. Si la requête et la représentation sont proches, la requête peut alors s'effectuer au sein de cette classe. Cela peut également être appliqué dans le contexte de la détection de copies ;
- exploration : fournir un aperçu d'une base d'objets-vidéo est une tâche complexe pour les bases de grande taille. Le regroupement des objets-vidéo permet d'avoir une vue d'ensemble grâce aux classes dont on peut extraire les objets-vidéo les plus représentatifs.

La classification non-supervisée d'objets-vidéo provenant d'une base est donc une étape préliminaire permettant de mener plus efficacement d'autres opérations de fouille au sein d'une base structurée. C'est pour cette raison que nous avons choisi d'étudier et d'implémenter un système dédié à cette tâche. En effet, disposer d'un système efficace de regroupement des objets-vidéo permettra ensuite de réaliser les quatre objectifs cités précédemment. Cependant, dans ce manuscrit nous n'implémentons et n'évaluons que le système de regroupement. Nous ne proposons pas de système de structuration, annotation, indexation ou exploration.

La tâche de regroupement est une tâche courante dans le domaine du traitement de la vidéo [Ant02]. On trouve ainsi plusieurs méthodes dédiées à un ou plusieurs des 4 objectifs que nous avons cités précédemment mais n'utilisant pas forcément des objets-vidéo comme éléments. Urruty *et al.* [UBD05] proposent le système *Kpyr* dédié à l'indexation de films d'entreprise. Ils décrivent les séquences vidéo en utilisant des descripteurs issus de la norme MPEG-7 [MSS02]. Le regroupement est effectué par l'algorithme K-Means [Mac67]. Zhong *et al.* [ZZC96] regroupent les scènes et les plans similaires afin de permettre une navigation efficace au sein d'une séquence vidéo ainsi que son annotation, le regroupement est basée sur un K-Means flou [GM88].

Outre l'utilisation de méthodes de regroupement comme étape préliminaire, elles ont aussi été utilisées pour créer des *résumés de séquences vidéo* [Gui06, FGMP08, BMEF09] et plus particulièrement le travail doctoral de J. Huart [Hua07] qui résume les séquences vidéo par des objets particuliers présents dans la vidéo appelés *objets-clés*. Cette dernière approche se rapproche de nos travaux de par son approche orientée-objet. Les objets-clés ne correspondent pas à nos objets-vidéo et sont obtenus par un processus automatique. Le regroupement n'est pas effectué sur une base vidéo mais sur une séquence vidéo qu'il s'agit de résumer. Ce résumé se compose d'images-clés ou de courts plans représentant les moments les plus significatifs de la vidéo. Il y a également de nombreux travaux qui ne portent que sur le regroupement vidéo sans l'utiliser pour l'accomplissement d'une tâche particulière. Ainsi, Turaga *et al.* [TVC09] ont comme objectif le regroupement des actions effectuées au sein d'une séquence vidéo. Ils caractérisent les actions grâce à plusieurs descripteurs et utilisent un algorithme de coupe normalisée [SM00] pour effectuer le regroupement. Agnihotri et Dimitr [AD00] regroupent des segments de vidéo par un algorithme des k-plus-proches-voisins [CH67] en les ayant préalablement décrit avec des SuperHistogrammes [DMAE99]. Schroff *et al.* [SZB09] proposent un système qui regroupe les plans de videos qui ont été filmés au même endroit. Les plans sont décrits par des histogrammes de textons [CDF+04] et regroupés par une méthode basée-modèle [KKM02]. Lee *et al.* [LOH05] proposent un système basé sur une segmentation des vidéo en graphe spatio-temporel de régions homogènes. De ces graphes, ils extraient automatiquement des objets qu'ils regroupent par une méthode d'espérance-maximisation [DLR77].

Les méthodes de classification non supervisée dédiées aux séquences vidéo que nous avons présentées ne sont soit pas appliquées à des objets-vidéo, soit, à des objets obtenus automatiquement et ne possédant pas la sémantique apportée par l'utilisateur dans nos objets-vidéo. De plus, ces approches n'impliquent pas l'utilisateur dans le processus de regroupement, elles ne permettent donc pas de personnaliser le regroupement en fonction des besoins de chaque utilisateur. Dans la suite de ce chapitre, nous proposons et implémentons un système de regroupement d'objets-vidéo qui implique l'utilisateur conformément au cadre VOMF.

3.5 Implantation de VOMF pour le regroupement d'objets-vidéo

Dans cette section, nous présentons le système que nous avons implémenté en nous basant sur VOMF. Dans un premier temps nous présentons les différents descripteurs que nous avons utilisés pour caractériser les objets-vidéo. Puis, nous présentons l'algorithme de clustering que nous avons décidé d'utiliser pour valider notre approche. Le système utilisant les descripteurs et l'algorithme présenté se nomme VOX (Video-Object Segmentation and Clustering System). Il est présenté plus précisément dans l'annexe B.

3.5.1 Descripteurs

Il existe de très nombreux descripteurs d'objets-vidéo. Nous présentons les descripteurs utilisés dans la validation expérimentale de la section 3.6. Afin de faliciter la comparaison et la reproductibilité des résultats, nous avons choisi des descripteurs compatibles avec les spécifications de la norme MPEG-7 [MSS02]. La norme MPEG-7 propose quatre familles de descripteurs vidéo [Sik01] : les descripteurs de couleur, les descripteurs de texture, les descripteurs de forme et les descripteurs de mouvement. Dans la suite de cette section, nous choisissons dans chacune des 3 premières familles un descripteur adapté à notre contexte et nous discutons l'utilisation d'un descripteur de mouvement.

3.5.1.1 Couleur

Les descripteurs couleur sont parmi les plus utilisés en fouille de données visuelles. La couleur est en effet une des principales façons de décrire un objet pour un humain. Ces descripteurs présentent une robustesse aux transformations géométriques mais peuvent être sensibles aux variations

d'illumination.

Les histogrammes couleur calculés dans l'espace RVB sont les descripteurs les plus simples mais ils sont très sensibles aux variations d'illumination. Pour cette raison nous n'appliquons pas un descripteur couleur dans l'espace RVB mais dans un espace Teinte-Saturation-Valeur (TSV) [Smi78] qui est moins sensible aux variations d'illumination. La Teinte représente la « couleur » (proche du sens humain des termes jaune, vert, rouge, etc.), la Saturation représente l'« intensité » de cette couleur et la Valeur représente la « brillance » de cette couleur. La formule de la transformation d'une image RVB (où chaque composante est normalisée en $[0, 1]$) en TSV est la suivante :

$$
\begin{aligned}
T_{TSV} &= \max\{R,V,B\} \\
S_{TSV} &= \left\{ \begin{array}{ll} \dfrac{\max\{R,V,B\} - \min\{R,V,B\}}{\max\{R,V,B\}} & \text{si } \max\{R,V,B\} > 0 \\ \emptyset & \text{sinon} \end{array} \right\} \\
V_{TSV} &= \left\{ \begin{array}{r} \emptyset \text{ si } S_{TSV} = \emptyset \\ 60^\circ \times \dfrac{V-B}{\max\{R,V,B\} - \min\{R,V,B\}} \text{ si } \max\{R,V,B\} = R \text{ et } V \geq B \\ 60^\circ \times \dfrac{V-B}{\max\{R,V,B\} - \min\{R,V,B\}} + 360^\circ \text{ si } \max\{R,V,B\} = R \text{ et } V \leq B \\ 60^\circ \times \dfrac{B-R}{\max\{R,V,B\} - \min\{R,V,B\}} + 120^\circ \text{ si } \max\{R,V,B\} = V \\ 60^\circ \times \dfrac{R-V}{\max\{R,V,B\} - \min\{R,V,B\}} + 240^\circ \text{ si } \max\{R,V,B\} = B \end{array} \right\}
\end{aligned}
\tag{3.1}
$$

En pratique, nous remplaçons $\emptyset$ par 0 dans l'implantation de la tranformation RVB $\rightarrow$ TSV. Afin de réduire encore la sensibilité de notre descripteur, nous quantifions l'espace TSV. Nous quantifions les composantes S et V uniformément en 3 valeurs (au lieu de 256 pour 8 bits). La composante T est quantifiée en 7 valeurs mais de façon non-uniforme (cf. figure 3.6). En effet, la composante de teinte n'est pas uniforme, certaines "couleurs" occupant une place plus importante que d'autres. Cet espace sous-quantifié contient 63 couleurs différentes au lieu des 16,7 millions de couleurs d'un espace RVB en 8 bits par composante. L'apport de ce type de quantification non-uniforme a été validé dans [Apt08].

Afin de caractériser la couleur des objets-vidéo dans cet espace, nous utilisons un *histogramme de structure de couleur* (HSC) [MOVY01]. Il présente les avantages d'offrir une représentation compacte (appliqué à l'espace TSV quantifié, c'est un histogramme de 63 bins), d'être rapide à calculer et d'être intégré à la norme MPEG-7. Le HSC donne une information de couleur globale comme un histogramme classique mais également une information sur la distribution locale des couleurs. Pour ce faire, on utilise un voisinage carré de 8x8 que l'on translate en chaque point de l'image. Chaque bin de l'histogramme représente une couleur dans l'espace TSV quantifié, sa valeur correspond au nombre de fois où au moins un pixel du voisinage était de cette couleur. L'histogramme obtenu est ensuite normalisé pour que la somme de ses valeurs soit égale à 1.

Ce descripteur a été conçu pour décrire des objets spatiaux. Nous pourrions l'étendre directement et considérer un voisinage spatio-temporel. Cependant, une telle approche poserait des problèmes en cas de variation d'échelle (objet-vidéo qui se rapprochent ou s'éloignent) et fort mouvement, et nécessiterait dans ces cas des éléments structurants adaptatifs. Nous avons donc choisi de garder la définition spatiale des HSC mais de prendre en compte la redondance d'information des objets-vidéo. Notre approche est illustrée dans la figure 3.7. Afin de l'appliquer sur des séquences vidéo nous le calculons indépendamment pour chaque trame de l'objet-vidéo. Nous calculons ensuite les distances entre toutes les descriptions obtenues. La description la plus proche de toutes les autres (medoide) est choisie pour représenter l'objet-vidéo. Nos descriptions se présentant sous

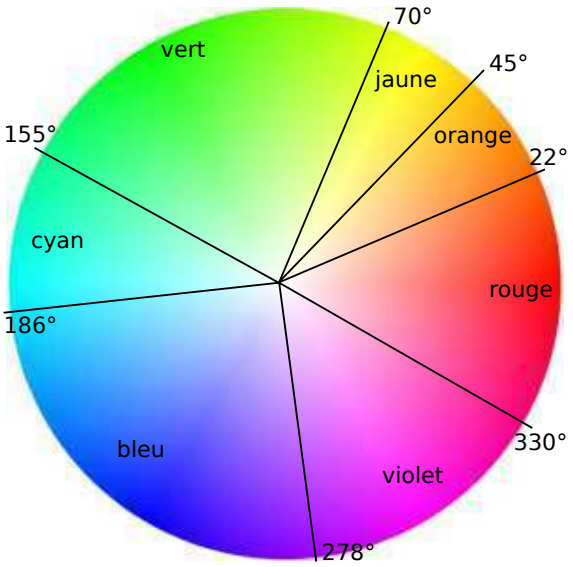


FIGURE 3.6 – Quantification non-uniforme de la teinte.

la forme d'histogrammes, nous aurions pu calculer la description moyenne et l'utiliser comme description de l'objet-vidéo. Mais, la description moyenne ne représente pas forcément une description existante. Cela induit le risque d'introduire une nouvelle description non présente dans les descriptions existantes des trames de l'objet-vidéo et possiblement non représentative de l'objet-vidéo. L'utilisation de la description médoïde permet de résoudre ce problème.

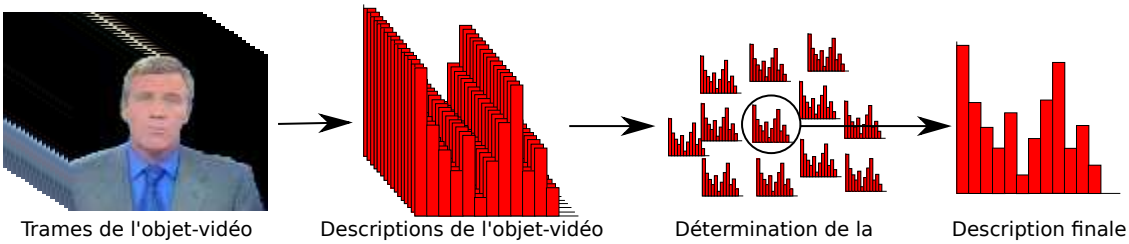


FIGURE 3.7 – Détermination de la description d'un objet-vidéo.

3.5.1.2 Texture

La texture est également très utilisée en fouille de données visuelles. Une texture consiste en la répétition d'un motif dans une ou plusieurs orientations spatiales (par exemple tuiles de toit, écailles de poisson) ou en un ensemble de petits éléments distribués aléatoirement (par exemple sable, herbe, etc.).

A l'instar des descripteurs de couleur, il existe une multitude de descripteurs de texture. Nous utilisons la *covariance morphologique* [Mat67, Mat75]. Ce descripteur est compatible avec la norme MPEG-7. Nous utilisons ce descripteur car son efficacité a été validée dans [Apt08] et il représente une description compacte de la texture. Notons que de meilleurs résultats ont été obtenus en combinant la covariance avec une *granulométrie* [Mat67, Mat75]. Cependant, si la covariance présente déjà un coût calculatoire important celui de la combinaison de la covariance et d'une granulomètrie est encore plus important, nous utiliserons donc uniquement la covariance. La covariance K est définie par :

$$K(f)_{P_{2,v}} = \text{Volume}(\varepsilon_{P_{2,v}}(f))/\text{Volume}(f) \tag{3.2}$$

où $P_{2,v}$ est un couple de points tel que le second point est obtenu par une translation de vecteur v du premier. $\varepsilon_{P_{2,v}}(f)$ est l'érosion de f par $P_{2,v}$ et Volume(f) désigne le volume de l'image f ou la somme de ses pixels. Dans le cas d'image couleur, on somme la norme euclidienne (L_2) des valeurs RVB des pixels de l'image f.

La covariance est appliquée en variant la longueur de la translation v afin de caractériser les motifs de périodes différentes. Nous appliquons également les translations dans différentes orientations afin d'être invariant à la rotation.

Ce descripteur ayant été conçu pour des objets spatiaux et non spatio-temporels nous utilisons également la description médoïde pour décrire l'objet-vidéo.

3.5.1.3 Forme

Si la couleur et la texture donnent des informations sur le contenu d'un objet, sa forme donne une information sur sa définition spatiale et est, à ce titre, également très utilisée en fouille de données visuelles.

La forme étant une information importante pour décrire un objet, de nombreuses méthodes pour la caractériser ont été développées. Dans notre application, nous utilisons les *descripteurs génériques de Fourier* [ZL02]. Nous avons choisi d'utiliser ces descripteurs car ils ont un coût calculatoire faible, sont compatibles avec la norme MPEG-7 et produisent une description compacte. De plus, une étude comparative [ZL02] a validé l'efficacité de ces descripteurs. Ces descripteurs consistent en une transformée polaire de Fourier de la forme que l'on souhaite caractériser. La transformée polaire obtenue est invariante à la translation. Afin de rendre la description également invariante à l'orientation et à l'échelle, les auteurs appliquent une étape de normalisation de l'orientation et de l'échelle.

Comme pour les descripteurs précédents, nous utilisons la description medoide pour décrire de la forme de l'objet-vidéo.

3.5.1.4 Mouvement

Le mouvement, s'il n'est pas utilisé pour les images fixes, est un élément important des systèmes de fouille vidéo. Les descripteurs qui lui sont dédiés permettent la caractérisation du mouvement global d'un objet-vidéo et/ou du mouvement interne d'un objet-vidéo. A l'instar des autres familles de descripteurs, il existe un grand nombre de descripteurs de mouvement. Néanmoins, si le mouvement est un élément-clé il pose des questions fondamentales. Par exemple, un objet se déplaçant de gauche à droite ou de droite à gauche est-il fondamentalement le même objet ? Le besoin de différencier des objets selon leur trajectoire répond à un besoin très précis, par exemple différencier des gens entrant dans un endroit des gens qui en sortent. Dans ce document, nous avons plutôt considéré qu'il s'agissait du même objet et avons préféré ignorer cette information, évitant ainsi de perturber le regroupement des objets-vidéo. Une information intéressante serait, par contre, de savoir si un objet est en mouvement ou non afin d'être capable de distinguer une voiture à l'arrêt d'une voiture en déplacement par exemple. Les informations de mouvement seront donc incluses dans de futures versions de notre système mais pas dans la version présentée ici.

3.5.2 méthode de regroupement

Il existe un grand nombre de méthodes de clustering, disposant chacune de qualités et de défauts. Le lecteur intéressé pourra notamment trouver dans [Ber02, Fas99, HBV01, JMF99, XI05] des panoramas du domaine. Nous détaillons les différentes caractéristiques d'une méthode de regroupement. Puis, nous présentons la méthode que nous utilisons avant d'en proposer une version impliquant l'utilisateur.

3.5.2.1 Caractéristiques des méthodes de regroupement

Lorsque l'on choisit une méthode de regroupement, il y a principalement trois choix à faire. La méthode choisie dépendra de ces choix.

Hiérarchique vs. partitionnement

Le regroupement par partitionnement génère une partition unique des données alors que le regroupement hiérarchique produit une suite de partitions représentée par un dendrogramme. L'avantage des méthodes hiérarchiques est qu'elles permettent de choisir a posteriori le niveau de partition désiré en coupant le dendrogramme à une certaine hauteur. Elles peuvent également créer une hiérarchie entre les classes ce qui permet d'obtenir différentes granularité de regroupement pour une même classification. Cependant, le dendrogramme représentant un nombre élevé de partitions (il y autant de niveaux de partitions qu'il y a d'objets), le regroupement hiérarchique est gourmand en mémoire. A l'inverse, le regroupement par partitionnement est peu gourmand en mémoire mais ne permet pas de choisir a posteriori la granularité du partitionnement. Il produit une partition unique de l'ensemble des objets.

À notre avis, le regroupement d'objets-vidéo dépend des besoins de l'utilisateur. Ainsi, nous utilisons une méthode de regroupement hiérarchique afin de permettre à l'utilisateur de choisir le niveau de partitionnement qu'il désire.

Dur vs. flou

Les méthodes de regroupement dures affectent chaque élément à une seule classe alors que les méthodes floues assignent, à chaque élément, un degré d'appartenance pour chacune des classes. Le regroupement flou est particulièrement utile lorsque les différentes classes que l'utilisateur désire sont proches, il a cependant un coût calculatoire plus élevé que le regroupement dur. Il est possible de convertir un regroupement flou en regroupement dur, en assignant chaque élément à la classe où son degré d'appartenance est le plus élevé.

L'objectif de notre système de regroupement est de découvrir les différents ensembles d'objets-vidéo contenus dans une base de séquences vidéo. L'utilisation d'une méthode dure est adaptée à cet objectif et nous souhaitons, pour le moment, qu'un objet-vidéo n'appartienne qu'à une seule classe.

Ascendant vs. descendant

Lorsque l'on dispose d'un ensemble de données que l'on désire regrouper sous forme d'une hiérarchie, deux stratégies sont possibles. L'approche descendante consiste à regrouper initialement tous les objets dans une seule classe. Puis, itérativement on divise une des classes de la partition en deux classes, par exemple celle ayant la plus faible similarité interne. Ce processus est répété jusqu'à satisfaction d'un critère d'arrêt ou jusqu'à ce qu'on obtienne une classe par objet. A l'inverse, l'approche ascendante crée une partition initiale où chaque objet est dans une classe dont il est le seul élément. Puis, itérativement on fusionne deux classes, par exemple celles dont les éléments sont les plus similaires. Ce processus est répété jusqu'à satisfaction d'un critère d'arrêt ou jusqu'à ce qu'on obtienne une classe contenant tous les objets. Notons qu'il est plus difficile et coûteux calculatoirement de diviser des classes plutôt que de les fusionner.

Nous avons choisi d'utiliser l'approche ascendante, principalement à cause de son coût calculatoire plus restreint et en l'absence d'un critère de choix de la classe à diviser.

Les différents choix que nous avons effectués nous conduisent à utiliser une méthode de clustering hiérarchique ascendant dur [KR90] pour notre système. Nous présentons cette méthode dans la section suivante. Ensuite, nous étudions comment impliquer l'utilisateur dans le processus pour obtenir des résultats personnalisés.

3.5.2.2 Regroupement hiérarchique ascendant

Le principal intérêt d'une approche hiérarchique est de disposer d'une hiérarchie des classes. Cette hiérarchie permet à l'utilisateur de choisir *a posteriori* le niveau de granularité qu'il désire pour la classification. En partant de la partition finale qui contient tous les objets-vidéo, il peut parcourir l'arbre des fusions successives afin d'obtenir les classes qu'il désire (cf figure 3.8). Par exemple, un utilisateur pourrait se contenter d'une classe regroupant tous les objets-vidéo représentant des êtres humains (dans ce cas on utiliserait la coupe donnant 2 classes dans la figure) alors qu'un autre pourrait vouloir faire la distinction entre les êtres humains qui marchent et les êtres humains qui sont immobiles face à une caméra (dans ce cas on utiliserait la coupe donnant 3 classes dans la figure). Ces deux désirs représentent deux niveaux différents dans la hiérarchie des partitions (si bien sûr l'on dispose d'une combinaison de descripteurs permettant de discriminer les humains assis des humains debouts).

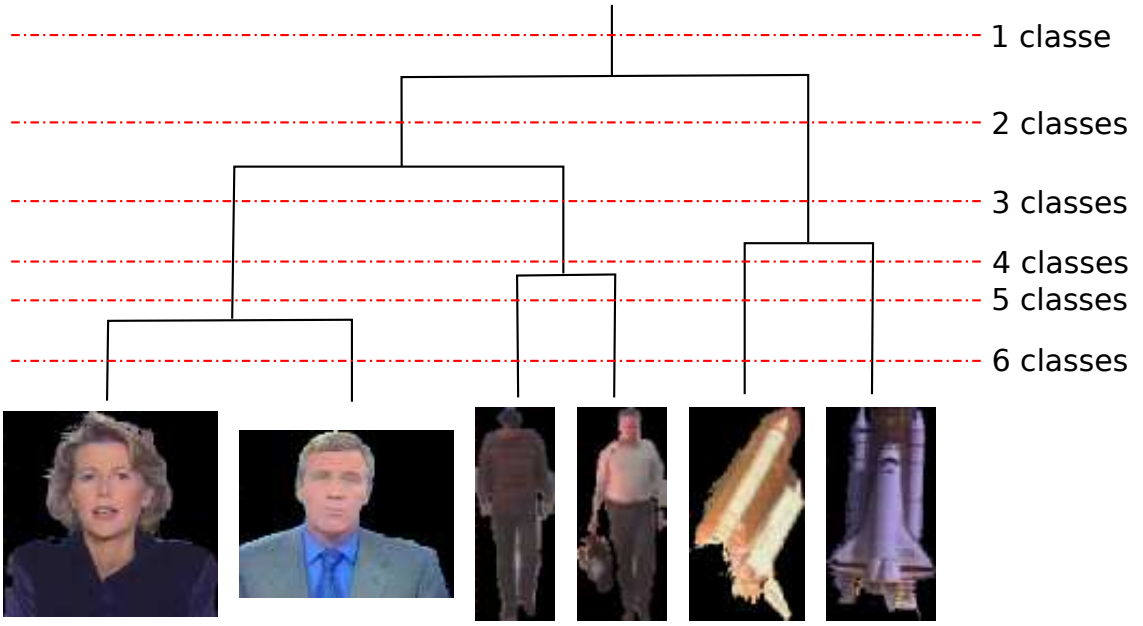


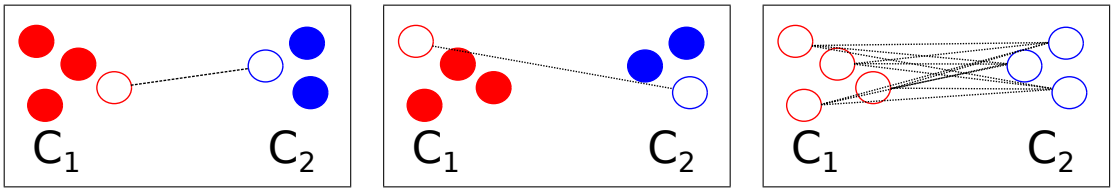
FIGURE 3.8 – Exemple de partitionnement hiérarchique.

Pour les méthodes hiérarchiques simples, les différentes variantes ne diffèrent qu'en la façon d'utiliser les mesures de distance. On parle de *lien simple* [Sib73], *lien moyen* [Voo86] et *lien complet* [Def77] illustrés dans la figure 3.9 et définis par la formule commune suivante :

$$D(C_1, C_2) = operation\,\{d(x, y) | x \in C_1, y \in C_2\} \tag{3.3}$$

où $D(C_1, C_2)$ est la mesure de distance entre deux classes C_1 et C_2, $d(x, y)$ est une mesure quelconque de distance et *operation* signifie *minimum* pour le *lien simple*, *moyenne* pour le *lien moyen* et *maximum* pour le *lien complet*. Leur complexité en $O(n^2)$ (n étant le nombre d'objets à regrouper) rend ces méthodes particulièrement coûteuses lorsque le nombre d'objets à classifier est important. Dans notre implantation, nous utilisons le *lien moyen* car c'est le seul lien qui prend en compte l'ensemble des objets de chaque classe.

La sélection d'une partition dans la hiérarchie des partitions permet une certaine personnalisation des résultats. Cependant, les descripteurs ne sont pas toujours capables de discriminer les objets-vidéo selon les désirs précis d'un utilisateur. Pour que les partitions obtenues soient plus en adéquation avec les désirs de l'utilisateur, nous devons l'impliquer plus profondément dans le processus. Nous étudions cette possibilité dans la section suivante.



lien simple *lien complet* *lien moyen*

FIGURE 3.9 – Les différents types de liens utilisables par les méthodes hiérarchiques.

3.5.2.3 Regroupement hiérarchique ascendant contraint

La méthode de regroupement hiérarchique ascendant présentée précédemment est par nature non-supervisée. Elle ne nécessite aucune connaissance de la base de données. On lui oppose généralement, les approches supervisées qui nécessitent des exemples pour toutes les classes recherchées. Depuis un peu plus d'une dizaine d'années, une troisième voie est apparue, celle des approches semi-supervisées. Ces approches partent du principe que, généralement, l'utilisateur possède intuitivement quelques informations sur la base de données et la classification qu'il en attend. Ces informations sont modélisables sous forme de contraintes.

Cohn [CCM00] introduit plusieurs types de contraintes. Les contraintes les plus simples sont *Must-Link* et *Cannot-Link*, et elles sont données avant la classification ou pendant la classification. Ces contraintes modélisent des relations entre deux objets de la base de données. *Must-Link* signifie que les deux objets liés par cette contrainte appartiennent à la même classe. À l'inverse, *Cannot-Link* signifie qu'ils n'appartiennent pas à la même classe. Deux autres types de contraintes sont intégrables à l'issue d'une première classification : « cet objet n'appartient pas à cette classe » et « cet objet devrait appartenir à cette classe ».

Pour notre système, nous utilisons les contraintes *Must-Link* et *Cannot-Link*. Les deux autres types de contrainte sont difficiles à intégrer dans une partition hiérarchique qui, par nature, n'est pas relancée itérativement. Une contrainte *Must-Link* modifiera la distance entre les deux objets-vidéo concernés en la fixant à zéro, entrainant la fusion des deux objets-vidéo dans la même classe dès la première étape du clustering. Une contrainte *Cannot-Link* fixera la distance entre les deux objets-vidéo à la distance maximale entre deux objets-vidéo de la base d'objets-vidéo. Nous ne fixons pas la distance comme infinie car la partition finale comporte tous les objets-vidéo. Il faut donc que les deux objets-vidéo puissent être fusionnés mais à un niveau élevé de la hiérarchie.

3.6 Expérimentations

Dans cette section, nous présentons les premiers résultats obtenus par notre système pour le regroupement d'objets-vidéo. Dans un premier temps, nous présentons les données que nous utilisons puis nous évaluons le regroupement proposé par notre approche.

3.6.1 Données

Il existe plusieurs jeux et bancs d'essai de données vidéo : CamVid [5] [BFC09] qui contient des séquences vidéo prises depuis le pare-brise d'une voiture en mouvement avec une segmentation et une catégorisation des objets réels présents dans la vidéo, le challenge annuel TRECVID [6] [SOK06] qui propose un contenu vidéo conséquent mais dont aucune des tâches ne s'apparente à du regroupement d'objet-vidéo, LabelMe Video [7] [YRLT09] qui permet à des utilisateurs de segmenter et d'annoter des séquences vidéo en ligne. Cependant, aucune de ces bases ne propose d'objets-vidéo

5. CamVid, http ://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/
6. TRECVID, http ://trecvid.nist.gov/
7. LabelMe Video, http ://labelme.csail.mit.edu/VideoLabelMe/

réels compatibles avec notre approche. En effet, CamVid ne propose pas de séquences vidéo génériques mais très spécifiques, elle ne contient que des séquences vidéo prises depuis une voiture. TRECVID ne propose pas de vérité-terrain orientée-objet. LabelMe video contient des objets-vidéo annotés par des utilisateurs mais dont la qualité est variable. Elle est surtout dédiée à l'annotation de contenu vidéo. Il n'existe par contre aucune base contenant les objets-vidéo de référence représentant les objets-réels présents dans la vidéo et des regroupements de référence de ces objets-vidéo.

Nous avons donc créé, comme pour la segmentation, notre propre base de référence. Nous avons utilisé des séquences vidéo issues de journaux télévisés français, du projet OpenVideo [8] et du projet CAVIAR [9]. L'utilisation de séquences vidéo diverses provenant de sources diverses a pour but de simuler une base de vidéo générique. De ces vidéos, nous avons extrait, en utilisant VOX, 12 objets-vidéo que nous représentons dans la figure 3.10. Les objets-vidéo obtenus sont de taille variable, leur durée est en moyenne de 30 trames.

FIGURE 3.10 – Notre base d'objets-vidéo.

3.6.2 Résultats

Dans cette section, nous étudions les résultats obtenus par notre approche selon deux besoins différents d'un utilisateur. Le premier cas est celui d'un utilisateur désirant deux classes, une contenant les êtres humains et une autre les navettes spatiales. Le deuxième cas est celui d'un utilisateur désirant trois classes, une contenant les navettes spatiales et les deux autres contenant d'un coté les présentateurs et de l'autre les personnes qui marchent.

En utilisant l'approche non supervisée, on ne peut satisfaire aucune des deux requêtes (voir la figure 3.11 pour le dendrogramme représentant la classification hiérarchique non supervisée de notre base d'objets-vidéo). En effet, si on choisit le niveau de partition donnant deux classes, on obtiendra une classe contenant les présentateurs et une autre contenant les navettes spatiales et les gens qui marchent. Cette classification, qui ne satisfait absolument pas un être humain, est due à la différence entre notre perception du concept « être humain » et les descripteurs bas-niveau que nous utilisons. En effet, du point de vue de nos descripteurs de forme, de couleur et de texture, les objets-vidéo de type *personne qui marche* sont beaucoup plus proches des objets-vidéo *navette spatiale* que des objets-vidéo *présentateur* (cf. figure 3.12). Il s'agit d'un effet du fossé sémantique. Par contre, en descendant d'un niveau dans la hiérarchie des partitions pour obtenir trois classes, il serait souhaitable que la classe contenant les objets-vidéo *navette spatiale* et *personne qui marche* se divise en deux classes contenant pour l'une les objets-vidéo *navette spatiale* et pour l'autre les objets-vidéo *personne qui marche*. Cependant, nous obtenons une classe contenant les objets-vidéo *présentateur*, une autre contenant les objets-vidéo *personne qui marche* plus un objet-vidéo *navette spatiale* et une dernière classe contenant les objets-vidéo *navette spatiale* restants. A l'instar du problème évoqué précédemment pour la partition en deux classes, il s'agit ici encore d'un problème de fossé sémantique. La figure 3.12 illustre l'objet-vidéo *navette spatiale* mal classé ainsi que les distances entre des exemples d'objets-vidéo de chaque classe. On constate que, même si pour un

8. OpenVideo, http ://www.open-video.org/
9. CAVIAR, http ://homepages.inf.ed.ac.uk/rbf/CAVIAR/

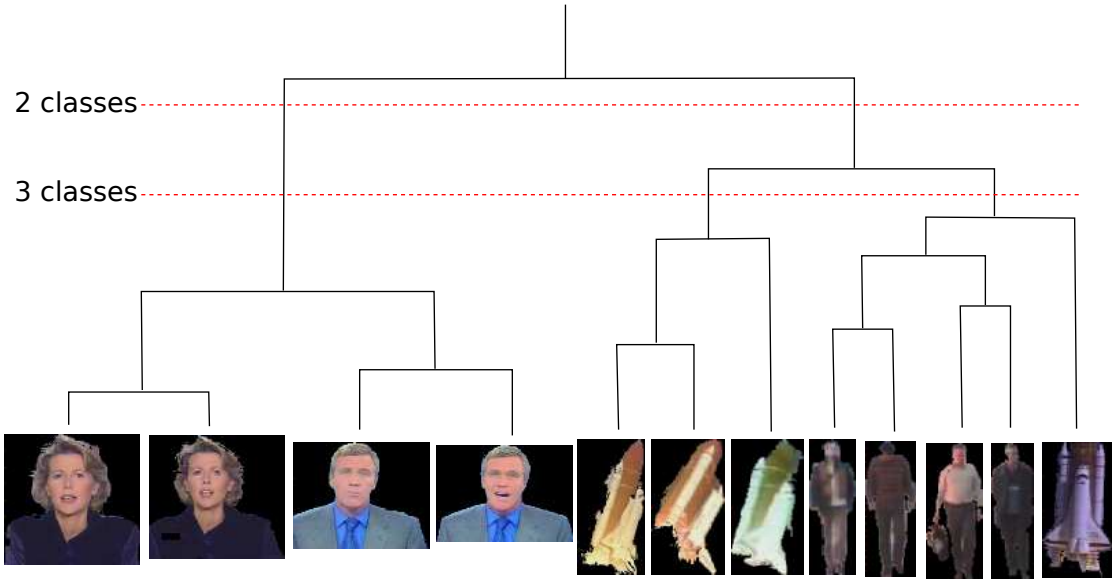


FIGURE 3.11 – Classification hiérarchique ascendante non supervisée de notre base d'objets-vidéo.

utilisateur les deux objets-vidéo *navette spatiale* représentent le même concept, les descripteurs bas-niveau basés sur la forme, la couleur et la texture seront plus proches (en termes de distance) entre l'objet-vidéo *navette spatiale* mal classé et l'exemple de *personne qui marche*. Ainsi, même avec des descripteurs calculés sur des objets-vidéo, des regroupements qui paraissent triviaux pour un être humain ne sont pas réalisables de façon non-supervisée. Il est cependant évident que nous nous appuyons sur quelques descripteurs relativement simples. Utiliser un plus grand nombre de descripteurs et/ou des descripteurs plus complexes et apportant une sémantique plus importante permettraient peut-être d'améliorer les résultats. De même, nous pourrions utiliser une méthode de clustering plus évoluée qui donnerait sans nul doute de meilleurs résultats. Cependant, plutôt que d'utiliser des descripteurs et des méthodes de regroupement plus complexes, nous préférons, à l'instar de ce que nous avons fait pour la segmentation guidée (cf chapitre 2), nous appuyer sur une intégration de l'utilisation dans le processus de regroupement.

L'introduction de contraintes *Must-Link* et *Cannot-Link* par l'utilisateur permet de résoudre les problèmes dus au fossé sémantique. Si nous reprenons le problème posé par l'objet-vidéo *navette spatiale* mal classé, il suffit de poser une contrainte *Must-Link* entre cet objet et un des autres objets-vidéo *navette spatiale*. Cette simple contrainte rattache l'objet-vidéo problématique aux autres objets-vidéo *navette spatiale* et permet d'obtenir les trois classes recherchées par l'utilisateur (cf figure 3.13). Concernant le problème posé par le regroupement de tous les êtres humains dans une classe et des navettes spatiales dans une autre, il faut introduire deux contraintes : une *Must-Link* et une *Cannot-Link* (cf figure 3.14). La première est la même que dans le cas précédent, nous rattachons de force l'objet-vidéo *navette spatiale* mal classé à un des autres objets-vidéo *navette spatiale*. La contrainte *Cannot-Link* est posée entre l'objet-vidéo *navette spatiale* mal classé et un des objets-vidéo *personne qui marche* afin de séparer les objets-vidéo *navette spatiale* des objets-vidéo *personne qui marche*.

L'introduction par l'utilisateur de contraintes simples a permis de combler en partie le fossé sémantique entre sa perception des objets et leur description bas-niveau.

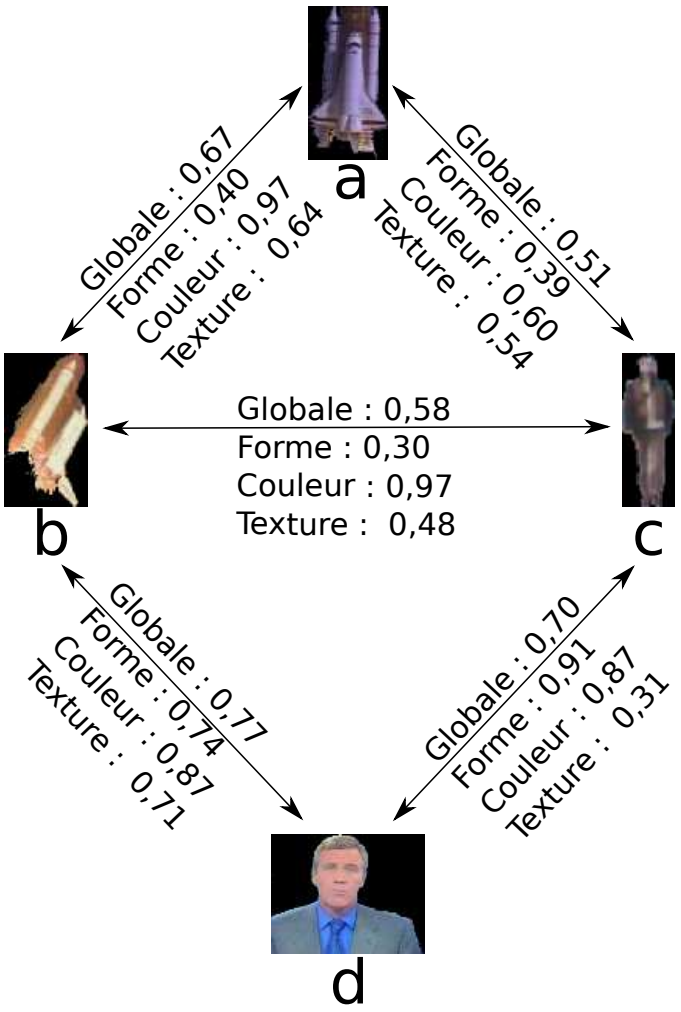


FIGURE 3.12 – Distances entre objets-vidéo : a) Objet-vidéo *navette spatiale* mal classé perturbant la classification en trois classes : *présentateur*, *personne qui marche* et *navette spatiale*, b) exemple de *navette spatiale*, c) exemple de *personne qui marche*, d) exemple de *présentateur*.

3.7 Conclusion

Dans ce chapitre, nous avons introduit une nouvelle taxonomie pour caractériser les SFV et l'avons utilisée pour étudier et comparer les SFV actuels. Les systèmes de fouille vidéo (SFV) récents s'appuient sur une description des séquences vidéo réalisée à l'échelle des objets ou des régions, mais sont appliqués sur des éléments tels que les plans ou les séquences vidéo intégrales. Nous avons discuté les répercussions du choix de l'objet comme élément sur les autres caractéristiques définies dans notre taxonomie. L'importance de la segmentation a été soulignée, et nous avons suggéré des moyens d'introduire des informations de nature sémantique dans les SFV orienté-objet. Puis, nous avons proposé VOMF, un cadre générique pour la fouille vidéo orientée-objet. VOMF offre de nouvelles perspectives, la fouille vidéo étant plus pertinente si les objets considérés sont les objets réels (du point de vue de l'utilisateur) présents dans les séquences vidéo. Ce cadre a été appliqué au contexte du clustering d'objets-vidéo donnant naissance au logiciel VOX. Nous avons utilisé pour cela des descripteurs bas-niveau ainsi qu'une méthode de clustering simple. Cette méthode et ces descripteurs n'ont pu combler le fossé sémantique ni s'adapter aux besoins précis des utilisateurs. Nous avons donc impliqué l'utilisateur dans le processus conformément à ce que nous préconisions dans la présentation de VOMF. Cette implication via l'imposition de contraintes entre certains couples d'objets a permis de guider efficacement la classification des objets-vidéo en accord avec

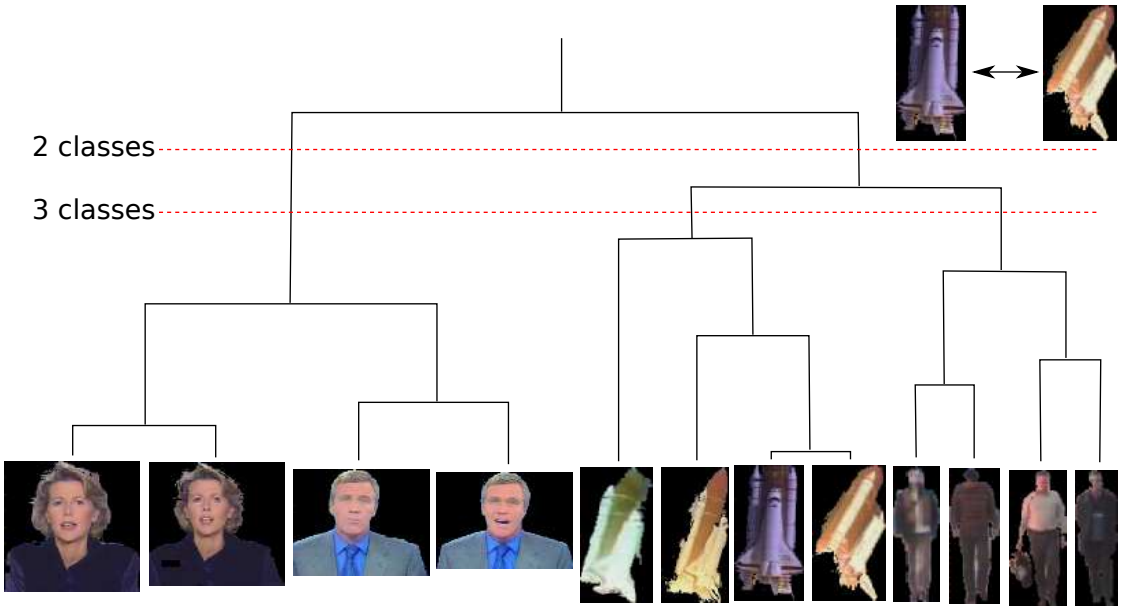


FIGURE 3.13 – Classification hiérarchique ascendante avec une contrainte *Must-Link* de notre base d'objets-vidéo.

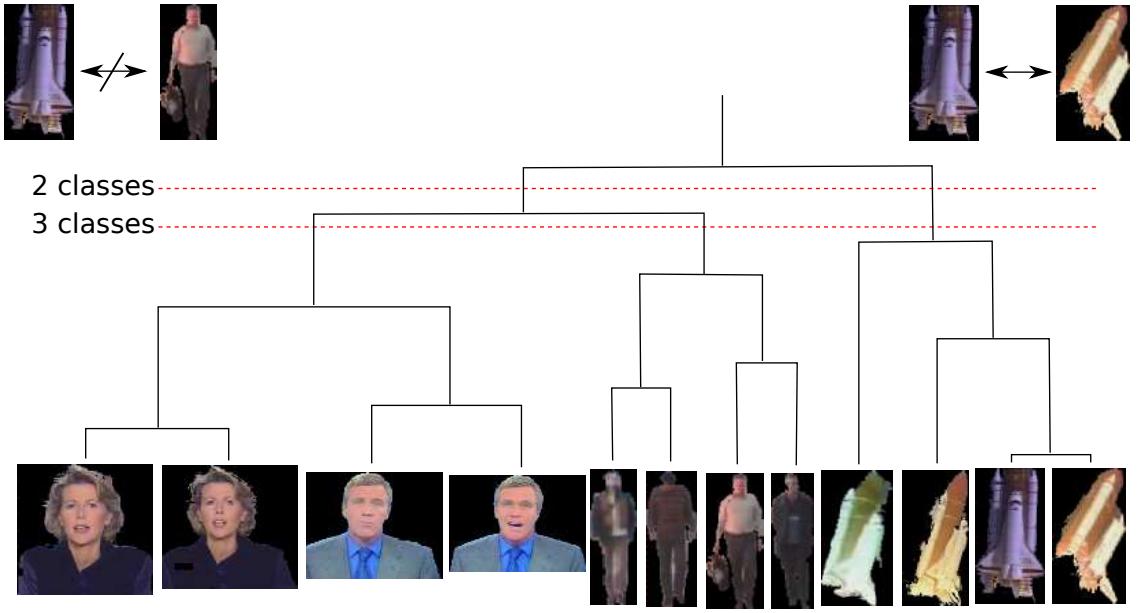


FIGURE 3.14 – Classification hiérarchique ascendante avec une contrainte *Must-Link* et une contrainte *Cannot-Link* de notre base d'objets-vidéo.

les besoins de l'utilisateur.

Les résultats préliminaires obtenus par notre système montrent que le framework VOMF peut être effectivement une solution. Cependant, VOX nécessiterait d'être appliqué sur une base de séquences vidéo plus étendue afin de valider plus profondément son efficacité pour résoudre le problème de la segmentation et la classification semi-supervisée d'objets-vidéo. De plus, le système implémente actuellement un nombre restreint de descripteurs. Il en va de même pour les algorithmes de clustering. Le clustering hiérarchique ascendant nous permet de construire facilement une hiérarchie des partitions dans laquelle il est possible de naviguer pour sélectionner le niveau

de granularité souhaité dans la partition. Cependant, cet algorithme, bien que simple, possède un coût calculatoire ($O(n^2)$) très important de même qu'une consommation mémoire importante dûe à la nécessité de stocker toutes les partitions intermédiaires de la hiérarchie. Il serait donc préférable d'utiliser d'autres méthodes hiérarchiques comme BIRCH [ZRL96], qui pose des conditions sur le nombre maximum de fils qu'un nœud peut avoir ainsi que sur le diamètre maximal d'une classe, ou CURE [GRS00] qui calcule les distances entre classes sur un nombre fixe de représentants de la classe au lieu de calculer les distances entre tous les membres des deux classes, réduisant ainsi de façon importante le coût calculatoire. Nous pourrions également inclure plus profondément les contraintes dans le système : en effet elle n'agissent pour l'instant que sur la matrice des distances entre les différents objets-vidéo et non directement sur l'algorithme de partitionnement. Ces contraintes pourraient être utilisées pour modifier la mesure de distance (par exemple en pondérant les descripteurs). De plus, d'une façon analogue à ce que nous envisageons pour la co-segmentation, nous pourrions à partir de contraintes données par l'utilisateur en inférer de nouvelles afin d'influencer plus profondément le processus de fouille.

Conclusion et perspectives

Dans cette thèse, nous avons abordé le problème de la structuration de bases de séquences vidéo. Plus précisément, nous cherchions à proposer un système permettant de faciliter l'utilisation de bases de séquences vidéo, souvent peu structurées. Nous avons pointé deux verrous pour la résolution de ce problème : le *fossé sémantique* et le *temps de calcul* nécessaire au traitement des séquences vidéo.

Une solution au problème posé par le fossé sémantique est contenue dans la proposition d'un cadre générique pour la fouille de données vidéo. Ce cadre implique l'utilisation d'*objets-vidéo*, élément portant une information sémantique mais dont l'extraction n'est pas triviale. Nous avons donc, dans le cadre proposé, décidé d'impliquer l'utilisateur dans l'étape d'extraction de ces objets. Une séquence vidéo représentant un volume important de données, son traitement est coûteux et nécessite donc a priori un temps de calcul important. Afin de minimiser ce temps de calcul, nous effectuons hors-ligne une pré-segmentation des séquences vidéo. Cette pré-segmentation repose sur une extension des zones quasi-plates aux séquences vidéo. L'importante réduction des données qu'elle induit permet d'effectuer une segmentation des objets-vidéo guidée par l'utilisateur pour un coût calculatoire très modeste, ce qui autorise une interactivité réelle. Cette efficacité calculatoire repose, outre la pré-segmentation, sur l'utilisation de structures efficaces et de descripteurs simples dont la combinaison et la comparaison présentent une complexité algorithmique faible. Une fois obtenus les objets-vidéo d'intérêt de la base de séquences vidéo, le cadre que nous proposons prévoit leur description sémantique. Nous avons ici aussi utilisé des descriptions simples et compactes afin d'être capable de gérer du point de vue mémoire un nombre possiblement important d'objets-vidéo et de permettre leur comparaison pour un coût calculatoire restreint. Pour la fouille de ces objets-vidéo, nous n'avons considéré que la tâche de regroupement (clustering), celle-ci étant préliminaire à plusieurs autres tâches de fouille. Le regroupement des objets-vidéo s'est également heurté au problème du *fossé sémantique*. Ici, il s'agissait de la différence entre la proximité des objets-vidéo du point de vue de leurs descriptions et la proximité des objets réels qu'ils représentent, du point de vue de l'utilisateur et de son objectif. Un traitement purement automatique ne donnant pas les résultats escomptés, nous avons egalement impliqué l'utilisateur dans le processus en lui demandant d'indiquer quelques contraintes entre objets-vidéo. Ces contraintes permettent alors de guider le processus de regroupement.

Nous avons ainsi proposé un cadre général permettant de lever en partie les deux verrous qui empêchaient la résolution du problème de la structuration de bases de séquences vidéo et ce en impliquant l'utilisateur et en proposant une méthode peu coûteuse d'un point de vue calculatoire.

Contributions

Dans le cadre des recherches présentées dans ce document, nous avons apporté plusieurs contributions que nous détaillons ici.

Dans le chapitre 1, nous avons traité des zones quasi-plates. Nous avons étudié les extensions couleurs existantes et nous avons déterminé quelles étaient les extensions les plus performantes de l'α-$\mathcal{Z}$ et de la la $(P_{\alpha_{Zanoguera}}, P_{\Omega_{Soille}})$-$\mathcal{Z}$. Les expériences menées sur la base de Berkeley ont

montré que le choix de l'espace couleur et de la métrique n'avait pas une grande influence (pour les espaces, métriques et méthodes testés) sur nos mesures d'évaluation. Nous avons ensuite proposé un nouveau schéma générique et incrémental pour la construction des zones quasi-plates. Ce schéma consiste à considérer la construction des zones quasi-plates comme une suite d'opérateurs et non plus comme un opérateur unique. Il permet d'appliquer différents critères de façon séquentielle et non plus conjointe. Nous avons appliqué ce schéma pour la construction de zones quasi-plates vidéo, en traitant séquentiellement les dimensions spatiales et temporelle, et obtenu des résultats nettement supérieurs à ceux obtenus par un traitement conjoint de ces dimensions. Les zones quasi-plates produisant une importante sur-segmentation, nous avons étudié les méthodes de filtrage existantes et proposé une nouvelle méthode de filtrage obtenant de meilleurs résultats. Nous avons ensuite étendu cette méthode de filtrage aux zones quasi-plates vidéo, et obtenu une réduction importante de la sur-segmentation tout en conservant une bonne qualité des zones quasi-plates dans une optique de fusion pour obtenir des objets-vidéo. Le traitement de vidéo étant coûteux, nous avons proposé quelques moyens de diminuer le temps de calcul de la production de zones quasi-plates. Nous avons proposé un algorithme permettant la construction de $(P_1, \ldots, P_n)$-$\mathcal{Z}$ séparant prédicats locaux et globaux afin de diminuer le coût calculatoire. Nous avons alors expliqué les apports des structures efficaces de type graphe, évalué le gain en temps de calcul qu'apporte l'utilisation de tables de correspondance dans le calcul des distances. Enfin, nous avons proposé, pour le cas d'applications temps-réel, une solution sacrifiant un peu la qualité des zones quasi-plates au bénéfice d'une grande réduction du temps de calcul.

Dans le chapitre 2 nous avons traité de la segmentation interactive basée sur une segmentation initiale en zones quasi-plates. Nous avons d'abord proposé une méthode permettant à l'utilisateur de guider par le dessin de marqueurs la fusion de zones quasi-plates afin d'obtenir les objets-vidéo qu'il désire. Les zones quasi-plates initiales pouvant être produites hors-ligne, cette méthode présente l'avantage d'être très rapide puisque la segmentation en objets-vidéo est obtenue, à partir de marqueurs saisis par l'utilisateur, par une étape peu coûteuse de fusion de zones quasi-plates basée sur ces marqueurs. Nous avons ensuite rendu cette méthode interactive, afin de pouvoir affiner la segmentation par correction des marqueurs, et inclus un mécanisme de correction des zones quasi-plates basé sur les marqueurs de l'utilisateur. Pointant le défaut principal de notre approche, la nécessité de devoir définir des marqueurs pour chaque séquence vidéo, nous avons abordé la problématique de la co-segmentation.

Dans le chapitre 3 nous avons traité de la fouille vidéo. Nous avons proposé une nouvelle taxonomie des systèmes de fouille vidéo. Nous avons ensuite étudié ce qu'impliquait l'utilisation de l'objet-vidéo comme élément d'un processus de fouille, nous en avons déduit qu'il était nécessaire d'impliquer l'utilisateur dans un tel processus. Ce constat a amené la proposition du cadre générique pour la fouille vidéo orientée-objet VOMF. Nous avons présenté des résultats préliminaires encourageants obtenus en appliquant VOMF au problème du regroupement d'objets-vidéo et en impliquant l'utilisateur dans le processus de fouille par le biais de la définition de contraintes.

Ces différentes contributions ont été validées par des publications de niveau international [WLG10, WLG11d, WLG11b] et national [WLG11a, WLG11e, WLG11c]. D'autres publications sont actuellement en soummission ou en préparation.

Cette thèse a, en outre, donné lieu au développement de deux logiciels : Vox et Odessa. Vox est l'implantation du cadre VOMF appliqué au regroupement d'objets-vidéo (cf. annexe B). Odessa permet la création de base de segmentations vidéo de référence et propose des outils pour l'évaluation de segmentation vidéo (cf. annexe C).

Perspectives scientifiques

Nous avons introduit un cadre générique pour la construction incrémentale des zones quasi-plates. Ce cadre permet l'application séquentielle de différents critères. Pour l'instant, nous ne

l'avons utilisé que pour traiter séquentiellement les dimensions spatiales et temporelle : ainsi la seule différence entre les deux traitements successifs était le voisinage pris en compte, les prédicats et leurs paramètres étant identiques. Il serait nécessaire, outre la variation de voisinage, d'utiliser également des prédicats et/ou des paramètres de prédicats différents pour le traitement des dimensions spatiales et temporelle. Il y aurait également des améliorations à apporter à l'algorithme $(P_1, \ldots, P_n)$-$\mathcal{Z}$ pour aller au delà du simple traitement différencié des prédicats locaux et globaux. Enfin, l'inconvénient principal des zones quasi-plates est la nécessité de devoir régler des paramètres pas forcément intuitifs (α, ω, etc.). Une méthode analysant l'image ou la séquence vidéo afin de fixer un seuil « idéal » permettrait de remédier à cet inconvénient. Les zones quasi-plates représentent un niveau, déterminé par les paramètres, dans une hierarchie de partitions (cf. équation 1.11). Dès lors, régler les paramètres revient à déterminer un niveau « idéal » (au sens d'un certain critère) dans la hiérarchie. Des travaux [Meu05] ont été menés sur ce problème et pourrait constituer une piste pour le réglage de paramètres des zones quasi-plates.

Nous avons pointé, dans ce manuscrit, la nécessité pour l'utilisateur de guider la segmentation sur chaque séquence vidéo. Cette tâche peut s'avérer fastidieuse sur une base de taille importante et ce même si la segmentation interactive est peu coûteuse d'un point de vue calculatoire. Nous avons indiqué qu'une solution possible serait le développement d'une méthode de co-segmentation de séquences vidéo, tout en précisant que ce domaine de recherche était encore très récent. Il est pour l'instant limité à la segmentation d'objets identiques sous différents points de vue mais évoluera dans l'avenir vers la segmentation d'objets de plus en plus hétérogènes.

Nous avons également indiqué que les contraintes utilisées actuellement pour guider le processus de fouille n'agissaient pour le moment que sur les distances entre deux objets-vidéo. Cela permet d'influencer le système de fouille mais nous pourrions l'influencer plus fortement en utilisant ces contraintes plus profondément dans le système. Il serait envisageable d'utiliser les contraintes pour déterminer une pondération entre les différents descripteurs. Cette pondération adapterait la distance globale (prenant en compte tous les descripteurs selon leurs différentes pondérations) au contexte particulier d'un base de séquences vidéo. Une autre voie d'amélioration serait, à l'instar de la co-segmentation évoquée précédemment, de développer des « co-contraintes », c'est-à-dire d'inférer de nouvelles contraintes à partir d'un ensemble de contraintes données par l'utilisateur. Ceci pourrait permettre de mieux guider le processus de fouille sans obliger l'utilisateur à définir un grand nombre de contraintes. Enfin, une dernière perspective concerne la définition des contraintes en elle-même, puisque pour l'instant nous n'utilisons que deux méthodes pour choisir le couple d'objets-vidéo que l'on contraint : une méthode aléatoire et une méthode où l'utilisateur choisit les deux objets-vidéo à lier par une contrainte. La première est peu efficace, la deuxième est peu réaliste dans le contexte d'une base contenant un grand nombre d'objets-vidéo. Dès lors, il serait intéressant de disposer de méthodes permettant de proposer de manière intelligente des couples d'objets-vidéo sur lesquels poser des contraintes de façon à guider efficacement le processus de fouille.

D'un point de vue applicatif, il serait intéressant d'appliquer les méthodes proposées dans ce manuscrit à d'autres domaines. Nous pourrions par exemple, tester notre approche de segmentation interactive sur des images tri-dimensionnelles (par exemple les images médicales). Des travaux récents ont par ailleurs appliqué la Morphologie Mathématique hors du domaine visuel, par exemple sur le domaine des données textuelles [LC11]. L'étude de l'application des zones quasi-plates dans ce contexte serait intéressante.

Perspectives industrielles

Nous pensons également que le faible coût calculatoire de notre approche permet son application dans le contexte de l'*informatique en nuage* (cloud computing) et des terminaux mobiles. L'*informatique en nuage* consiste à déporter les traitements et le stockage des données sur des serveurs distants au lieu de la machine physique de l'utilisateur. Les intérêts sont principalement de pouvoir y accéder de n'importe quelle machine connectée à Internet, de pouvoir facilement collabo-

rer en utilisant des données communes et de lancer des calculs lourds depuis un terminal mobile à faible capacité calculatoire et mémoire (par exemple les smartphones, tablettes tactiles, netbooks, etc.). Dans le cadre de VOMF, il s'agirait de stocker les séquences vidéo, leurs segmentations en ZQP, leurs objets-vidéo, les descriptions de ces objets et les résultats d'opération de fouille sur les serveurs du nuage afin qu'ils soient accessibles à une multitude d'utilisateurs. Les traitements les plus lourds : segmentation en zones quasi-plates, segmentation guidée, description et fouille seraient effectués sur les serveurs distants également. Seules les interactions avec l'utilisateur seraient effectuées sur les terminaux mobiles. Notons que les interfaces des terminaux tactiles (smartphones, tablettes tactiles) se prêtent particulièrement au dessin des marqueurs nécessaires à notre approche de segmentation guidée. Ce schéma permet d'effectuer tous les traitements hors-ligne (segmentation en zones quasi-plates et description des objets-vidéo) sur les serveurs distants et donc sans monopoliser la machine de l'utilisateur. La limitation du rôle de la machine de l'utilisateur à celui de terminal est permise par les capacités actuelles des réseaux de communication (capables notamment de diffuser une séquence vidéo en temps-réel). Le faible coût calculatoire de notre approche (surtout vrai pour la segmentation guidée, plus discutable pour l'étape de fouille), permet d'avoir une interaction réelle même si les calculs sont effectués à distance. De plus, pour la segmentation guidée l'on pourrait même envisager de l'effectuer sur les terminaux mobiles malgré leur faible capacité. Un tel système permettrait d'effectuer des tâches de fouille vidéo n'importe où et serait intéressant dans des contextes applicatifs nécessitant l'utilisation de la vidéo hors de bureau. En outre, ce système serait particulièrement intéressant dans un contexte multi-utilisateurs. Il permet en effet, d'utiliser des objets-vidéo extraits par d'autres personnes ainsi que de partager des résultats de fouille vidéo. La figure 3.15 illustre l'application de VOMF dans le contexte de l'informatique en nuage.

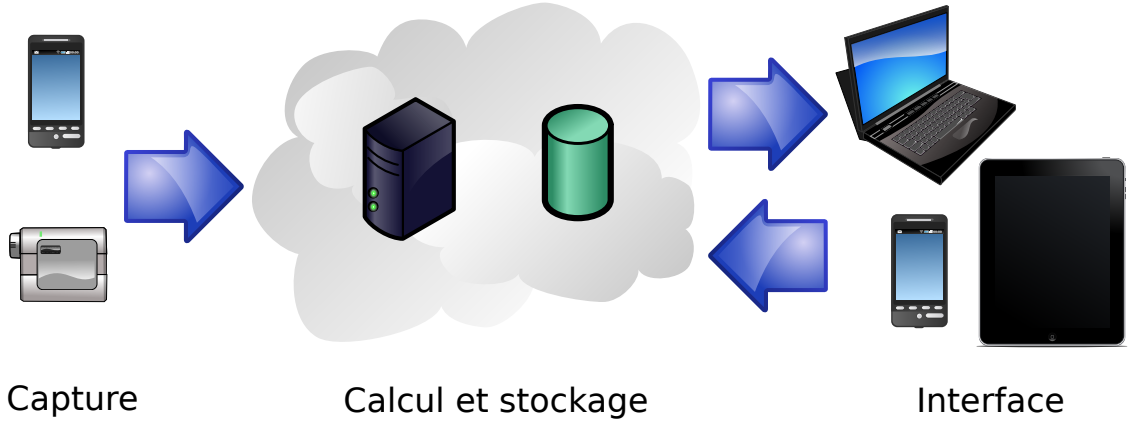


FIGURE 3.15 – Organisation via le nuage d'un système implémentant le cadre VOMF.

Publications

Publications liées à la thèse

Article en soumission

1. Jonathan Weber, Sébastien Lefèvre, Pierre Gançarski, Toward object-oriented video mining, *Information Systems*.

Communications à des manifestations internationales à comité de lecture

1. Jonathan Weber, Sébastien Lefèvre, Pierre Gançarski, Interactive Video Segmentation based on Quasi-Flat Zones, *IEEE International Symposium on Image and Signal Processing and Analysis (ISPA)*, Dubrovnik, Croatie, Septembre 2011.
2. Jonathan Weber, Sébastien Lefèvre, Pierre Gançarski, Spatio-temporal quasi-flat zones for morphological video segmentation, *International Symposium on Mathematical Morphology (ISMM)*, Intra, Italie, Juillet 2011, Springer-Verlag Lecture Notes in Computer Sciences, Volume 6671, pages 178-189.
3. Jonathan Weber, Sébastien Lefèvre, Pierre Gançarski, Video Object Mining : Issues and Perspectives, *IEEE International Conference on Semantic Computing (ICSC)*, Pittsburgh, USA, Septembre 2010, pages 85-90.

Communications à des manifestations nationales à comité de lecture

1. Jonathan Weber, Sébastien Lefèvre, Pierre Gançarski, Segmentation vidéo interactive par zones quasi-plates, *Colloque GRETSI sur le Traitement du Signal et des Images*, Bordeaux, Septembre 2011.
2. Jonathan Weber, Sébastien Lefèvre, Pierre Gançarski, Zones quasi-plates spatio-temporelles et segmentation morphologique de séquences vidéo, *Congrès francophone ORASIS de vision par ordinateur*, Praz-sur-arly, France, Juin 2011.
3. Jonathan Weber, Sébastien Lefèvre, Pierre Gançarski, Fouille vidéo orientée objet, une approche générique, *Atelier Fouille de données complexes, Journées Francophones Extraction et Gestion des Connaissances (EGC 2011)*, Brest, France, pages 9-20, Janvier 2011.

Hors thèse

Article en soumission

1. Jonathan Weber, Sébastien Lefèvre, Spatial & Spectral Morphological Template Matching, *Pattern Recognition*.

Communications à des manifestations internationales à comité de lecture

1. Jonathan Weber, Sébastien Lefèvre, A multivariate Hit-or-Miss transform for conjoint spatial and spectral template matching, *IEEE International Conference on Image and Signal Processing*, Cherbourg, Juillet 2008, Springer-Verlag Lecture Notes in Computer Sciences, Volume 5099, pages 226-235.
2. Anne Puissant, Jonathan Weber, Sébastien Lefèvre, Coastline extraction in VHR imagery using mathematical morphology with spatial and spectral knowledge, *ISPRS Congress*, Beijing, Juillet 2008.
3. Sébastien Lefèvre, Jonathan Weber, David Sheeren, Automatic building extraction in VHR images using advanced morphological operators, *IEEE/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas (URBAN)*, Paris, Avril 2007.

Articles dans des revues nationales à comité de lecture

1. David Sheeren, Sébastien Lefèvre, Jonathan Weber, La morphologie mathématique binaire pour l'extraction automatique des bâtiments dans les images THRS, *Revue Internationale de Géomatique*, numéro thématique "SAGEO 2006", 2007, pages 333-352.

Communications à des manifestations nationales à comité de lecture

1. Jonathan Weber, Sébastien Lefèvre, David Sheeren, Détection des bâtiments dans les images THRS avec la morphologie mathématique, *Colloque International de Géomatique et d'Analyse Spatiale (SAGEO)*, Strasbourg, Septembre 2006.

Divers

1. Jonathan Weber, Sébastien Lefèvre, David Sheeren, Extraction automatique de bâtiments dans des images satellites THRS en utilisant des opérateurs de Morphologie Mathématique, *Ateliers PNTS*, Nantes, Septembre 2007.

Table des figures

Liste des tableaux

Liste des Algorithmes

Annexe A

Pelican

Le Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection (LSIIT) de l'Université de Strasbourg développe depuis 2004 une plate-forme permettant la réalisation et l'exécution d'outils d'analyse et de traitement des images, sous l'impulsion de Sébastien Lefèvre [Lef09]. Cette plate-forme est implémentée en Java et se nomme Pelican (Polyvalent Extensible Library for Image Computing and ANalysis). Elle est utilisée dans le cadre des travaux de recherche en traitements d'image et permet la conception, l'expérimentation et l'utilisation de méthodes de traitements d'images dans un environnement générique. Le logo du projet est présenté dans la figure A.1.

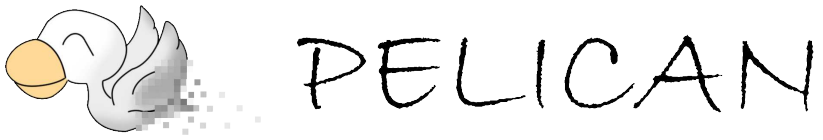


Figure A.1 – Logo du projet Pelican.

L'objectif de Pelican est de fournir un environnement de traitements d'images génériques multi-plateforme capable de traiter tous les types d'images que ce soit d'un point de vue géométrique (images 1D, 2D, 3D, séquences vidéo 2D+t ou 3D+t), spectral (binaire, niveaux de gris, couleur ou multibandes) ou numérique (valeurs de pixels binaires, entières ou décimales). Cette généricité des données a permis son utilisation dans des domaines d'applications variés (imagerie médicale, imagerie satellite, imagerie astronomique, recherche par le contenu, indexation vidéo, etc). Elle s'adresse donc à un panel d'utilisateur très varié et est constamment enrichie par les contributions des chercheurs, doctorants et stagiaires constituant son équipe de développement. Pelican comprend actuellement 1093 classes représentant plus de 100 000 lignes de code.

Du point de vue de son architecture, Pelican est construit de façon modulaire et présente trois couches (cf. figure A.2) : le noyau, les algorithmes et les interfaces. Le noyau contient les structures de données, les spécifications que doit respecter un algorithme et les fonctions utilitaires communes. Il assure également la liaison entre les algorithmes et les interfaces. Les algorithmes représentent la partie traitement de la plate-forme, ils sont composés d'algorithmes de recherche développés par l'équipe de développement ou d'algorithmes existants dans la littérature. Notons que l'écriture d'algorithme est facilitée par les spécifications présentes dans le noyau qui de plus assure une compatibilité entre les différents algorithmes. Les interfaces permettent l'accès à la plate-forme et à ses algorithmes. Chaque interface est adaptée à un profil d'utilisateur particulier. En outre, il est possible d'utiliser Pelican comme bibliothèque de fonctions ou produit tiers pour des projets ayant besoin d'outils de traitements d'images.

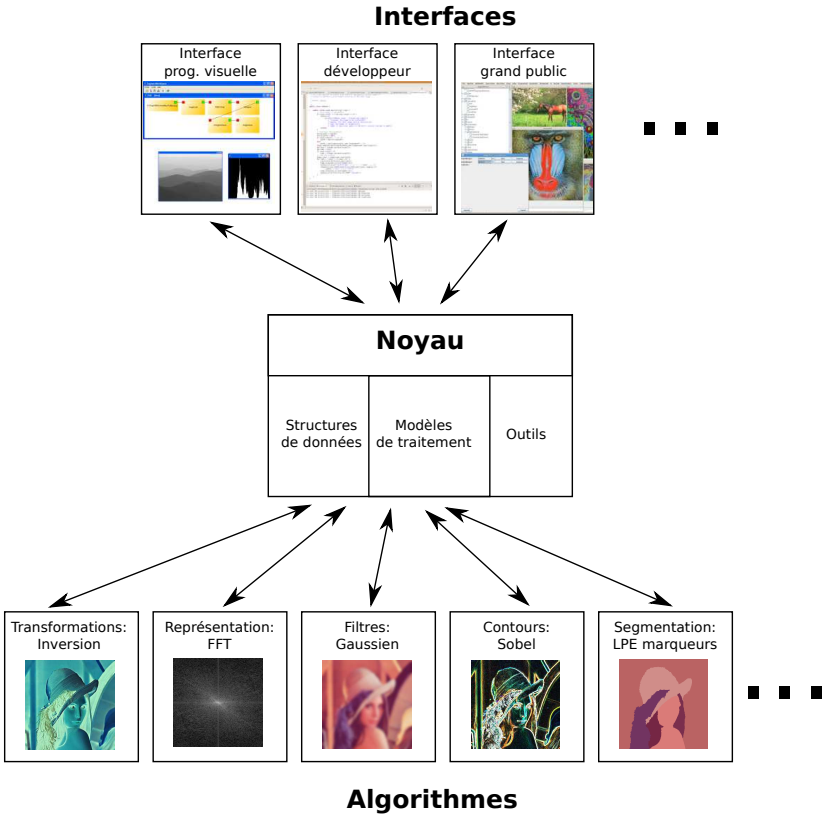


FIGURE A.2 – Architecture du projet PELICAN (figure issue de [Lef09]).

Dans cette thèse, nos apports et travaux en rapport avec PELICAN ont principalement été l'implantation d'algorithmes de traitements d'images existants ainsi que l'implantation des algorithmes et méthodes proposées dans ce manuscrit. Ainsi que la réalisation, dans le cadre du stage de Vincent Danner, de classes permettant la gestion d'images (au sens large) de grande taille, c'est-à-dire ne tenant pas en mémoire.

Annexe B

Vox

L'implantation du framework VOMF dans le contexte du clustering d'objets-vidéo a été réalisée au sein du logiciel Vox (Video Object Segmentation and Clustering System). Ce logiciel est écrit en langage Java et utilise la plate-forme de traitement d'images Pelican (cf. annexe A). Son implantation repose sur le modèle MVC (Modèle-Vue-Contrôleur) qui sépare données, traitements et interface. Vox comprend actuellement 50 classes représentant plus de 5000 lignes de code (la majorité des traitements sur les séquences vidéo sont effectués par des classes de Pelican). Nous prévoyons de mettre Vox à la disposition de la communauté courant 2011.

L'objectif de Vox est de fournir un environnement complet permettant le test des algorithmes et méthodes que nous développons pour l'utilisation du cadre VOMF au contexte du regroupements d'objets-vidéo. Sa partie traitement se compose de quatre briques modulaires (cf. figure B.1). Deux de ses briques sont dédiées à des traitements hors-ligne (*pré-segmentation* et *description*) tandis que les deux autres (*segmentation interactive* et *clustering interactif*) nécessitent l'intervention de l'utilisateur. Vox est conçu pour être modulaire, c'est-à-dire que derrière ces briques il n'y a pas un algorithme particulier. En effet, n'importe quel algorithme dédié à la tâche d'une brique pourrait être implémenté au sein de cette brique. Actuellement, les méthodes suivantes sont implémentées dans les différentes briques :

- Pré-traitement : Zones quasi-plates spatio-temporelles et gradient euclidien spatio-temporel ;
- Segmentation interactive : Zones quasi-plates interactives guidées par marqueurs, ligne de partage des eaux guidée par marqueurs et la ligne de partage des eaux par propagation de marqueurs ;
- Description : Histogramme de structure de couleur, covariance morphologique et descripteurs génériques de Fourier ;
- Classification : Clustering hiérarchique ascendant et clustering hiérarchique ascendant contraint.

D'autres méthodes seront ajoutées aux différentes briques, permettant à l'utilisateur d'avoir plus de choix pour chaque étape de Vox.

Il est à noter que la méthode de pré-traitement dépend de la méthode de segmentation choisie. En effet, un gradient morphologique est le pré-traitement des méthodes basées sur une ligne de partage des eaux tandis que les zones quasi-plates spatio-temporelles correspondent au pré-traitement de la méthode interactive de zones quasi-plates guidées par marqueurs.

Vox permet donc de lancer hors-ligne les traitements les plus coûteux en temps de calcul tels que la pré-segmentation et la description des objets-vidéo. Pour cela, l'utilisateur sélectionne les séquences vidéo à pré-segmenter (cf. figure B.2) ou les descripteurs à calculer. Vox lance alors les calculs et sauvegarde automatiquement les résultats, il affiche une barre de progression qui indique à l'utilisateur l'avancement du processus.

Concernant la segmentation interactive, l'interface est identique à celle présentée pour les zones quasi-plates interactives guidées par marqueurs (cf figure 2.8). La seule différence est que l'on demande à l'utilisateur, lorsqu'il ajoute un type de marqueurs, s'il s'agit de marqueurs de fond ou d'objets-vidéo. Cette information sera utilisée lors de l'étape de clustering où l'on ne classifiera que

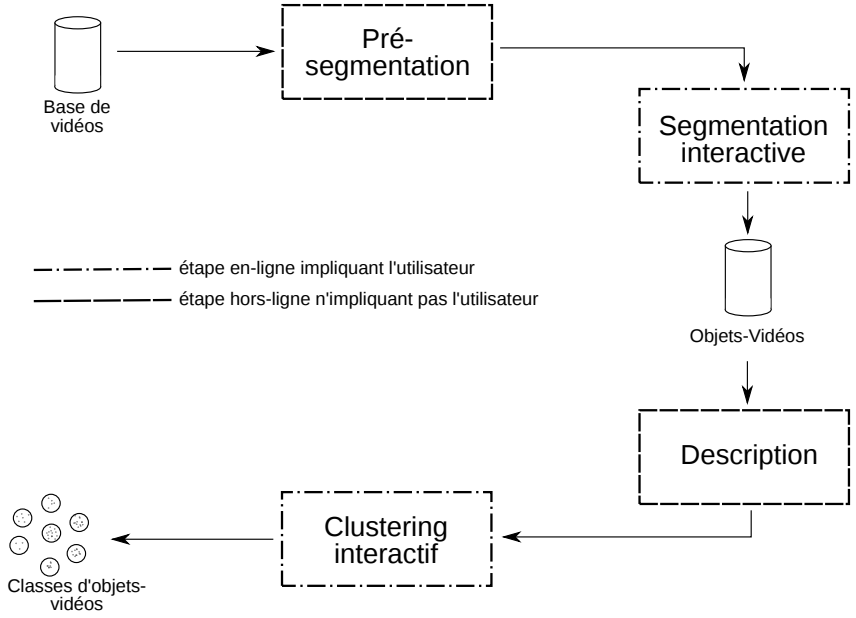


FIGURE B.1 – Flux de traitement des données de VOX.

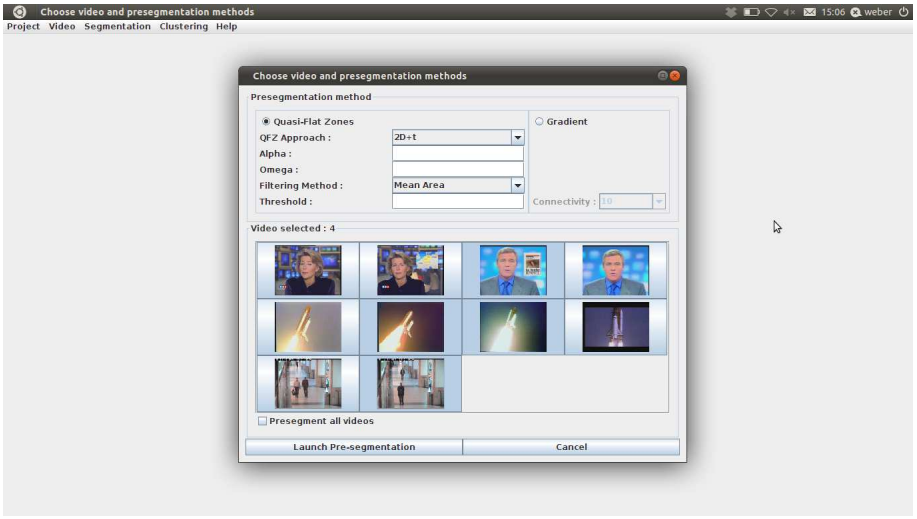


FIGURE B.2 – Vue d'ensemble de l'interface de Vox et l'interface permettant de lancer les calculs de pré-segmentation.

les objets-vidéo. Concernant la classification, si l'on opte pour le clustering ascendante contraint le système propose à l'utilisateur quelques échantillons de couple d'objets-vidéo (cf. figure B.3) : l'utilisateur doit alors indiquer si les deux objets-vidéo sont à contraindre par un *Must-Link* ou un *Cannot-Link*.

Une fois le clustering achevé, l'utilisateur doit déterminer la partition finale des objets-vidéo qu'il désire. Pour cela, il parcourt la hiérarchie des partitions et choisit le niveau qu'il désire à l'aide de l'interface présentée dans la figure B.4.

Du côté de la gestion des données, la structure de la base que crée Vox est écrite en XML, ce qui permet son édition même en dehors de VOX. Les séquences vidéo ajoutées au système, les

FIGURE B.3 – Interface proposant à l'utilisateur d'introduire une contrainte *Must-Link* ou *Cannot-Link* entre deux objets-vidéo représentés par une de leurs trames.

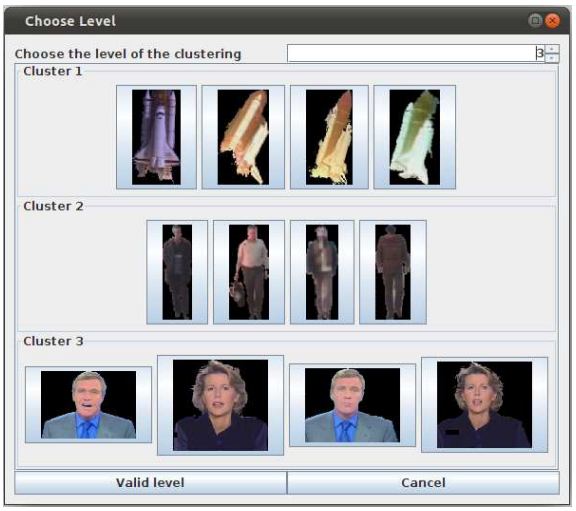


FIGURE B.4 – Interface de choix du niveau de la partition finale dans la hiérarchie.

pré-segmentations, les segmentations et les descriptions sont sauvegardées par une sérialisation de leur structure de données Java. Cette méthode permet de les charger très rapidement en mémoire.

VOX permet la segmentation interactive d'objets-vidéo et leur classification non-supervisée ou semi-supervisée. Sa construction modulaire permettra d'y ajouter de nouvelles méthodes pour les différentes étapes au fur et à mesure de son existence. Si les étapes hors-ligne de pré-segmentation et de description des objets-vidéo sont gourmandes en temps de calcul, les étapes en-ligne sont peu coûteuses et permettent une réaction rapide du système aux interactions de l'utilisateur.

Annexe C

ODESSA

L'absence d'équivalent vidéo de la base de Berkeley [MFTM01], nous a amené à créer des segmentations de séquences vidéo de référence. Afin de faciliter cette opération et pour permettre une évaluation facile des segmentations, nous avons développé le logiciel ODESSA (Outil D'Évaluation de Segmentation Supervisé et Assisté). Ce logiciel est écrit en langage Java et utilise la plate-forme de traitement d'images PELICAN (cf. annexe A). Son implantation repose, à l'instar de VOX sur le modèle MVC (Modèle-Vue-Contrôleur) qui sépare données, traitements et interface. Son interface générale est illustrée par la figure C.1. ODESSA comprend actuellement 31 classes représentant plus de 3000 lignes de code (la majorité des traitements sur les séquences vidéo sont effectués par des classes de PELICAN). Nous prévoyons de mettre ODESSA à la disposition de la communauté courant 2011.

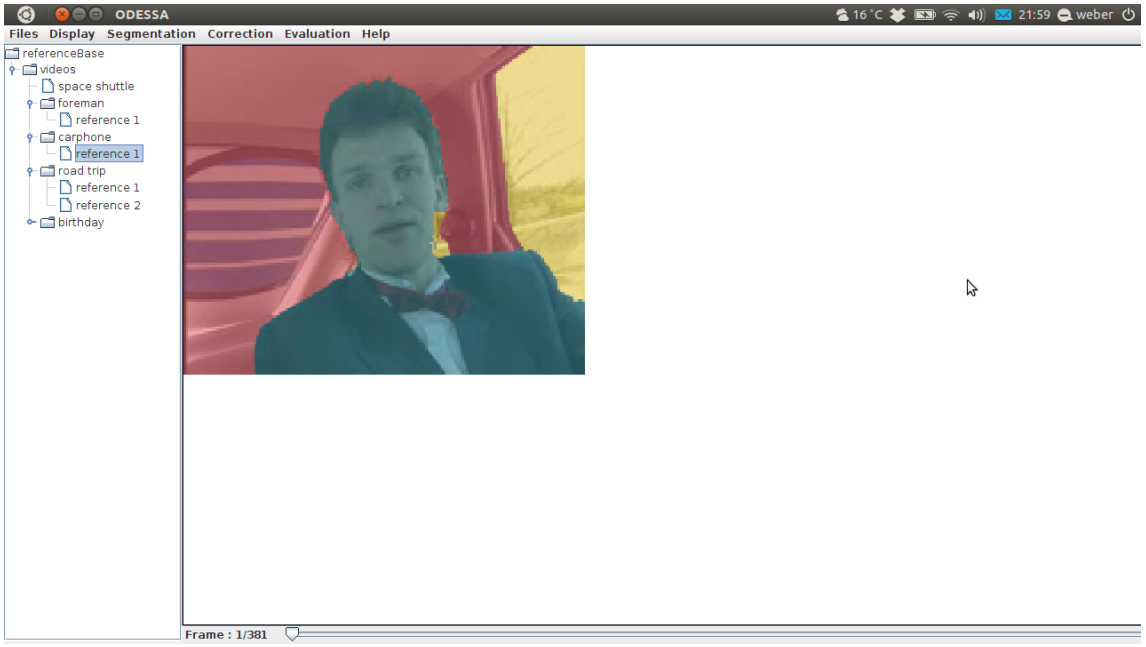


FIGURE C.1 – Vue d'ensemble de l'interface d'ODESSA.

L'objectif d'ODESSA est double, il s'agit de :

- permettre la création d'une base de segmentation de référence, similaire à la base de Berkeley mais pour les séquences vidéo ;
- permettre l'évaluation de segmentations par rapport à ces références.

ODESSA permet donc, à l'instar de la base de Berkeley, de créer plusieurs segmentations de référence pour une même séquence vidéo. Cette fonctionnalité est intéressante pour l'évaluation de sur-segmentation afin d'évaluer si l'assemblage des régions sur-segmentées permet d'obtenir les objets d'intérêts désirés par différents utilisateurs. Actuellement, 3 méthodes sont disponibles pour la réalisation de segmentation de référence : la méthode que nous proposons dans ce manuscrit, la *ligne de partage des eaux guidée par marqueurs* et la *ligne de partage des eaux propagée par marqueurs* [FL10]. Nous projetons d'ajouter d'autres méthodes, la construction modulaire du logiciel permettant de le faire facilement. Outre les méthodes de segmentation guidée, ODESSA propose un mécanisme de correction de segmentation de référence au niveau pixel (cf. figure C.2).

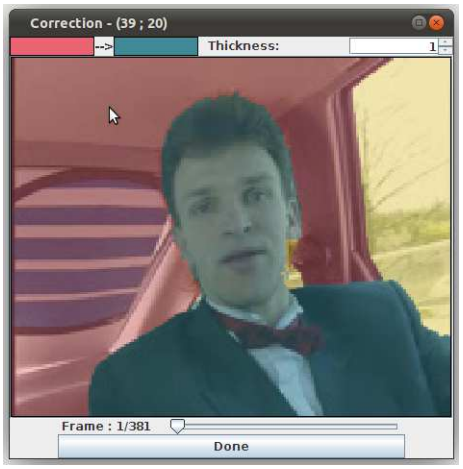


FIGURE C.2 – Interface de correction de segmentation de référence d'ODESSA.

Pour l'évaluation de segmentation par rapport à une ou plusieurs segmentations de référence, ODESSA propose les métriques utilisées dans ce manuscrit (cf figure C.3) :

- ratio de sur-segmentation
- précision maximale
- les différents indices de Jaccard

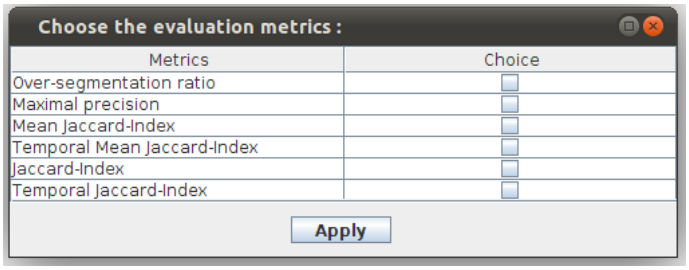


FIGURE C.3 – Interface de choix des métriques d'évaluation.

D'autres métriques seront ajoutées par la suite afin de permettre à la majorité des utilisateurs de disposer des métriques d'évaluation qui les intéressent. Les résultats d'évaluation sont exportables au format CSV (Comma-Separated Values) qui présente l'avantage d'être utilisable sur la majorité des tableurs ainsi que d'être facilement parsé.

Pour que les utilisateurs puissent évaluer les résultats obtenus par leurs algorithmes sur les données de référence, il est nécessaire qu'ils puissent les importer dans ODESSA. Il en est de même, s'ils disposent déjà de segmentations de référence et qu'ils veulent les intégrer dans le système pour bénéficer des métriques d'évaluation. Nous avons donc créé un format simple pour coder les

segmentations, afin que sa lecture et son écriture puisse être facilement implémentées par d'autres utilisateurs dans leurs logiciels. Ce format est un fichier texte. Il a été créé pour être facilement parsable, il n'est absolument pas optimisé pour prendre le moins de place possible en mémoire. Il peut au contraire produire des fichiers de segmentation volumineux si les séquences vidéo sont longues. Sa structure est la suivante :

```
<en-tête>
data
<données>
```

L'en-tête du fichier contient les informations suivantes :

```
date <chaîne de caractères> (optionel)
width <entier>// largeur de la segmentation vidéo
heigth <entier>// hauteur de la segmentation vidéo
length <entier> // durée de la segmentation vidéo (en trames)
```

L'en-tête et les données sur la segmentation sont séparées par une ligne contenant le mot « data ». Chaque ligne de la section *données* contient un couple d'entiers :

```
<nb_p> <e>
```

Ce format considère une segmentation vidéo comme étant un tableau unidimensionnel et non tri-dimensionnel (cf. figure C.4). e représente la valeur d'étiquette d'une région, nb_p le nombre de pixels consécutifs, dans le tableau unidimensionnel, ayant cette étiquette.

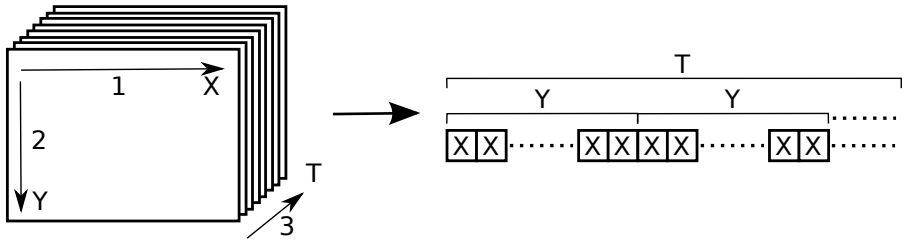


FIGURE C.4 – Réprésentation d'une séquence sous forme de tableau unidimensionnel.

Bibliographie

[AB94] R. Adams et L. Bischof : Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.

[AC07a] A. Anjulan et N. Canagarajah : A novel video mining system. *In 14th IEEE International Conference on Image Processing*, pages 185–188. IEEE, 2007.

[AC07b] A. Anjulan et N. Canagarajah : Object based video retrieval with local region tracking. *Signal Processing : Image Communication*, 22(7-8):607–621, 2007.

[AD00] L. Agnihotri et N. Dimitr : Video clustering using superhistograms in large archives. *In International Conference on Advances in Visual Information Systems (VISUAL)*, pages 62–73, 2000.

[AEH+00] O. Avaro, A. Eleftheriadisb, C. Herpelc, G. Rajand et L. Warde : MPEG-4 Systems : Overview. *Signal Processing : Image Communication*, 15(4–5):281–298, 2000.

[AL07] E. Aptoula et S. Lefèvre : A comparative study on multivariate mathematical morphology. *Pattern Recognition*, 40(11):2914–2929, 2007.

[AL09] E. Aptoula et S. Lefèvre : On the morphological processing of hue. *Image and Vision Computing*, 27(9):1394–1401, 2009.

[Ang03] J. Angulo : *Morphologie mathématique et indexation d'images couleur. Application à la microscopie en biomédecine*. Thèse de doctorat, Ecole des Mines de Paris, 2003.

[Ant02] S. Antani : A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition*, 35(4):945–965, Avril 2002.

[Apt08] E. Aptoula : *Analyse d'images couleur par morphologie mathématique. Application à la description, l'annotation et la recherche d'images*. Thèse de doctorat, Université Louis Pasteur - Strasbourg I, Juillet 2008.

[AS03] J. Angulo et J. Serra : Color segmentation by ordered mergings. *In Proceedings of the IEEE International Conference on Image Processing*, pages 125–128, 2003.

[Bad92] A. J. Baddeley : An error metric for binary images. *Robust Computer Vision*, pages 59–78, 1992.

[BC08] D. Brezeale et D.J. Cook : Automatic video classification : A survey of the literature. *IEEE Transactions on Systems, Man and Cybernetics-part C : Applications and Reviews*, 38(3):416–430, 2008.

[Ber02] P. Berkhin : Survey of clustering data mining techniques. Rapport technique, Accrue Software, San Jose, CA, 2002.

[BFC09] G. J. Brostow, J. Fauqueur et R. Cipolla : Semantic object classes in video : A high-definition ground truth database. *Pattern Recognition Letters*, 30:88–97, Janvier 2009.

[BG98] S. Balakrishnama et A. Ganapathiraju : Linear discriminant analysis - a brief tutorial. Institute for Signal and Information Processing, 1998.

[BJ01] Y. Y. Boykov et M. P. Jolly : Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. *In IEEE International Conference on Computer Vision*, volume 1, pages 105–112, 2001.

[BK04] Y. Boykov et V. Kolmogorov : An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1124–1137, Septembre 2004.

[BKP+11] D. Batra, A. Kowdle, D. Parikh, J. Luo et T. Chen : Interactively co-segmentating topically related images with intelligent scribble guidance. *International Journal of Computer Vision*, 93:273–292, Juillet 2011.

[BMEF09] D. Besiris, A. Makedonas, G. Economou et S. Fotopoulos : Combining graph connectivity & dominant set clustering for video summarization. *Multimedia Tools and Applications*, 44:161–186, Septembre 2009.

[BMM99] R. Brunelli, O. Mich et C.M. Modena : A survey on automatic indexing of video data. *Journal of Visual Communication and Representation*, 10(2):78–112, 1999.

[BNG03] U. Braga-Neto et J. Goutsias : A theoretical tour of connectivity in image processing and analysis. *Journal of Mathematical Imaging and Vision*, 19:5–31, July 2003.

[Bob01] M. Bober : MPEG-7 visual shape descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):716–719, 2001.

[BPK+09] D. Batra, D. Parikh, A. Kowdle, T. Chen et J. Luo : Seed image selection in interactive cosegmentation. *In Proceedings of the 16th IEEE international conference on Image processing*, pages 2369–2372, 2009.

[BRS04] R.V. Babu, K.R. Ramakrishnan et S.H. Srinivasan : Video object segmentation : A compressed domain approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(4):462–474, April 2004.

[BS07] D. Brunner et P. Soille : Iterative area filtering of multichannel images. *Image and Vision Computing*, 25(8):1352–1364, 2007.

[BZS08] A. Basharat, Y. Zhai et M. Shah : Content based video matching using spatiotemporal volumes. *Computer Vision and Image Understanding*, 110(3):360–377, 2008.

[CCC11] W.-S. Chu, X.-P. Chen et C.-S. Chen : Momi-cosegmentation : simultaneous segmentation of multiple objects among multiple images. *In Proceedings of the 10th Asian conference on Computer vision - Volume Part I*, pages 355–368. Springer-Verlag, 2011.

[CCM00] D. Cohn, R. Caruana et A McCallum : Semi-supervised clustering with user feedback. Rapport technique, TR2003-1892, Cornell University, 2000.

[CD85] J-P. Cocquerez et J. Devars : Détection de contours dans les images aériennes : Nouveaux opérateurs. *Traitement du signal*, 2(1):45–65, 1985.

[CDBPD07] F. Chevalier, J-P. Domenger, J. Benois-Pineau et M. Delest : Retrieval of objects in video by similarity based on graph matching. *Pattern Recognition Letters*, 28(8):939–949, 2007.

[CDF+04] G. Csurka, C. R. Dance, L. Fan, J. Willamowski et C. Bray : Visual categorization with bags of keypoints. *In International Workshop on Statistical Learning in Computer Vision (ECCV)*, pages 1–22, 2004.

[CDW05] A.P. Carleer, O. Debeir et E. Wolff : Assessment of very high spatial resolution satellite image segmentations. *Photogrammetric Engineering & Remote Sensing*, 71(11):1285–1294, Novembre 2005.

[CH67] T. Cover. et P. Hart : Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, Janvier 1967.

[Cha05] S. Chabrier : *Contribution à l'évaluation de performance en segmentation d'images*. Thèse de doctorat, Université d'Orléans, 2005.

[CIS11] Cisco visual networking index : Forecast and methodology, 2010-2015, white paper, 2011.

[CM02] D. Comaniciu et P. Meer : Mean shift : a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5): 603–619, Mai 2002.

[CP00] P. Correia et F. Pereira : Objective evaluation of relative segmentation quality. *In International Conference on Image Processing*, pages 308–311, Septembre 2000.

[CP05] D. Chetverikov et R. Peteri : A brief survey of dynamic texture description and recognition. *In Proceedings of International Conference on Computer Recognition Systems (CORES)*, pages 17–26. Springer, 2005.

[CPSK07] K.J. Cios, W. Pedrycz, R.W. Swiniarski et L.A. Kurgan : *Data Mining A Knowledge Discovery Approach*. Springer, 2007.

[CS94] J. Crespo et R. Schafer : The flat zone approach and color images. *In* J. Serra et P. Soille, éditeurs : *Mathematical morphology and its applications to image processing*, pages 85–92. Kluwer Academic Publishers, 1994.

[CSS+97] J. Crespo, R. Schafer, J. Serra, C. Gratin et F. Meyer : The flat zone approach : a general low-level region merging segmentation method. *Signal Processing*, 62(1):37–60, 1997.

[dALdLA11] S.E.F. de Avila, A.P.B. Lopes, A. da Luz et A.A. Araújo : Vsumm : A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32:56–68, Janvier 2011.

[Def77] D. Defays : An efficient algorithm for complete link method. *The Computer Journal*, 20:364–366, 1977.

[DFWL10] S. Derivaux, G. Forestier, C. Wemmert et S. Lefèvre : Supervised image segmentation using watershed transform, fuzzy classification and evolutionary computation. *Pattern Recognition Letters*, 31(15):2364–2374, 2010.

[DLR77] A.P. Dempster, N.M. Laird et D.B. Rubin : Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[DM09] F. Drucker et J. MacCormick : Fast superpixels for video analysis. *In Proceedings of the 2009 international conference on Motion and video computing (MVC)*, pages 55–62. IEEE Computer Society, 2009.

[DMAE99] N. Dimitrova, J. Martino, L. Agnihotri et H. Elenbaas : Color superhistograms for video representation. *In International Conference on Imap Processing (ICIP)*, pages 314–318, 1999.

[DZL06] K. Dai, J. Zhang et G. Li : Video mining : concepts, approaches and applications. *In Proceedings of the International Conference on Multi-Media Modelling (MMM)*, pages 477–480, Janvier 2006.

[Fas99] D. Fasulo : An analysis of recent work on clustering algorithms. Rapport technique, TR 01-03-02, University of Washington, 1999.

[FGMP08] M. Furini, F. Geraci, M. Montangero et M. Pellegrini : On using clustering algorithms to produce video abstracts for the web scenario. *In Proceedings of the IEEE Consumer Communication & Networking 2008 (CCNC2008)*, pages 1112–1116. IEEE Communication Society, Janvier 2008.

[FH04] P. F. Felzenszwalb et D. P. Huttenlocher : Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, Septembre 2004.

[FJ02] M.A.F. Figueiredo et A.K. Jain : Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, Mars 2002.

[FL10] F.C. Flores et R.A Lotufo : Watershed from propagated markers : An interactive method to morphological object segmentation in image sequences. *Image and Vision Computing*, 28(11):1491–1514, 2010.

[GGM04] H. Greenspan, J. Goldberger et A. Mayer : Probabilistic space-time video modeling via piecewise GMM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:384–396, 2004.

[GKHE10] M. Grundmann, V. Kwatra, M. Han et I. Essa : Efficient hierarchical graph-based video segmentation. volume 1, pages 2141–2148, 2010.

[GL98] C. Gu et M-C. Lee : Semiautomatic segmentation and tracking of semantic video objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):572–584, 1998.

[GLFT09] X. Gao, X. Li, J. Feng et D. Tao : Shot-based video retrieval with optical flow tensor and HMMs. *Pattern Recognition Letters*, 30(2):140–147, 2009.

[GM88] J.J. De Gruijter et A.B. McBratney : A modified fuzzy k-means method for predictive classification. *In Classification and Related Methods of Data Analysis : Proceedings of the First Conference of the International Federation of Classification Societies (IFCS)*, 1988.

[GRS00] S. Guha, R. Rastogi et K. Shim : CURE : an efficient clustering algorithm for large databases. *Information Systems*, 26(1):35–58, 2000.

[Gui06] M. Guironnet : *Méthodes de résumé de vidéo à partir d'informations bas niveau, du mouvement de caméra ou de l'attention visuelle.* Thèse de doctorat, Université Joseph Fournier, Grenoble, 2006.

[GWL07] F. Ge, S. Wang et T. Liu : New benchmark for image segmentation evaluation. *Journal of Electronic Imaging*, 16(3):16, 2007.

[Han03] A. Hanbury : A 3d-polar coordinate colour representation well adapted to image analysis. *In Scandinavian Conference on Image Analysis*, volume 2749 de *Lecture Notes in Computer Science*, pages 804–811. Springer, 2003.

[HBV01] M. Halkidi, Y. Batistakis et M. Vazirgiannis : Clustering algorithms and validity measures. *Scientific and Statistical Database Management*, pages 3–22, 2001.

[HCC06] C.-C. Hsu, H.T. Chang et T.-C. Chang : Efficient moving object extraction in compressed low-bit-rate video. *In Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia*, pages 411–414, Washington, DC, USA, 2006. IEEE Computer Society.

[HHM94] S. Hambrusch, X. He et R. Miller : Parallel algorithms for gray-scale digitized picture component labeling on a mesh-connected computer. *Journal of Parallel and Distributed Computing*, 20(1):56–68, 1994.

[HS81] B.K.P. Horn et B.G. Schunk : Determining optical flow. *Artificial Intelligence*, 15:185–204, 1981.

[Hua07] J. Huart : *Extraction et analyse d'objets-clés pour la structuration d'images et de vidéos.* Thèse de doctorat, INP Grenoble, 2007.

[IP97] F. Idris et S. Panchanathan : Review of Image and Video Indexing Techniques. *Journal of Visual Communication and Image Representation*, 8(2):146–166, 1997.

[Jac01] P. Jaccard : Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[JBP10] A. Joulin, F. Bach et J. Ponce : Discriminative clustering for image co-segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1943–1950, 2010.

[JD01] S. Jeannin et A. Divakaran : MPEG-7 visual motion descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):720–724, 2001.

[JMF99] A. K. Jain, M. N. Murty et P. J. Flynn : Data clustering : a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[JYTK11] F. Jiang, J. Yuan, S. A. Tsaftaris et A. K. Katsaggelos : Anomalous video event detection using spatiotemporal context. *Computer Vision and Image Understanding*, 115(3):232–333, Mars 2011.

[KKM02] S. Kamvar, D. Klein et C. Manning : Interpreting and extending classical agglomerative clustering algorithms using a model-based approach. *In International Conference on Machine Learning (ICML)*, pages 283–290, 2002.

[KR84] L. Kitchen et A. Rosenfeld : Scene analysis using region-based constraint filtering. *Pattern Recognition*, 17(2):189–203, 1984.

[KR90] L. Kaufman et P.J. Rousseeuw : *Finding Groups in Data An Introduction to Cluster Analysis*. Wiley Interscience, New York, 1990.

[LC09] D. Liu et T. Chen : Video retrieval based on object discovery. *Computer Vision and Image Understanding*, 113(3):397–404, 2009.

[LC11] S. Lefèvre et V. Claveau : Topic segmentation : application of mathematical morphology to textual data. *In International Symposium on Mathematical Morphology (ISMM)*, volume 6671, pages 472–481, Intra, Italie, Juillet 2011. Springer-Verlag Lecture Notes on Computer Science.

[Lef09] S. Lefèvre : Approches multivaluées et supervisées en morphologie mathématique et applications en analyse d'image. Habilitation à diriger des recherches, Université de Strasbourg, 2009.

[LHV03] S. Lefèvre, J. Holler et N. Vincent : A review of real-time segmentation of uncompressed video sequences for content-based search and retrieval. *Real-Time Imaging*, 9(1):73–98, 2003.

[LN85] M.D. Levine et A.M. Nazif : Dynamic measurement of computer generated image segmentations. *IEEE Transactions on Pattern Analysis and Machine intelligence*, 7(2):155–164, 1985.

[LOH05] J. Lee, J. Oh et S. Hwang : Clustering of video objects by graph matching. *In IEEE International Conference on Multimedia and Expo (ICME)*, pages 394–397, Juillet 2005.

[Low99] D.G. Lowe : Object recognition from local scale-invariant features. *In IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157, 1999.

[LYP05] Z. Liu, J. Yang et N.S. Peng : Semi-automatic video object segmentation using seeded region merging and bidirectional projection. *Pattern Recognition Letters*, 26(5):653–662, Avril 2005.

[MA08] A.G. Money et H. Agius : Video summarisation : A conceptual framework and survey of the state of the art. *Journal of Visual Communication and Image Representation*, 19(2):121–143, 2008.

[Mac67] J. B. MacQueen : Some methods for classification and analysis of multivariate observations. *In* L. M. Le Cam et J. Neyman, éditeurs : *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[Mat67] G. Matheron : *Eléments pour une théorie des milieux poreux*. Masson, Paris, 1967.

[Mat75] G. Matheron : *Random Sets and Integral Geometry*. Wiley, New York, 1975.

[MBPL05] F. Manerba, J. Benois-Pineau et R. Leonardi : Real-time rough extraction of foreground objects in MPEG 1,2 compressed video. *In Proceeding of the Workshop on Image Analysis For Multimedia Interactice Services(WIAMIS)*, Avril 2005.

[Meu05] Cyril Meurie : *Segmentation d'images couleur par classification pixellaire et hiérarchie de partitions*. Thèse de doctorat, Université de Caen, 2005.

[MFTM01] D. Martin, C. Fowlkes, D. Tal et J. Malik : A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. *In Proceedings of the 8th International Conference on Computer Vision*, volume 2, pages 416–423, Juillet 2001.

[ML98] F. Marques et J. Llach : Tracking of generic objects for video object generation. *In IEEE International Conference on Image Processing*, volume 3, pages 628–632, Los Alamitos, CA, USA, 1998. IEEE Computer Society.

[MM99] F. Meyer et P. Maragos : Morphological scale-space representation with levelings. *In International Conference on Scale-Space Theories in Computer Vision*, volume 1682, pages 187–198. Lecture Notes in Computer Science, 1999.

[MMM10] E. Moxley, T. Mei et B. S. Manjunath : Video annotation through search and graph reinforcement mining. *IEEE Transactions on Multimedia*, 12(3):184–193, 2010.

[MO10] K. McGuinness et N.E. O'Connor : A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444, Février 2010.

[MOVY01] B. S. Manjunath, J. R. Ohm, V. V. Vasudevan et A. Yamada : Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6): 703–715, 2001.

[MSS02] B. Manjunath, P. Salembier et T. Sikora : *Introduction to MPEG-7*. Wiley, 2002.

[MTP04] M. Muhlbaier, A. Topalis et R. Polikar : Learn++.mt : A new approach to incremental learning. *In Multiple Classifier Systems*, pages 52–61, 2004.

[MTS+05] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir et L. Van Gool : A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005.

[MZC+99] B. Marcotegui, F. Zanoguera, P. Correia, R. Rosa, R. Mech et M. Wollborn : A video object generation tool allowing friendly user interaction. *In IEEE International Conference on Image Processing*, volume 2, pages 391–395, 1999.

[NMI79] M. Nagao, T. Matsuyama et Y. Ikeda : Region extraction and shape analysis in aerial photographs. *Computer Graphics and Image Processing*, 10(3):195–223, 1979.

[OS11] G. K. Ouzounis et P. Soille : Pattern spectra from Partition Pyramids and Hierarchies. *In International Symposium on Mathematical Morphology and Its Application to Signal and Image Processing*, pages 108–119, Berlin, Heidelberg, 2011. Springer-Verlag.

[OW11] G. K. Ouzounis et M.H.F. Wilkinson : Hyperconnected Attribute Filters Based on k-Flat Zones. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):224–239, 2011.

[PCB08] S. Poullot, M. Crucianu et O. Buisson : Scalable mining of large video databases using copy detection. *In Proceeding of the 16th ACM international conference on Multimedia (MM)*, pages 61–70. ACM, Octobre 2008.

[PLC11] B. Perret, S. Lefèvre et C. Collet : Toward a new axiomatic for hyperconnections. *In International Symposium on Mathematical Morphology and Its Application to Signal and Image Processing*, pages 85–95, Berlin, Heidelberg, 2011. Springer-Verlag.

[PMC09] B. L. Price, B. S. Morse et S. Cohen : Livecut : Learning-based interactive video segmentation by evaluation of multiple propagated cues. *In IEEE International Conference on Computer Vision*, pages 779–786, 2009.

[RBD92] J.-F. Rivest, S. Beucher et J. Delhomme : Marker-controlled segmentation : an application to electrical borehole imaging. *Journal of Electronic Imaging*, 1(2):136–142, 1992.

[RDD02] A. Rosenfeld, D. Doermann et D. DeMenthon, éditeurs. *Video Mining*. Springer, 2002.

[RHC99] Y. Rui, T.S. Huang et S.F. Chang : Image retrieval : current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10(4):39–62, 1999.

[RL03] I. Ruthven et M. Lalmas : A survey on the use of relevance feedback for information access systems. *The Knowledge Engineering Review*, 18(2):95–145, 2003.

[RP66] A. Rosenfeld et J.L. Pfaltz : Sequential operations in digital picture processing. *Journal of the ACM*, 13:471–494, Octobre 1966.

[RSSZ09] W. Ren, S. Singh, M. Singh et Y.S. Zhu : State-of-the-art on spatio-temporal information based video retrieval. *Pattern Recognition*, 42(2):267–282, Février 2009.

[RZ08] W. Ren et Y. Zhu : A video summarization approach based on machine learning. *In International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 450–453, Los Alamitos, CA, USA, 2008. IEEE Comp. Soc.

[Ser98] J. Serra : Connectivity on complete lattices. *Journal of Mathematical Imaging and Vision*, 9:231–251, 1998.

[Ser06] J. Serra : A lattice approach to image segmentation. *Journal of Mathematical Imaging and Vision*, 24(1):83–130, 2006.

[SG09] P. Soille et J. Grazzini : Constrained connectivity and transition regions. *In International Symposium on Mathematical Morphology and Its Application to Signal and Image Processing*, pages 59–69, Berlin, Heidelberg, 2009. Springer-Verlag.

[Sib73] R. Sibson : SLINK : An optimally efficient algorithm for the single-link cluster method. *Computer Journal*, 16(1):30–34, 1973.

[Sik01] T. Sikora : The MPEG-7 visual standard for content description-an overview. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):696–702, 2001.

[SM00] J. Shi et J. Malik : Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.

[Smi78] A.R. Smith : Color gamut transform pairs. *In Proceedings of Conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 12–19, New York, NY, USA, 1978.

[SOD10] A. F. Smeaton, P. Over et A. R. Doherty : Video shot boundary detection : Seven years of TRECVid activity. *Computer Vision and Image Understanding*, 114(4):411 – 418, 2010.

[Soi07] P. Soille : On genuine connectivity relations based on logical predicates. *In Proceedings of the 14th International Conference on Image Analysis and Processing*, pages 487–492, Washington, DC, USA, 2007. IEEE Computer Society.

[Soi08] P. Soille : Constrained connectivity for hierarchical image partitioning and simplification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1132–1145, Juillet 2008.

[Soi10] P. Soille : Constrained connectivity for the processing of very-high-resolution satellite images. *International Journal of Remote Sensing*, 31(22):5879–5893, 2010.

[Soi11] P. Soille : Preventing Chaining through Transitions While Favouring It within Homogeneous Regions. *In International Symposium on Mathematical Morphology and Its Application to Signal and Image Processing*, pages 96–107, Berlin, Heidelberg, 2011. Springer-Verlag.

[SOK06] A.F. Smeaton, P. Over et W. Kraaij : Evaluation campaigns and TRECVid. *In Proceedings of the ACM international workshop on Multimedia Information Retrieval (MIR)*, pages 321–330, New York, NY, USA, 2006.

[SS93] J. Serra et P. Salembier : Connected operators and pyramids. *In Proceedings of SPIE, Non-Linear Algebra and Morphological Image Processing*, volume 2030, pages 65–76, 1993.

[SS95] P. Salembier et J. Serra : Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Transactions on Image Processing*, 4:1153–1160, 1995.

[SS01] L. Shapiro et G. Stockman : *Computer Vision*. Prentice-Hall, Inc., 2001.

[SW09] C. G. M. Snoek et M. Worring : Concept-based video retrieval. *Foundations and Trends in Information Retrieval*, 4(2):215–322, 2009.

[SZ08] J. Sivic et A. Zisserman : Efficient visual search for objects in videos. *Proceedings of the IEEE*, 96(4):548–566, 2008.

[SZB09] F. Schroff, C. L. Zitnick et S. Baker : Clustering Videos by Location. *In Proceedings of the British Machine Vision Conference*, 2009.

[SZR11] J. Steiner, S. Zollmann et G. Reitmayr : Incremental superpixels for real-time video analysis. *In 16th Computer Vision Winter Workshop*, Février 2011.

[TCAA05] B.U. Toreyin, A.E. Cetin, A. Aksay et M.B. Akhan : Moving object detection in wavelet compressed video. *Signal Processing : Image Communication*, 20(3):255–264, Mars 2005.

[TCR09] L. F. Teixeira et L. Corte-Real : Video object matching across multiple independent views using local descriptors and adaptive learning. *Pattern Recognition Letters*, 30(2):157–167, 2009.

[Tek05] A.M. Tekalp : Handbook of image & video processing, second edition. chapitre Video Segmentation, pages 471–489. Elsevier, 2005.

[TSL00] J.B. Tenenbaum, V. Silva et J.C. Langford : A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, Décembre 2000.

[TVC09] P. Turaga, A. Veeraraghavan et R. Chellappa : Unsupervised view and rate invariant clustering of video sequences. *Computer Vision and Image Understanding*, 113:353–371, Mars 2009.

[UBD05] T. Urruty, F. Belkouch et C. Djeraba : Kpyr, une structure efficace d'indexation de documents vidéo. *In INFORSID*, pages 403–418, 2005.

[VKR10] S. Vicente, V. Kolmogorov et C. Rother : Cosegmentation revisited : Models and optimization. *In European Conference on Computer Vision (ECCV)*, pages 465–479, 2010.

[VM95] C. Vachier et F. Meyer : Extinction value : a new measurement of persistence. *In IEEE Workshop on nonlinear signal and image processing*, volume 1, pages 254–257, 1995.

[VM04] P. Villegas et X. Marichal : Perceptually-weighted evaluation criteria for segmentation masks in video sequences. *IEEE Transactions on Image Processing*, 13(8):1092–1103, Août 2004.

[Voo86] E.M. Voorhees : Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. *Information Processing and Management*, 22(6):465–476, 1986.

[VS91] L. Vincent et P. Soille : Watersheds in digital spaces : An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, Juin 1991.

[Wil] M. Wilkinson : An axiomatic approach to hyperconnectivity. *In International Symposium on Mathematical Morphology and Its Application to Signal and Image Processing*, pages 35–46, Berlin, Heidelberg. Springer-Verlag.

[WLG10] J. Weber, S. Lefèvre et P. Gancarski : Video object mining : Issues and perspectives. *In IEEE International Conference on Semantic Computing (ICSC)*, pages 85–90, Pittsburgh, USA, Septembre 2010.

[WLG11a] J. Weber, S. Lefevre et P. Gançarski : Fouille vidéo orientée objet, une approche générique. *In Atelier Fouille de données complexes, Journées Francophones Extraction et Gestion des Connaissances (EGC 2011)*, pages 9–20, Janvier 2011.

[WLG11b] J. Weber, S. Lefèvre et P. Gançarski : Interactive video segmentation based on quasi-flat zones. *In IEEE International Symposium on Image and Signal Processing and Analysis (ISPA)*, Dubrovnik, Croatie, Septembre 2011.

[WLG11c] J. Weber, S. Lefèvre et P. Gançarski : Segmentation vidéo interactive par zones quasi-plates. *In Colloque GRETSI sur le Traitement du Signal et des Images*, Bordeaux, France, Septembre 2011.

[WLG11d] J. Weber, S. Lefèvre et P. Gançarski : Spatio-temporal quasi-flat zones for morphological video segmentation. *In International Symposium on Mathematical Morphology (ISMM)*, volume 6671, pages 178–189, Intra, Italie, Juillet 2011. Springer-Verlag Lecture Notes on Computer Science.

[WLG11e] J. Weber, S. Lefèvre et P. Gançarski : Zones quasi-plates spatio-temporelles et segmentation morphologique de séquences vidéo. *In ORASIS - Congrès des jeunes chercheurs en vision par ordinateur*, Praz-sur-Arly France, Juin 2011.

[XI05] R. Xu et D. Wunsch II : Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, May 2005.

[YLP10] J. You, G. Liu et A. Perkis : A semantic framework for video genre classification and event analysis. *Signal Processing : Image Communication*, 25(4):287–302, 2010.

[YMB77] W. A. Yasnoff, J. K. Mui et J. W. Bacus : Error measures for scene segmentation. *Pattern Recognition*, 9:217–231, 1977.

[YRLT09] J. Yuen, B. C. Russell, C. Liu et A. Torralba : Labelme video : Building a video database with human annotations. *In Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1451–1458, 2009.

[Zan01] F. Zanoguera : *Segmentation interactive d'images fixes et de séquences vidéo basée sur des hierarchies de partitions*. Thèse de doctorat, Ecole des Mines de Paris, 2001.

[Zeb88] R. Zeboudj : *Filtrage, Seuillage Automatique, Contraste et Contours : du Pré-Traitement à l'Analyse d'image*. Thèse de doctorat, Université de Saint Etienne, 1988.

[ZFG08] H. Zhang, J. E. Fritts et S. A. Goldman : Image segmentation evaluation : A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, 2008.

[Zha96] Y. J. Zhang : A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346, 1996.

[Zha01] Y. J. Zhang : A review of recent evaluation methods for image segmentation. *In Sixth International Symposium on Signal Processing and its Applications*, volume 1, pages 148–151, 2001.

[Zhu08] X. Zhu : Semi-supervised learning literature survey. Rapport technique, University of Wisconsin-Madison, 2008. TR-1530.

[ZL02] D. Zhang et G. Lu : Shape-based image retrieval using generic fourier descriptor. *Signal Processing : Image Communication*, 17(10):825 – 848, 2002.

[ZL03] D. Zhang et G. Lu : Evaluation of MPEG-7 shape descriptors against other shape descriptors. *Multimedia Systems*, 9:15–30, 2003.

[ZLTZ07] S. Zhai, B. Luo, J. Tang et C.-Y. Zhang : Video abstraction based on relational graphs. *In Proceedings of the Fourth International Conférence on Image and Graphics*, pages 827–832. IEEE Computer Society, 2007.

[ZM02] F. Zanoguera et F. Meyer : On the implementation of non-separable vector levelings. *In International Symposium on Mathematical Morphology*, pages 369–377. CSIRO Publishing, 2002.

[ZRL96] T. Zhang, R. Ramakrishnan et M. Livny : BIRCH : an efficient data clustering method for very large databases. *SIGMOD Record*, 25(2):103–114, 1996.

[ZZC96] D. Zhong, H. Zhang et S.F. Chang : Clustering Methods for Video Browsing and Annotation. *In Storage and Retrieval for Image and Video Databases (SPIE)*, pages 239–246, 1996.

Résumé

Nous observons actuellement une augmentation importante du volume de données vidéo disponibles. L'utilisation efficace de cette masse de données nécessite d'en extraire de l'information. Dans cette thèse, nous proposons d'utiliser les méthodes de fouille de données et de les appliquer sur les objets-vidéo d'intérêt afin de combler le fossé sémantique en impliquant l'utilisateur dans le processus. Extraire ces objets à partir des pixels nécessite de manipuler un grand volume de données, induisant un traitement couteux (en temps et en mémoire) peu compatible avec une implication interactive de l'utilisateur. Ainsi, nous proposons d'appliquer le processus interactif de segmentation sur une réduction des données, les zones quasi-plates. N'étant définies que pour les images fixes, nous proposons une extension des zones quasi-plates aux séquences vidéo ainsi qu'une nouvelle méthode de filtrage. La segmentation est effectuée interactivement par l'utilisateur qui dessine des marqueurs sur les objets d'intérêt afin de guider la fusion des zones quasi-plates composant ces objets. Elle est effectuée sur un graphe d'adjacence de régions représentant les zones quasi-plates spatiotemporelles ainsi que leurs relations d'adjacence. L'utilisation de cette structure assure un faible temps de calcul. Les objets-vidéo obtenus sont ensuite utilisés dans un processus de fouille interactif guidé par des descripteurs extraits automatiquement de la video et des informations données par l'utilisateur. La forte interactivité avec l'utilisateur, à la fois lors de l'étape de segmentation puis lors de l'étape de fouille favorise la synergie entre données numériques et interprétation humaine.

Mots-clés : Segmentation vidéo ; zones quasi-plates ; morphologie mathématique ; segmentation interactive ; fouille de données vidéo ; filtrage ; objet-vidéo.

Abstract

Today, we observe an expansion of the amount of available video data. Efficient use of this data mass requires to be able to extract information from it. In this thesis, we propose to use data mining methods and apply them on video-objects of interest, in order to bridge the semantic gap by involving the user in the process. The extraction of such video-objects from pixels implies the handling of large data volume. This leads to expensive computing (in terms of computation time and memory) which is not compatible with an interactive user involvement. Thus, we propose to apply the interactive segmentation process on a data reduction, the quasi-flat zones. Quasi-flat zones are only defined for still images, so we propose an extension of the quasi-flat zones to video data and a new filtering method. The segmentation is performed interactively by the user which has to draw markers on the objects of interest, in order to guide the merging of the quasi-flat zones which compose these objects. This process is performed on a region adjacency graph which contains spatiotemporal quasi-flat zones as nodes and their spatiotemporal adjacency relations as edges. The use of such structure provides a low computation cost. Obtained video objects are then used in an interactive mining process guided by descriptors automatically extracted from the video and information given by the user. The high interactivity with the user, both at the segmentation step and at the mining step promotes synergy between digital data and human interpretation.

Keywords : Video segmentation ; quasi-flat zones ; mathematical morphology ; interactive segmentation ; video mining ; filtering ; video-object.