Segmentation et structuration d'un document vidéo pour la caractérisation et l'indexation de son contenu sémantique

Claire-Hélène Demarty

Segmentation et Structuration d'un Document Vidéo pour la Caractérisation et l'Indexation de son Contenu Sémantique

Application aux Journaux Télévisés

THÈSE

présentée à
l'École Nationale Supérieure des Mines de Paris
par

Claire-Hélène Demarty

pour obtenir le titre de

DOCTEUR
en
MORPHOLOGIE MATHÉMATIQUE

Soutenue le 24 janvier 2000 devant le jury composé de :

Jean SERRA	*Président*	École des Mines de Paris
Claude LABIT	*Rapporteur*	IRISA
Philippe SALEMBIER	*Rapporteur*	Université Polytechnique de Catalogne
Philippe JOLY	*Examinateur*	LIP6 - Université Paris 6
Rémi RONFARD	*Examinateur*	Institut National de l'Audiovisuel
Henri SANSON	*Examinateur*	CCETT - France Télécom
Serge BEUCHER	*Directeur de thèse*	École des Mines de Paris

Remerciements

Il est coutumier de réserver une page, la première, en début d'un mémoire de thèse, aux remerciements que l'heureux ex-thésard nouvellement promu "docteur" adresse à tous ceux qui, durant trois ans ou plus, ont su l'entourer, le conseiller et le guider sur le chemin épineux qu'est celui de la réalisation d'une thèse. Et bien sûr je ne faillirai pas à la règle, puisqu'il m'appartient ici d'y écrire tout ce qui a aussi fait partie de ma vie, non seulement professionnelle, mais aussi affective et relationnelle. Avant de me lancer, je désire cependant souligner la difficulté d'un autre genre de cet exercice, puisqu'il s'agit ici de résumer en une page, l'ensemble des rencontres qui auront contribué à l'aboutissement de ces quatre années de recherche. Ces quelques remerciements seront donc succincts, mais j'espère les avoir déjà bien entamés d'une certaine façon de vive voix pendant mon séjour au Centre de Morphologie Mathématique.

Je tiens donc vivement à remercier :

- en tout premier lieu Serge Beucher, mon directeur de thèse, pour son écoute, sa grande disponibilité et les conseils avisés qu'il a su me prodiguer ;
- Jean Serra, qui a bien voulu m'accueillir au sein de son laboratoire et m'offrir la chance de poursuivre ma découverte de la Morphologie Mathématique ;
- Henri Sanson, Françoise Chassaing, Philippe Salembier, Jacques Levy-Vehel, Luis Garrido et Bertrand Guilheneuf, pour avoir soutenu et suivi attentivement mes recherches dans le cadre de notre CTI-CCETT ;
- Philippe Joly, Claude Labit et Rémi Ronfard, pour avoir gentiment accepté de juger mon travail en participant à mon jury ;
- Etienne Decencière, pour ses nombreuses et avisées relectures des versions préliminaires de ce mémoire ;
- Lothar Bergen pour son amitié, ses encouragements et nos nombreuses discussions ;
- tous mes collègues et amis qui ont partagé un bout de ma route durant ces quatre années et qui ont contribué activement à les rendre si riches et agréables : Laura Andriamasinoro, Jesus Angúlo, Antoine Aubert, Christophe Bernard, Hélène Beucher, Nicolas Bez, Michel Bilodeau, Roland Brémond, Jacek Cichosz, Luc Decker, Arnaud Delarue, Elena Diaz, Michel Gauthier, Cristina Gomila, Dimitri et Irena Gorokhovik, Christophe Gratin, Nicole Guégan, Allan Hanbury, Marcin Iwanowski, Nicolas Jeannée, Dominique Jeulin, Jean-Claude Klein, Pascal Laurenge, Marc Le Guyader, Fabrice Lemonnier, Beatriz Marcotegui, Fernand Meyer, Mariusz Mlynarczuk, Liliane Pipault, Marie-Hélène Pires, Óscar Ribes, Valéry Risson, Chris Roth, Rosa Ruiloba, Raphaël Sasportas, Laurent Savary, Edouard Squillaci, Hugues Talbot, Gilberto Tillian, Corinne Vachier, Thomas Walter, Marc Waroquier, Kiki Wiginton, Shahram Zahirazami, Frédéric Zana, Francisca Zanoguera ...

Enfin, cette longue liste ne serait pas complète si je ne terminais pas par ma famille, mes parents et beaux-parents, et surtout Joël et petite Marine, qui ont dû bien souvent supporter une femme et maman pas très disponible, et qui malgré tout, m'ont apporté tout l'amour et le soutien nécessaires durant les moments difficiles de ces années d'équilibrisme entre thèses (au pluriel !) et bébés (bientôt au pluriel aussi !).

À mes parents,
À Joël,
À notre petite Marine.

Table des matières

Chapitre 1

Introduction

1.1 Avant-propos

L'indexation est, depuis déjà quelques dizaines d'années, un thème de recherche très prisé. Après l'indexation de documents textuels, il s'agit aujourd'hui d'être capable d'indexer des documents vidéo, audio, ou plus généralement multimédia. Pour saisir pleinement l'intérêt accru porté à l'indexation, attardons-nous sur la définition de ce terme telle qu'elle est proposée dans la littérature [63, 28] : cette notion correspond à l'organisation des données à indexer, en fonction d'un ordre à définir basé sur un ou des attributs particuliers. Bien sûr cet ordonnancement sous-entend auparavant que l'on a été capable de localiser ces attributs et d'en exprimer leurs valeurs, au sein même des données.

Une telle volonté d'organiser les données multimédia découle alors tout naturellement de la constatation de la multiplication du nombre de ces documents, leur croissance exponentielle étant directement due à la mise à disposition de moyens techniques aisés d'utilisation et peu coûteux de création de telles données, mais aussi de moyens de communication accrus et facilités par l'avènement d'internet.

Face à cette multitude de données multimédia de contenus absolument divers, se pose alors le problème de la recherche d'informations au sein de ces bases de données gigantesques, qu'il n'est plus possible actuellement de trier et classifier entièrement manuellement. Or cette recherche d'informations est aujourd'hui à la base de nombreuses applications journalistiques, éducatives, médicales, etc., qui ne peuvent fonctionner sans une indexation solide préalable des données. La constatation du caractère essentiel de l'indexation a ainsi donné lieu au développement d'une nouvelle norme en cours de conception, MPEG7, qui vise à la normalisation de toutes les informations, appelées descripteurs, qu'il est possible d'extraire d'un document. En revanche les outils d'extraction de ces descripteurs ne sont pas visés par cette normalisation.

Dans ce contexte, et puisqu'il s'agit d'atteindre et d'extraire toute information sémantique d'un document, toute intervention manuelle n'est pas obligatoirement à bannir du processus d'indexation. Il apparaît toutefois essentiel de concevoir de nouveaux outils, capables d'amorcer le processus d'indexation en réalisant de façon automatique les premières étapes, plus "simples", mais aussi et surtout plus fastidieuses, que sont la structuration du document et l'extraction de premiers attributs. Par structuration, on entend le découpage linéaire du document en entités variées (par exemple les prises de vue, les images, etc. pour un document vidéo) et la réorganisation, le regroupement de ces entités en fonction des liens ou relations qui les unissent, que ces entités soient adjacentes ou non. Par la suite, nous nommerons respec-

tivement ces deux notions structuration linéaire et structuration relationnelle. L'élaboration d'outils automatiques réalisant cette première phase permet ainsi de réserver l'intervention manuelle à la détection de critères sémantiques d'un niveau plus élevé, pour lesquels une action humaine reste la seule efficace actuellement.

Ce mémoire vise donc l'élaboration et le développement d'outils automatiques, destinés à l'établissement d'une structuration linéaire et relationnelle d'un document vidéo et à l'extraction de premiers attributs de ce document. Notons d'ores et déjà à ce stade que nous nous sommes restreints aux documents de type vidéo et non aux documents multimédia en général, mais nous reviendrons sur ce point lors de la description du cadre de notre travail dans la section suivante.

En plus du caractère automatique de la structuration d'un document vidéo que nous désirons élaborer au final, la grande diversité des données à indexer impose également que ces outils soient les plus généraux possibles, i.e. applicables à des documents vidéo de contenu quelconque. Cette contrainte conduit donc à la difficulté supplémentaire d'extraire des informations de documents qu'on ne connaît pas a priori. En d'autres mots, il s'agit d'extraire toute information possible sans savoir à l'avance ce que les images ou les séquences peuvent contenir.

La deuxième question soulevée naturellement par le problème que nous cherchons à résoudre concerne notre capacité à l'heure actuelle à extraire automatiquement, par des techniques existantes de traitement d'images et d'intelligence artificielle, un haut niveau sémantique d'information. Dans un domaine applicatif restreint, les outils élaborés bénéficient de la connaissance dont on dispose sur les données d'entrée qui sont toutes d'un même type (médicales par exemple). De ce fait, il est envisageable d'atteindre dans ce cas un niveau sémantique élevé. Dans le cadre plus général que nous nous sommes fixés, i.e. les documents en entrée sont de contenus variés, il est au contraire peu probable qu'il soit possible d'extraire directement la totalité de l'information sémantique par des outils de traitements d'images. Les travaux existants à ce jour dans le domaine de l'indexation confirment par ailleurs cette assertion, par leurs résultats et leur répartition en techniques d'un bas niveau [76, 41, 56] et d'un haut niveau sémantique [31, 32, 29, 90].

Aussi avons-nous adopté une position intermédiaire qui consistera par la suite à n'élaborer que des outils d'un relatif bas niveau sémantique, mais suffisamment généraux pour être applicables à toute sorte de documents vidéo. Nous faisons également le pari que, plus qu'un seul outil sophistiqué, complexe et donc peu robuste, la combinaison de plusieurs outils plus simples, associés à des règles logiques, permet d'accéder à un niveau sémantique élevé, à moindre coût. Ces outils de base seront également vus par la suite comme une série de prétraitements simples et rapides permettant de déterminer en aval quel autre traitement plus spécifique serait à appliquer à des morceaux particuliers d'un document.

L'obtention de ces qualités de simplicité et rapidité, tout en conservant un niveau de résultats et une robustesse fiables, sera possible par l'utilisation d'opérateurs de morphologie mathématique. Ce choix permettra ainsi à nos outils de bénéficier des capacités de filtrages et d'extraction d'objets de cette théorie, qui a par ailleurs déjà prouvé sa puissance dans de nombreuses applications et problèmes industriels de traitement d'images.

En résumé, et avant de passer à la description plus technique du cadre de nos travaux, ce mémoire a pour but d'élaborer une structure à la fois linéaire et relationnelle d'un document

vidéo quelconque et de fournir les outils automatiques permettant de la réaliser. Ces outils devront partager les deux caractéristiques de généralité de leur domaine d'application (documents vidéo quelconques en entrée) et de simplicité et rapidité des techniques mises en œuvre, le but final étant de prouver que la combinaison de tels outils permet d'élaborer une structure déjà très complète des documents, renseignant sur un contenu sémantique d'un niveau déjà élevé.

Nous aurons donc le souci constant de mettre en avant ces qualités particulières à la fois par une description détaillée des choix que nous serons amenés à effectuer et des caractéristiques techniques des outils en résultant, mais aussi par de nombreux exemples et applications permettant d'illustrer concrètement leur potentiel. En outre nous aurons à cœur, tout au long de ce mémoire, de poursuivre une certaine continuité, une ligne directrice de nos travaux visant à complèter au fur et à mesure de la conception de nouveaux outils, une structure de document vidéo toujours plus complexe et plus élaborée.

Enfin, nous ne pouvons terminer cet avant-propos sans souligner que ce mémoire est l'aboutissement d'une collaboration entre le Centre de Morphologie Mathématique, le groupe Fractales de l'INRIA et l'Unité de Théorie du Signal et Communications de l'Université Polytechnique de Catalogne, collaboration établie dans le cadre d'une *concertation thématique informelle*, régie par le CCETT, France Télécom.

1.2 Cadre du travail

L'objectif est ici de fournir brièvement le cadre technique de nos travaux, c'est-à-dire les caractéristiques que nous nous sommes imposées à la fois sur les données d'entrée, les outils proposés eux-mêmes et le domaine applicatif.

Tout d'abord, rappelons que les outils et les techniques que nous proposerons par la suite seront applicables à des documents vidéo uniquement, ou du moins à la partie vidéo des documents multimédia à indexer. Toute vidéo se traduisant concrètement en une série d'images, les moyens dont nous disposons et les méthodes que nous emploierons reposeront donc toutes sur des techniques de traitements d'images fixes ou de séquences. Et bien sûr, au final, l'indexation que nous réaliserons sera également une indexation d'images fixes ou de séquences. Ainsi, nulle utilisation ne sera faite du son, pour prendre un exemple simple, même si son étude couplée à celle de la partie vidéo a déjà prouvé qu'elle permettait une augmentation sensible de la robustesse et de la qualité des résultats obtenus [29].

Si l'ensemble des moyens utilisés par la suite relèvent du traitement d'images et d'un peu d'intelligence artificielle, aucun lien ne sera fait avec le domaine des bases de données et de leur gestion. Par conséquent, nous ne traiterons pas dans ce mémoire la formulation des requêtes utilisateurs à la recherche d'information, ni leur adéquation avec la structuration obtenue. Ceci étant, c'est certainement à partir de la structuration relationnelle, dernière étape de nos travaux, que l'adéquation avec les requêtes sera réalisée. Cette partie de la structuration constituera un point de départ naturel pour la construction de bases de données elles-aussi relationnelles, à inscrire au nombre des perspectives d'extension de notre travail.

Nous avons en outre fait le choix de travailler avec des séquences d'images couleur et non comprimées comme données d'entrée. Le choix de la couleur nous permet en effet une information plus riche que celle issue d'une même séquence mais en luminance uniquement. Quant aux données non comprimées, elles répondent à notre souci de rester le plus général possible, toutes les données disponibles à ce jour n'étant pas forcément comprimées. C'est

par exemple le cas de films extrêmement anciens, qui ne sont parfois même pas disponibles sous forme numérique. Nous reviendrons en temps et en heure sur les avantages des méthodes travaillant à partir de données comprimées, mais citons d'ores et déjà le principal qui consiste à réduire les temps de calcul. Or notre choix de n'élaborer que des outils simples permettra d'atteindre des rapidités comparables et à partir de données brutes.

Par ailleurs, si nous nous sommes imposés, du fait du champ très large de situations couvertes par l'indexation, de ne construire que des outils applicables à des documents vidéo quelconques, nous illustrerons nos résultats dans un domaine d'applications très précis, à savoir les journaux télévisés. Cette restriction n'en est en fait pas vraiment une dans la mesure où ce domaine reste malgré tout très général de par la grande diversité des sujets traités. Les informations télévisées constituent en outre la source d'information la plus réutilisée, et les journalistes télévisés représentent eux-mêmes les utilisateurs les plus nombreux de ces archives.

Enfin, dans un cadre technique plus général, notons que les outils dont nous visons la réalisation s'inscrivent dans une phase amont de la normalisation MPEG7, puisque les entités que nous obtiendrons comme résultats de la structuration (prise de vue, etc.) correspondent à la notion de descripteurs d'une vidéo tels qu'ils sont décrits dans la norme.

1.3 Plan et contenu de l'ouvrage

L'annexe A propose, sous la forme d'un lexique, un rappel de l'ensemble des définitions et du vocabulaire de l'indexation qui seront utilisés au cours de ce mémoire. Le lecteur trouvera par ailleurs dans l'annexe B une brève introduction à la morphologie mathématique et aux opérateurs nécessaires à l'élaboration de nos outils. La lecture préalable de cette deuxième annexe lui fournira donc toutes les notions indispensables à une bonne compréhension de notre propos. En plus d'un premier chapitre destiné à la définition complète de ce que nous mettons derrière le terme de structuration d'un document vidéo, ce mémoire se divise en trois parties principales, dédiées aux trois composantes justement de cette structuration, que seront la structuration linéaire temporelle, la structuration linéaire spatiale et la structuration relationnelle.

Partie I : Structuration linéaire temporelle. La structuration linéaire temporelle, que l'on peut brièvement décrire comme un découpage temporel du document vidéo en diverses entités allant d'un groupe de prises de vue à l'image, se subdivise elle-même en deux étapes : le découpage du fichier vidéo en entités de taille minimale la prise de vue, suivi du sous-découpage de ces prises de vue en morceaux plus petits et de l'extraction d'images clés, i.e. représentatives de leur contenu. Cette première partie est donc elle-même découpée en deux chapitres correspondant à ces deux étapes : le macro-découpage et le micro-découpage. Nous fournirons ainsi deux algorithmes de détection de transitions, effets de montage délimitant les prises de vue, puis plusieurs outils permettant d'extraire d'une prise de vue des sous-morceaux, comme par exemple des mouvements de caméra. Enfin nous développerons une technique de sélection d'images clés.

Partie II : Structuration linéaire spatiale. La deuxième partie poursuit la structuration linéaire d'un document vidéo. Une fois la segmentation temporelle terminée et les images clés sélectionnées, ces dernières donnent lieu à une structuration spatiale cette fois. Cette structuration spatiale correspondant directement à la segmentation des images et en l'extraction d'objets d'intérêt, nous proposerons donc là encore deux chapitres distincts. Le premier fournira une description très détaillée d'un processus de segmentation

morphologique d'images couleur. Le deuxième permettra de passer en revue divers outils d'extraction d'objets d'intérêt que peuvent être les visages, le texte, les incrustations, etc.

Partie III : Structuration relationnelle. Enfin notre troisième et dernière partie sera l'occasion de bâtir la composante relationnelle de notre structuration. Au contraire de la structuration linéaire, pour laquelle les entités successivement extraites possèdent la caractéristique d'être adjacentes, il s'agira dans cette dernière partie de mettre en évidence des relations pouvant exister entre des entités pas forcément voisines, ni de même type. Ces relations à caractère sémantique peuvent bien sûr être de tout ordre ; de ce fait nous n'en fournirons que le principe général et de nombreux exemples particuliers et applications.

Chapitre 2

Structuration d'un document

Ainsi que cela a déjà été formulé dans le chapitre introductif, les algorithmes et outils développés dans ce mémoire appartiennent essentiellement au domaine du traitement d'images et de séquences. Ils s'appliquent donc uniquement aux documents vidéo, ou du moins à la partie vidéo des documents à indexer et bien sûr aux films cinématographiques.

L'objet de la première partie de ce chapitre est toutefois d'expliciter le terme plus général de document, au travers de l'exemple du document vidéo. L'analyse du document vidéo fait en effet apparaître une structure hiérarchique, capable de représenter les composants d'un document au sens général. L'élaboration de cette structure est d'une importance capitale pour le processus d'indexation, mais également, dans le cadre d'une recherche dans une base de données, pour formuler des requêtes utilisateurs, ainsi que les réponses correspondantes du moteur de recherche. Seule la description de la structure du document vidéo fait l'objet de la deuxième partie de ce chapitre. Les algorithmes visant à son élaboration la plus complète possible constituent le cœur de ce mémoire et seront développés par la suite. Toutes les définitions énoncées dans ce chapitre, et par la suite dans le reste de ce mémoire, peuvent être retrouvées dans l'annexe A.

2.1 Qu'est-ce qu'un document vidéo ?

2.1.1 Fichier vidéo

Avant même de chercher à définir ce que l'on entend par document vidéo, il est nécessaire de faire la distinction entre ce terme et celui de fichier vidéo.

Par fichier vidéo, on entend le fichier physique contenant les données numériques, i.e. une succession d'images dans un format donné et quelques informations supplémentaires telles que le nombre total d'images, leur fréquence d'acquisition, leur format, leurs tailles, etc. Le terme de fichier vidéo recouvre donc la notion de données numériques brutes, directement issues du système d'acquisition. On parle alors aussi de séquence d'images. Par la suite nous donnerons une autre définition au terme de séquence, aussi, pour éviter les confusions, le mot séquence sera-t'il toujours suivi de l'attribut **d'images** pour évoquer un fichier vidéo, le terme seul de **séquence** étant réservé à un autre usage.

Le terme de document vidéo sous-entend, quant à lui, l'élaboration d'une nouvelle structure de données capable de transcrire l'organisation, ou plus exactement la hiérarchisation des images entre elles.

2.1.2 Document vidéo

Le document vidéo, en tant que nouvelle structure de données, est plus élaboré que le simple fichier vidéo. Sous cette notion de document vidéo, on regroupe en effet à la fois le fichier vidéo et la structure syntaxique, sémantique et hiérarchique selon laquelle les images sont organisées. Le terme général "syntaxique" se rapporte à l'aspect formel d'un langage, et à ses règles d'écriture (la syntaxe d'une phrase par exemple). Dans le cas du document vidéo, la structure syntaxique recouvre donc toutes les règles formelles de construction du document, dont un exemple est l'alternance prise de vue - transition. Au contraire, par sémantique, on entend tout ce qui est relatif au sens et à la signification des unités composant le document. Enfin l'adjectif hiérarchique rendra compte de l'organisation des unités extraites du document en plusieurs niveaux hiérarchiques les uns par rapport aux autres, tant dans le sens syntaxique que sémantique.

Cette structure n'étant pas directement accessible à partir du fichier vidéo, elle nécessite d'être extraite par une série de traitements, constituant ce que nous appelons le processus de structuration d'un document vidéo. La structure obtenue par un tel processus est construite à partir de toute information qu'il est possible d'extraire de l'image ou de la séquence d'images. Pour un document au sens plus général du terme, elle s'étend sans difficulté : au squelette fourni par la structuration de la partie vidéo du document se raccrochent toutes les informations disponibles, de toute nature et de toute provenance (du son, des scripts, etc.).

Au niveau du document, la structure obtenue est donc beaucoup plus complexe et plus riche que le simple fichier d'images. C'est sur cette structure, et non sur le fichier vidéo lui-même, que seront en outre menées les requêtes dans un processus de recherche d'information.

Dans le cadre restreint des documents vidéo, plusieurs structures et, par là, plusieurs structurations, plus ou moins intuitives, ont été proposées. Nous les analysons dans la section suivante.

2.2 Quelles entités pour quel découpage ?

La structuration la plus intuitive d'un document vidéo correspond sans doute à un découpage temporel linéaire du fichier vidéo en entités individuelles de plusieurs niveaux, correspondant toutes à une certaine unité sémantique.

Dans une telle structuration, les entités de plus haut niveau sont constituées du regroupement d'une ou plusieurs entités de niveau immédiatement inférieur, et ainsi de suite, jusqu'à obtenir un découpage en unités atomiques, i.e. qu'il est impossible de diviser. De nombreux travaux [59, 60, 61, 29] proposent ainsi deux ou trois niveaux de hiérarchie en séquences, scènes et prises de vue, la prise de vue étant l'entité atomique dans ce cas. Si la définition de cette dernière entité est constante dans la littérature, et ce depuis l'apparition, à l'époque des films muets, de la théorie du montage vidéo [40], comme étant l'ensemble des images comprises entre deux arrêts de caméra, les définitions des scènes et séquences, à caractère sémantique, différent.

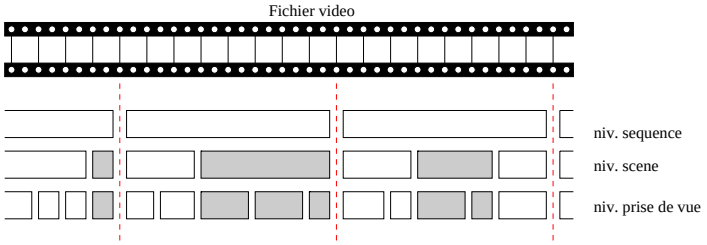
Davenport *et al.* [29] considèrent par exemple la séquence comme une collection de prises de vue formant une "unité naturelle", résultant d'une certaine continuité sur les plans temporel, spatial, perceptuel, etc. Une telle entité *séquence* se caractérise essentiellement par le fait qu'elle n'est plus perçue comme une succession de prises de vue, mais comme une unité à part entière.

Cette définition regroupe les deux notions de scène et séquence des travaux de Hjelsvold *et al.* [59, 60, 61] et Hammoud *et al.* [52], dans lesquels la scène ne possède une certaine continuité que selon le critère spatio-temporel, et la séquence selon un autre critère (même unité d'action).

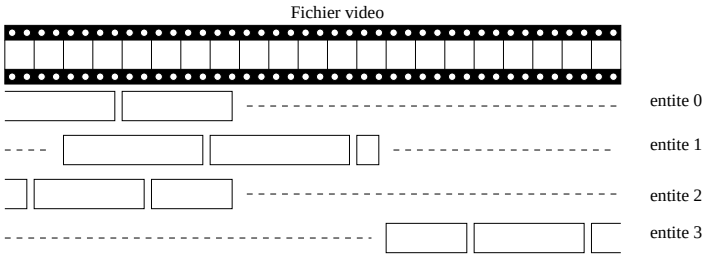
Ces deux dernières définitions sont par ailleurs les mêmes que celles utilisées dans le domaine du montage vidéo [10, 24].

Hjelsvold *et al.* proposent en outre un quatrième niveau de hiérarchie, les "unités composées" (*compound units*) correspondant à des ensembles de séquences en relation, par un quelconque critère de similarité.

En parallèle de cette structuration entièrement linéaire, Seyrat *et al.* [86] proposent un autre découpage basé sur des objets visuels ou sonores, dans lequel les diverses entités peuvent se recouvrir, comme nous l'illustrons dans la figure 2.1, partie (b).



(a) structuration linéaire classique



(b) structuration linéaire avec recouvrement

FIG. 2.1: Représentations de deux structurations linéaires différentes : (a) structuration linéaire classique, sans recouvrement des entités d'un même niveau hiérarchique ; (b) structuration avec recouvrement possible des entités.

L'extraction de telles entités répond alors à des critères plus sémantiques comme la présence d'un même objet, d'une même personne, la présence de scènes extérieures ou intérieures, de jour ou de nuit. Une telle structuration est évidemment plus complexe à obtenir et ne tient compte en aucune façon du montage vidéo effectué lors de la création du fichier vidéo.

Cette deuxième forme de structuration est plus proche également d'une structure relationnelle, telle que celle que nous utiliserons dans ce mémoire. Cette dernière tend à représenter des relations entre les diverses entités de la structure linéaire précédemment extraite [52].

En résumé, trois structures sont donc disponibles dans la littérature, deux sont issues d'un découpage linéaire du fichier vidéo, soit avec recouvrement des entités, soit sans ; la troisième est une structure relationnelle.

Notons dès à présent l'ambivalence entre les deux types linéaire et relationnel : dès lors que l'on crée des scènes ou séquences par regroupement de prises de vue, on établit une certaine forme de relations sémantiques entre entités voisines, relations qui entrent à part entière dans

la structure relationnelle du document vidéo. Cette deuxième structure va plus loin en ce sens qu'elle tient compte également d'autres relations établies entre des entités non plus adjacentes mais situées en divers points du fichier vidéo.

Ces trois structures étant à présent exposées, nous détaillons dans la suite de ce chapitre la ou plutôt les structurations que nous avons adoptées pour bâtir le squelette du document vidéo.

2.3 Structuration d'un document vidéo

Face aux deux classes de modèles de structuration d'un document vidéo, i.e. linéaire ou relationnelle, il apparaît évident que ces deux types de structures sont essentiels à une représentation complète de l'organisation du document. Nous avons donc cherché dans ce mémoire à bâtir tout d'abord une structure linéaire de type découpage temporel et spatial, détaillée dans la section 2.3.1, puis à établir les prémices d'une structure relationnelle entre les entités issues de la structure linéaire (cf. section 2.3.2).

Toute autre information issue de l'indexation du document viendra alors se greffer sur le squelette que constitue en fait cette double structure. En ce sens son établissement fait lui-aussi partie du processus d'indexation : il s'agit d'une première étape essentielle, sans laquelle l'organisation et le stockage des autres informations et, par la suite, la recherche sur le document indexé, ne peuvent avoir lieu.

2.3.1 Structuration linéaire

Comme nous l'avons déjà évoqué en introduction de cette section, la structuration linéaire proposée ici s'organise autour d'un découpage temporel de la séquence d'images et d'un découpage spatial des images. Le découpage temporel est de deux types : on distinguera ainsi un macro-découpage et un micro-découpage.

2.3.1.1 Segmentation temporelle : macro-découpage et micro-découpage

La structuration la plus intuitive de toutes celles détaillées dans la section précédente est sans aucun doute la structuration en prises de vue, scènes et séquences [24, 10, 61, 29]. Donnons tout d'abord la définition adoptée pour ces trois entités. Pour chaque terme défini, on donne entre parenthèses son équivalent anglais, lorsqu'il existe.

Définition 1. Prise de vue ou plan(*shot*) *Une prise de vue est une série d'images acquises par une seule caméra, d'un seul tenant, et n'ayant donné lieu à aucun montage. Cet ensemble d'images représente une action continue dans le temps et dans l'espace.*

Définition 2. Scène (*scene*) *Une scène est constituée d'un ensemble de prises de vue ayant une même unité de lieu (divers points de vue par exemple).*

Définition 3. Séquence ou segment (*sequence*) *Une séquence regroupe diverses prises de vue et scènes ayant une même unité de sujet (par exemple un reportage).*

On soulignera à ce stade la différence de notion entre l'entité **séquence** et la notion de **séquence d'images**, comme autre dénomination du **fichier vidéo** (cf. section 2.1.1).

Dans un premier temps, nous adoptons cette structuration afin d'effectuer un découpage temporel du document à indexer.

Parmi les trois entités retenues, et d'après sa définition, la prise de vue joue le rôle d'entité de base, au sens atomique. Une prise de vue correspond en effet à la collection d'images contenues entre deux arrêts de la caméra. Au niveau syntaxique, on atteint, avec cette notion, un stade particulier du découpage temporel possible d'un document vidéo. Le passage du fichier vidéo entier à une structure en séquences, scènes et prises de vue, constitue ce qu'on appelle le macro-découpage temporel.

Définition 4. Macro-découpage temporel *Par macro-découpage temporel, on entend la partie du découpage temporel d'un fichier vidéo ne descendant pas au-dessous de la prise de vue, qui est alors, pour cette étape, l'entité atomique.*

La définition de la prise de vue telle qu'elle est donnée ne contient la notion d'atomicité que dans le cadre du montage d'un document vidéo à partir de séries d'images acquises d'un seul tenant par une caméra. S'il n'y a pas d'arrêt de la caméra lors de l'acquisition d'une prise de vue, on peut cependant envisager que la caméra ait un ou plusieurs mouvements différents dans une même prise de vue. Par exemple, il est tout à fait envisageable et même courant qu'une prise de vue débute sur un champ très large de la scène, puis soit restreinte par un zoom à un champ d'intérêt plus petit. De même, une prise vue peut débuter par une scène fixe puis se poursuivre par un mouvement panoramique. Dans ces deux cas, l'extraction de l'information supplémentaire contenue dans le mouvement de la caméra commence par un sous-découpage, toujours temporel, des prises de vue elles-mêmes. Ces morceaux de prises de vue peuvent également être cohérents de par la présence de bandeaux d'incrustation de texte (cf. journaux télévisés), ou de mouvement d'objets. En ce sens, tout découpage d'une prise de vue en morceaux plus petits, ayant une certaine cohérence syntaxique ou sémantique, constitue le micro-découpage temporel du document vidéo.

Cette fois-ci, l'unité atomique de ce découpage est l'image. Le choix de certaines images particulières d'un morceau de prise de vue, comme images représentatives du contenu informationnel, constitue alors l'étape finale du micro-découpage temporel.

Définition 5. Micro-découpage temporel *Le micro-découpage temporel consiste en un découpage temporel des prises de vue d'un document vidéo en sous-unités allant jusqu'à l'entité atomique constituée par l'image clé.*

La succession de ces deux découpages temporels est représentée de façon schématique dans la figure 2.2.

2.3.1.2 Segmentation spatiale

La logique de la structuration linéaire entreprise dans cette section se poursuit par un découpage, cette fois-ci spatial, des images clés, ou images caractéristiques, retenues pour chaque entité issue du découpage temporel. Ce découpage spatial n'est autre qu'une segmentation d'images, à ceci près que les régions extraites doivent également posséder un contenu sémantique propre. En ce sens on cherche à obtenir une segmentation en objets.

Ainsi, pour une scène représentant un personnage assis dans l'herbe sous un arbre, les trois objets principaux à extraire sont le personnage, l'arbre et la zone d'herbe. Bien sûr, d'autres segmentations plus fines, aboutissant par exemple à un découpage du personnage en plusieurs régions (sa tête et son corps, etc.), sont également acceptables dans un contexte d'indexation : chaque objet a son propre contenu sémantique. Par contre, si la zone d'herbe est absolument homogène en texture et en couleur, on cherchera à éviter une partition de cet

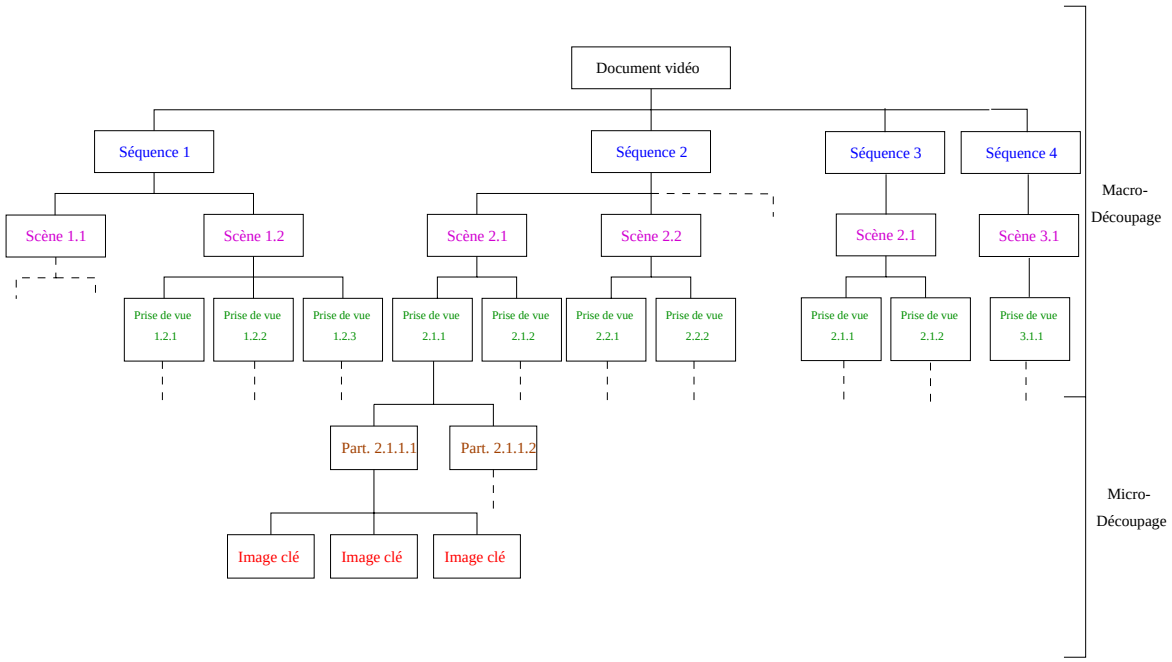


FIG. 2.2: Découpage temporel d'un document vidéo : macro-découpage et micro-découpage.

objet en plusieurs régions, dans le cas où le découpage obtenu n'apporte aucune information sémantique supplémentaire.

Cette segmentation spatiale peut, au même titre que la segmentation temporelle du document, être représentée de façon hiérarchique (cf. figure 2.3).

2.3.1.3 Segmentation spatio-temporelle

En parallèle des segmentations temporelle et spatiale, il est envisageable de combiner les deux informations de façon à améliorer à la fois le découpage temporel et le découpage spatial.

Ainsi, des segmentations spatiales proches d'une image à l'autre permettent de renforcer la continuité temporelle ; au contraire deux segmentations totalement différentes indiquent plutôt une rupture de modèle, donc une transition. Ceci est cependant conditionné à l'obtention pour chaque image d'une segmentation stable temporellement, ou plus exactement entre deux images de contenu proche.

Dans le cadre des techniques de segmentation d'images, le lien entre les informations temporelles et spatiales est plutôt employé dans le sens inverse : on utilise la continuité temporelle pour justement obtenir une segmentation spatiale stable dans le temps, par exemple par des techniques de projection de la segmentation d'un instant t à l'instant $(t+1)$, i.e. les régions de la segmentation à t sont directement projetées sur l'image à $(t+1)$ et servent de point de départ à la nouvelle segmentation.

Que ce soit donc l'utilisation du découpage temporel pour reforcer la segmentation spatiale ou l'inverse, la combinaison des deux informations doit permettre de renforcer la structure du document vidéo. Dans le cadre de ce mémoire, nous ne rentrerons pas plus avant dans l'explication de telles techniques.

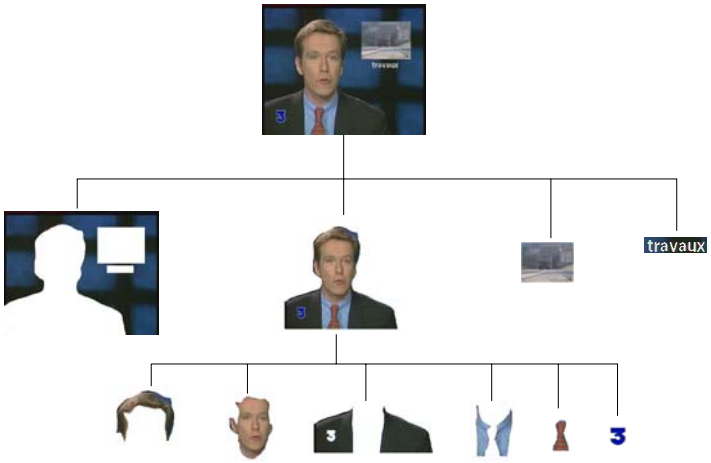


FIG. 2.3: Découpage spatial : segmentation en objets.

2.3.2 Structuration relationnelle

Après la réalisation du découpage linéaire, somme toute assez simple à formuler, vient l'établissement d'une autre forme de structuration, non linéaire : il s'agit à ce stade d'établir des relations entre les entités précédemment extraites (cf. section 2.3.1). Deux entités en relation satisfont aux deux propriétés suivantes : elles ne sont pas forcément du même type, i.e. une prise de vue peut être en relation avec un objet ou avec une scène ; elles ne sont pas toujours adjacentes temporellement.

Par "être en relation", on entend l'existence d'un lien sémantique quelconque (par exemple un même personnage intervient dans deux scènes différentes) entre les deux entités. Avec cette définition, il est même possible d'étendre la notion de relation aux documents eux-mêmes.

L'ensemble de relations qu'il est possible de construire pour un document donné est appelé graphe de relations.

Définition 6. Graphe de relations *Un graphe de relations correspond à la représentation de l'ensemble des relations sémantiques ou syntaxiques, de tout ordre, qui existent entre deux entités quelconques d'un même document.*

Notons que la notion de graphe de relations peut bien sûr être étendue à la base de données entière des documents vidéo, puisque des relations peuvent être établies entre des éléments de documents vidéo distincts.

Ce graphe constitue la structuration relationnelle du document et c'est lui qui permet, au travers des liens établis, de fournir des réponses multiples à une requête donnée, dans un processus de recherche d'information dans la base de documents indexés. On peut en effet avoir établi, par un moyen dont nous ne discuterons pas ici, qu'une entité donnée (appelons-la A) est une réponse possible à la requête utilisateur, et fournir des réponses supplémentaires sous la forme des entités en relation avec A, et avec un degré d'adéquation dépendant de la nature de la relation entre chaque entité et A. D'autre part, il est possible d'imaginer un renforcement de l'adéquation entre des entités répondant au processus de recherche et une requête, si ces entités sont en relation.

La construction concrète de cette seconde structure se poursuit encore une fois, et du même coup se complexifie, avec toute nouvelle information sémantique sur le document (son, scripts,

auteurs, etc.). A priori les outils mis en jeu, du moins au niveau du traitement d'images, sont alors d'un niveau plus élevé que ceux servant à l'établissement de la structure linéaire.

Enfin, si l'intérêt de la structure relationnelle réside certainement dans le fait que les entités mises en relation peuvent ne pas être adjacentes, alors que la structure linéaire effectue par définition une décomposition en éléments voisins, il est cependant évident que ces deux structures sont liées. Ainsi :

- le simple fait de savoir que deux entités sont voisines (structure linéaire) peut déjà contenir une information de nature relationnelle ;
- des liens établis lors de la construction du graphe de relations (structure relationnelle) peuvent permettre de revenir et de valider ou d'invalider la structure linéaire établie. Des illustrations de ce dernier point sont fournies dans le chapitre 8, section 8.2.

Par cette remarque, on retrouve toute l'ambiguïté et l'imbrication des techniques d'intelligence artificielle de type "bottom-up" ou "top-down".

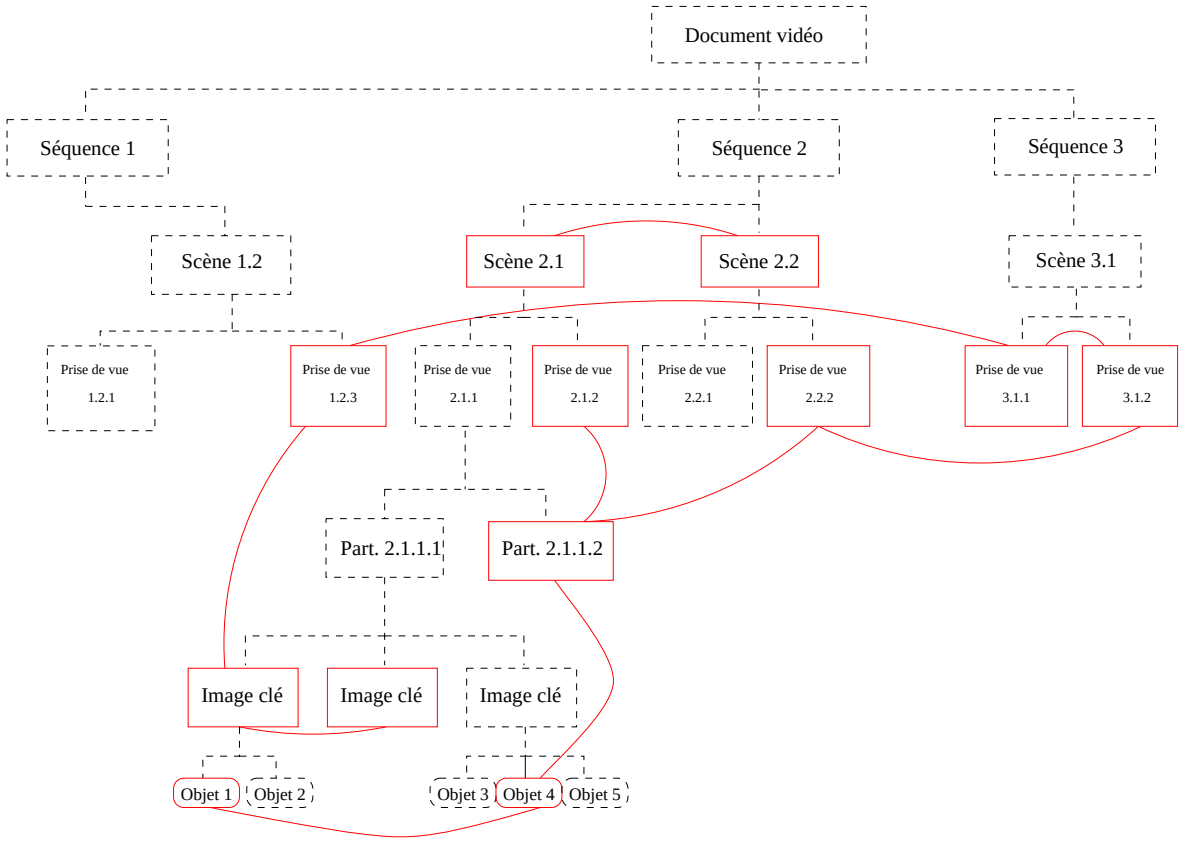


FIG. 2.4: Graphe de relations d'un document vidéo.

Nous présentons un exemple de ce deuxième réseau constitué par le graphe de relations dans la figure 2.4. Il s'agit véritablement d'une structure de graphe au sens mathématique, dans laquelle les nœuds sont constitués par les diverses entités d'un document, et un arc entre deux nœuds a et b est orienté et correspond à la relation "a est en relation avec b" (cette relation peut être symétrique ou non). Les arcs d'un tel graphe sont de plus multivalués et leurs valeurs correspondent aux types de relations existant entre les deux nœuds correspondant.

Les notions de structures linéaire et relationnelle étant établies, nous proposons en quelques lignes, dans le paragraphe suivant, une description de la façon dont ces deux structures sont organisées sur le plan informatique, dans le cadre de la réalisation pratique des outils de structuration.

2.4 Structure informatique des données

Ce court paragraphe a pour but de présenter dès à présent la structure informatique concrète dans laquelle les informations extraites dans ce mémoire seront stockées. Cette structure comporte une double organisation représentant respectivement les découpages linéaire et relationnel.

La structure de document vidéo contient donc tout d'abord une série d'informations relatives à la notion de fichier vidéo, i.e. aux données brutes constituées par les images. Parmi ces informations, se trouvent les tailles d'images en X et Y, le nombre d'images contenues dans le fichier vidéo, la fréquence d'acquisition des images à partir de la source et leur format de stockage. Ceci est regroupé dans la partie **Data** de la structure de document vidéo :

DocData
Largeur_Image
Hauteur_Image
Nombre_Images
Fréquence_Acquisition
Format_Image

La partie structure véritablement, notée **Struc**, du document vidéo est conçue simplement sous la forme de quelques informations telles que numéros des images de départ et de fin du document vidéo, nombre de séquences, pointeur sur la liste des séquences. Chaque séquence est elle-même de la même forme, à ceci près qu'elle contient un nombre de scènes et un pointeur vers la liste chronologique de ces scènes. Enfin chaque scène est décomposée de la même façon à partir de prises de vue.

DocStruc
Numéro_Image_Début
Numéro_Image_Fin
Nombre_Séquences
Liste_Séquences

Séquence
Numéro_Image_Début
Numéro_Image_Fin
Nombre_Scènes
Liste_Scènes

Scène
Numéro_Image_Début
Numéro_Image_Fin
Nombre_PrisesDeVue
Liste_PrisesDeVue

Au final, on obtient donc la représentation UML (*Unified Modeling Language*)[1] présentée dans la figure 2.5.

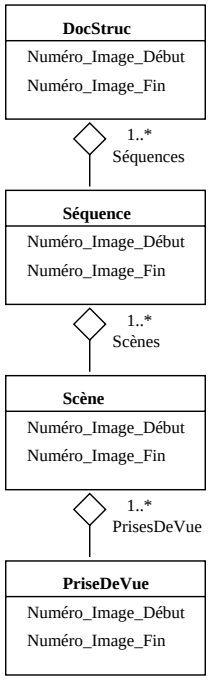


FIG. 2.5: Représentation de la hiérarchie Document Vidéo, Séquence, Scène et Prise de vue.

Nous détaillons à présent la structure de prise de vue de façon plus précise. Il est, de façon évidente, possible et même souhaitable de rajouter certains champs de cette structure aux structures de scènes et de séquences dont la base a été présentée ci-dessus.

De la même façon que pour les structures de scène et séquence, la structure de prise de vue contient les numéros des images de début et de fin de prise de vue. On rajoute à ces deux premières informations le nombre d'images clés qui ont été extraites ainsi qu'un tableau des index de ces images clés. Les prises de vue (tout comme les scènes et séquences) sont organisées en double liste chaînée ; aussi deux pointeurs, l'un vers la prise de vue précédente et l'autre vers la prise de vue suivante, sont-ils présents. Enfin, et ceci constitue cette fois-ci la base de la structure relationnelle du document vidéo, le nombre de relations établies pour la prise de vue courante et un pointeur vers ces dernières sont ajoutés. Une liste d'attributs représentant des informations variées propres à la prise de vue est également présente à ce niveau. On obtient ainsi la structure complète suivante, dont on remarquera qu'elle reste extensible, i.e. tout nouveau champ représentant une information différente peut être simplement ajouté aux

champs existants :

PriseDeVue
Numéro_Image_Début
Numéro_Image_Fin
Nombre_ImagesClés
Liste_ImagesClés
Nombre_Relations
Liste_relations
PriseDeVue_Précédente
PriseDeVue_Suivante
Liste_Attributs

Liste_Attributs
Type
Présence_Changement
Nature_Changement
Présence_Personne
Présence_Flash
...

Parmi les attributs possibles, certains sont optionnels tels que la présence de changement, de flash, etc., et d'autres doivent être présents pour une définition correcte de l'entité. C'est le cas du champ *type* permettant de classifier les différents types de prises de vue. Les types considérés sont en effet très divers au niveau informatique, puisque sont regroupés, sous l'étiquette *prise de vue*, aussi bien les prises de vue au sens de la définition 1 que les transitions, ou les morceaux de prises de vue.

On considère ainsi les transitions comme une classe de prises de vue particulière résultant du montage vidéo, les coupures étant alors des prises de vue de longueur nulle, ce qui se représente dans notre structure de données par : **start = end + 1**. La durée est alors de : **end - start + 1 = 0**. Au niveau informatique, l'étiquette prise de vue correspond donc plus à la notion de "suite d'images consécutives" ; quant à la notion de prise de vue au sens de la définition 1, elle correspond à l'ensemble des *suites d'images* contenues entre deux *suites d'images* de type transition.

On dresse alors la liste des types de prises de vue suivante :

- **normale** : prise de vue, ou morceau de prise de vue, sans caractéristique particulière ;
- **coupure** : transition de type coupure ;
- **fondu** : transition de type fondu ;
- **mouvement** : prise de vue, ou morceau de prise de vue, correspondant à un mouvement de caméra ;
- **flash** : morceau de prise de vue correspondant à un flash ;
- etc.

La structuration linéaire apparaît donc dans la succession en ordre chronologique de ces diverses prises de vue, scènes et séquences. Quant à la structure relationnelle, outre les pointeurs vers l'ensemble des relations extraites pour un élément donné, elle nécessite une structure de données supplémentaire, permettant véritablement d'obtenir la notion de graphe de relations

comme structure à part entière et parallèle à la structuration linéaire. Ce graphe de relations se symbolise en réalité par un ensemble de composantes connexes maximales d'entités en relation (deux composantes connexes maximales distinctes n'ont aucune intersection, i.e. aucune entité commune) dont la structure est la suivante :

Composante_Connexe_Maximale
Nombre_Relations Liste_Relations Contenu_Sémantique

Outre un pointeur vers la liste des éléments en relation, cette structure contient toutes sortes d'informations sur le contenu sémantique de la composante connexe. Chaque relation contient alors des informations sur la nature propre de l'entité pointée (de type prise de vue, objet, etc.) et sur son contenu sémantique et son degré d'adéquation à la composante.

Relation
Entité Type_Relation Adéquation

En particulier les notions de **Séquence**, **Scène**, **Prise de vue** impliquent de façon intrinsèque l'existence de relations : deux prises de vue sont en effet déjà **en relation** de par leur appartenance à une même scène, par exemple.

Les grandes lignes de l'organisation informatique de la structure d'un document vidéo, tant sous la forme linéaire que relationnelle, étant établies, nous concluons ce chapitre sur l'organisation de la suite de ce mémoire.

2.5 Conclusion et mise en pratique

Nous venons de définir les entités constituant la structure linéaire d'un document, les relations existant entre ces entités donnant lieu à la construction d'un graphe relationnel. C'est à cette double structure que nous désirons aboutir comme support de la représentation sémantique du document.

Ce mémoire se poursuit par la présentation des outils élaborés dans le but de construire la structure linéaire, temporelle (partie I) et spatiale (partie II), et le graphe relationnel de tout document vidéo (partie III).

Ainsi, les chapitres 3 et 4 sont consacrés à la construction des outils de découpage temporel des séquences en prises de vue, et des prises de vue en images clés. La segmentation spatiale des images clés extraites est elle-même détaillée dans le chapitre 5, suivie d'un panel d'outils d'extraction d'objets particuliers des images clés dans le chapitre 6. Enfin, le chapitre 7 propose des outils de construction du graphe relationnel.

Soulignons d'ores et déjà que le niveau de découpage en prises de vue, de même que l'extraction des images clés et leur segmentation, étapes non triviales, seront cependant obtenus et concevables relativement aisément en comparaison de la structuration en scènes et séquences et de l'établissement de relations, qui demandent quant à eux des outils de plus haut niveau.

Première partie

Structuration linéaire temporelle

Chapitre 3

Macro-découpage : du fichier vidéo aux prises de vue

Le processus d'indexation d'un document vidéo consiste à la fois à extraire puis structurer toute l'information disponible dans ce document, ces deux étapes étant étroitement liées. Il a été établi au chapitre 2 que la structuration de l'information pouvait être de deux types : soit linéaire, soit relationnelle. La structuration linéaire, première étape naturelle, peut en outre être divisée en deux parties : un macro-découpage temporel s'arrêtant à l'entité atomique *prise de vue*, et un micro-découpage, toujours temporel et permettant cette fois-ci d'aller au-delà de la prise de vue, jusqu'aux images clés. La construction du macro-découpage fait l'objet de ce chapitre.

3.1 Introduction

3.1.1 Analyse du problème

Au chapitre précédent, la correspondance entre la partie linéaire de la structuration et un découpage temporel en morceaux élémentaires du document a été soulignée. Ainsi, dans une première étape, tout document vidéo se décompose chronologiquement en une succession de séquences ou segments, scènes et prises de vue (cf. paragraphe 2.3.1). Parmi ces trois éléments, la prise de vue possède la caractéristique supplémentaire d'être l'élément de base, i.e. l'élément atomique en terme de montage, si on reprend la définition 1.

A partir de cette définition, il est possible d'expliciter la notion de transition entre deux prises de vue successives :

Définition 7. Transition *On appelle transition tout effet de montage qui permet de passer d'une prise de vue à une autre. Il s'agit donc d'une série artificielle d'images ajoutée lors du montage par un opérateur.*

De cette dualité prise de vue - transition, il apparaît qu'il est équivalent, pour extraire une prise de vue d'un document vidéo, de détecter les transitions encadrant cette prise de vue. Cette détection des transitions est d'autant plus aisée qu'elles apparaissent de manière explicite comme des ruptures de modèle, plus ou moins brusques.

L'extraction de scène et de séquence faisant appel à d'autres informations d'un plus haut niveau syntaxique, nous proposons donc uniquement dans ce chapitre un premier découpage

temporel en prises de vue, par détection des transitions. Quelques pistes permettant le regroupement de plusieurs prises de vue successives en scènes ou en segments feront l'objet du chapitre 7.

Du fait de la grande diversité des transitions existantes, deux algorithmes s'appliquant à deux catégories distinctes de transitions sont proposés dans les sections 3.2 et 3.3, ainsi qu'une technique de fusion des deux ensembles de résultats obtenus (cf. section 3.4).

Mais avant de décrire plus précisément les algorithmes de détection de transitions mis en place, une description et une classification détaillées des transitions existantes est proposée. De cette étude découlent les choix que nous avons effectués pour caractériser et bâtir nos algorithmes.

3.1.2 Classification des différentes transitions

Avec l'avènement des techniques de montage numérique, de multiples transitions, toutes plus sophistiquées les unes que les autres, ont été créées, rendant d'autant plus difficile leur détection "après montage". Devant leur grande diversité, il devient alors nécessaire de les classifier en fonction de leurs caractéristiques propres, ceci afin de déterminer sur quels critères nous fonderons nos algorithmes de détection.

Avant de détailler plus avant la classification proposée ici, quelques définitions, remarques et notations générales sont introduites.

On notera ainsi toujours la prise de vue avant transition par (I) et la prise de vue après transition par (J). Une image prise à l'instant t dans la prise de vue (I) est alors notée I_t. Du fait du caractère numérique de nos données (i.e. les images), précisons que le temps est une variable discrète et qu'elle sera exprimée en nombres d'images.

La transition entre les prises de vue (I) et (J) a lieu entre les instants t_0 et t_1, soit entre les images I_{t_0} et J_{t_1}. On appelle durée totale de la transition l'intervalle de temps ouvert $]t_0, t_1[$. Entre les instants t_0 et t_1, les images de la transition, notées T_t avec $t \in]t_0, t_1[$, sont obtenues par combinaison des images des prises de vue (I) et (J) dans l'intervalle $[t_0, t_1]$. En ce sens, une transition peut elle-aussi être considérée comme une prise de vue particulière, au sens succession d'images, et on peut la noter (T).

Enfin, si on se place cette fois-ci à l'échelle d'un pixel p d'une image donnée, et si on le suit au cours du temps à travers la transition (T), ce pixel passe par trois états : p appartient à la prise de vue (I), puis p est dans la transition proprement dite pendant une durée qui lui est propre, inférieure ou égale à la durée totale de la transition, enfin p appartient à la prise de vue (J). On définit alors la notion d'état de transition :

Définition 8. Etat de transition *L'état de transition d'un pixel p correspond à la situation particulière de ce pixel qui est modifié du fait de la transition. Cet état de transition du pixel possède une durée propre pendant laquelle p est modifié suivant une transformation donnée, modélisant la transition.*

Dans l'intervalle $]t_0, t_1[$, tous les pixels passent par l'état de transition, mais tous ne sont pas forcément en même temps dans cet état. La succession ou l'ordre dans lequel les pixels passent par l'état de transition est appelé modèle géométrique de la transition. Des exemples de tels modèles sont étudiés par la suite.

Définition 9. Modèle géométrique *Le modèle géométrique peut se définir comme l'ordre dans lequel les pixels passent par l'état de transition. Mathématiquement, si on note G le*

modèle géométrique d'une transition :

$$G = \{G_t \,|t \in]t_0, t_1]\,\} \tag{3.1}$$

où G_t est un sous-ensemble des pixels de l'image à l'instant t. G_t peut être représenté comme une image binaire, les pixels à 1 correspondant aux pixels modifiés à l'instant t par rapport à l'instant $(t-1)$. Chaque ensemble G_t correspond donc à l'indicatrice de l'ensemble de définition de la fonction "état de transition", à l'instant t.

Le modèle géométrique est donc une autre des caractéristiques d'une transition donnée. La transformation subie par chaque pixel (i.e. par sa couleur ou son niveau de gris) dans l'état de transition en est une autre. En outre la durée de la transition totale (qui correspond à l'intervalle de temps total nécessaire au passage par l'état de transition de tous les pixels de l'image) et la durée de l'état de transition pour chaque pixel particulier sont également à rajouter aux caractéristiques, ce qui porte leur nombre à quatre, avec l'état de transition.

Afin de définir entièrement l'état de transition des pixels, nous introduisons à présent la notion de masque de transition répondant à la définition suivante :

Définition 10. Masque de transition *Le masque de transition M_t entre deux images successives d'un document vidéo est une image à niveaux de gris, qui traduit le lieu et la valeur des changements intervenant entre les deux images. Le niveau de gris de chaque pixel de l'image correspond à une valeur de dissemblance entre les instants $(t-1)$ et t. Dans le cas d'une transition parfaite entre deux prises de vue fixes et sans bruit, le lieu des points du masque de transition ayant une valeur de dissemblance non nulle à l'instant t correspond à l'ensemble G_t de la définition du modèle géométrique (cf. définition 9).*

Par la suite, on parlera souvent de masque de transition binaire pour signifier par extension la notion de modèle géométrique. Un seuillage du masque de transition M_t pour ne conserver que les points non nuls donne en effet, une approximation du modèle géométrique G_t, approximation qui devient exacte, dans le cas idéal d'une transition entre deux prises de vue fixes et sans bruit.

Ces notions étant établies, nous proposons de classifier l'ensemble des transitions existantes suivant deux axes (cf. figure 3.1) desquels on extrait quatre groupes principaux :

- **groupe 1** : ensemble des transitions ne possédant pas de modèle géométrique (tous les pixels sont traités en même temps) et de durée nulle (i.e. $t_1 = t_0 + 1$). De ce dernier point, il découle que la durée de l'état de transition pour chaque pixel est également nulle, et qu'il n'y a pas de modification des couleurs suivant un schéma propre à la transition. Cette catégorie regroupe l'ensemble des coupures.
- **groupe 2** : ensemble des transitions possédant un modèle géométrique et d'une durée totale $(t_1 - t_0)$. Par contre l'état de transition des pixels est de durée nulle. Ceux-ci passent directement d'une prise de vue à une autre, sans subir de modification du fait de la transition proprement dite. Le balayage horizontal est un exemple classique de ce type de transitions.
- **groupe 3** : ensemble des transitions possédant un modèle géometrique, une durée totale de $(t_1 - t_0)$ et un modèle de modification de la couleur d'un pixel dans l'état de transition. Cette modification a une durée non nulle, propre à chaque pixel. Cette catégorie regroupe l'ensemble des transitions les plus sophistiquées.
- **groupe 4** : ensemble des transitions sans modèle géométrique (toute l'image est affectée en même temps), d'une durée totale de $(t_1 - t_0)$ non nulle. Ces transitions ont un modèle

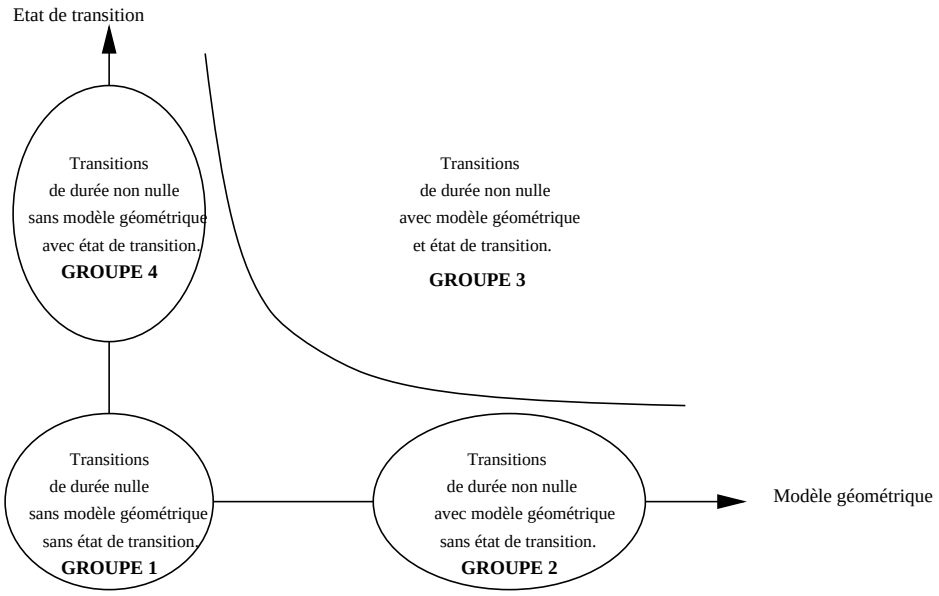


FIG. 3.1: Classification des transitions de montage en fonction de deux critères : la présence d'un modèle géométrique et celle d'un état de transition.

de modification des couleurs dans l'état de transition, et la durée du passage dans l'état de transition pour chaque pixel est égale à la durée totale de la transition. Dans cette catégorie, l'exemple le plus connu est celui du fondu enchaîné.

Cette classification repose donc à la fois sur les quatre caractéristiques d'une transition extraites en début de paragraphe. Aux deux extrêmes de cette représentation linéaire de l'ensemble des transitions, les groupes 1 et 4 se distinguent de part leur antériorité historique, leur fréquence d'utilisation dans les documents vidéo et leurs différences de modèles. Nous proposons d'étudier à présent de façon plus approfondie les exemples de transitions de ces deux groupes que sont les coupures et les fondus enchaînés.

Les transitions intermédiaires n'étant que des étapes progressives pour aller du modèle de la coupure à celui du fondu enchaîné, leur description en sera alors aisément dérivée par la suite.

3.1.2.1 Coupure

La transition de type coupure est la plus simple possible :

Définition 11. Coupure *Une coupure est l'effet de transition qui consiste à accoler deux prises de vue successives.*

Aucun effet de montage n'est rajouté. De là découlent donc les propriétés suivantes :

- la coupure est de durée totale nulle, il s'agit de ce que l'on appelle une transition brusque ; la transition n'existe pas en soi, ou plutôt elle n'existe que par son absence ;
- aucun modèle géométrique de transition n'est disponible, tous les pixels de l'image sont affectés en même temps ;
- il n'y a pas modification des couleurs des pixels dans l'état de transition ;

– de ce dernier point il découle que la durée de passage dans l'état de transition pour chaque pixel est nulle.

Avec le formalisme mathématique dont nous disposons, une coupure entre les prises de vue (I) et (J) est de durée nulle :

$$t_1 = t_0 + 1 \tag{3.2}$$

et se représente par la succession d'images :

$$I_{t_0-1}, I_{t_0}, J_{t_0+1}, J_{t_0+2} \tag{3.3}$$

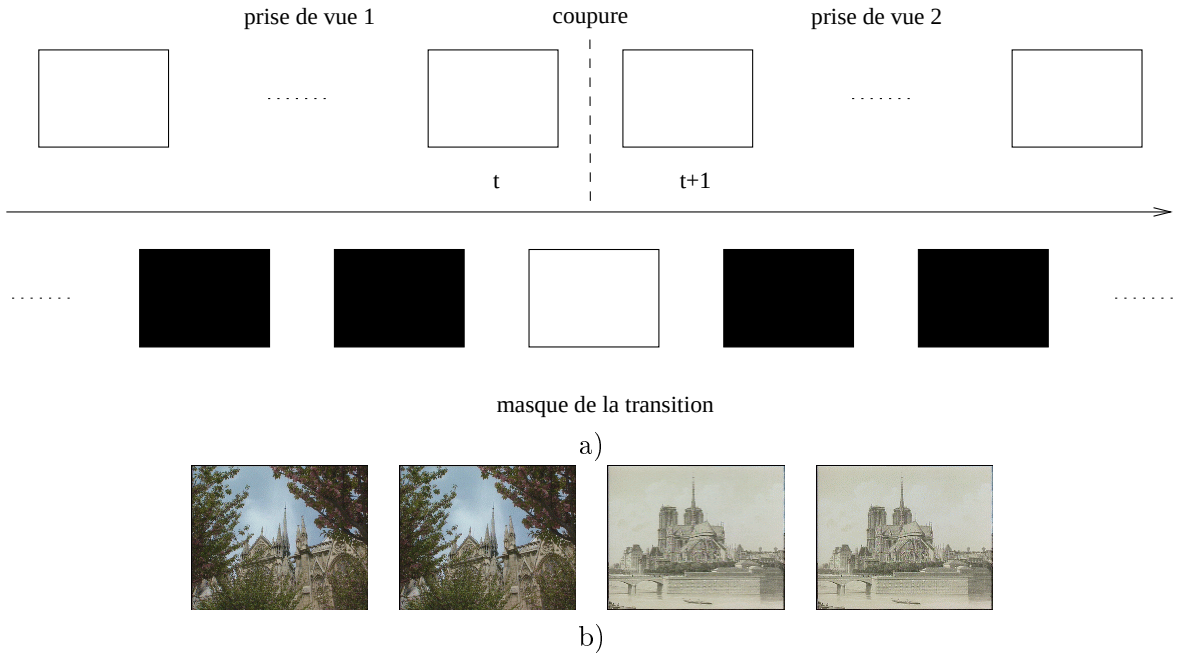


FIG. 3.2: Représentation d'une coupure : a) modèle du masque de transition. Au cours du temps le masque traduit les changements de prises de vue. b) exemple d'une coupure réelle.

Ce modèle est illustré dans la figure 3.2.

Idéalement, si on considère deux prises de vue fixes (i.e. caméra immobile et scène fixe sans objet en mouvement), une coupure entre ces deux prises de vue est symbolisée par l'apparition d'un masque entièrement blanc à l'instant de la coupure (cf. figure 3.2.a)). En pratique, il faut tenir compte du bruit (acquisition, capteurs, codage, signal, etc.), du mouvement des objets de la scène et de la caméra. De plus il est toujours possible d'avoir une succession de deux scènes similaires, pour lesquelles les pixels ne sont pas tous modifiés lors de la coupure.

3.1.2.2 Fondu enchaîné

Au contraire de la coupure qui se caractérise par un changement brutal du contenu de l'image entre l'instant t et l'instant $t + 1$, le fondu enchaîné est une transition progressive d'une prise de vue (I) à une autre prise de vue (J), qui possède une durée propre.

La transformation appliquée est la même en chaque pixel de l'image à un instant donné. A chaque instant du fondu, on effectue simplement une moyenne pondérée des valeurs d'un pixel donné dans la prise de vue (I) et dans la prise de vue (J), de façon à ce que l'influence de la prise de vue (I) soit décroissante au cours du temps, et au contraire l'influence de la prise de vue (J) soit croissante au cours du temps.

Ceci se formalise aisément de la façon suivante : soient t_0 l'instant de début et t_1 l'instant de fin du fondu. L'image résultante T_t entre les instants t_0 et t_1 s'écrit :

$$T_t = (1-\alpha)I_t + \alpha J_t \tag{3.4}$$

avec : $\alpha = f(\frac{t-t_0}{t_1-t_0})$ où $f(0) = 0$, $f(1) = 1$ et f continue. En pratique, on a le plus souvent : $f = Id$ (fonction identité).

On retrouve ainsi aux deux instants t_0 et t_1 les égalités, qui, en toute rigueur, sont impossibles à écrire (suivant notre modèle mathématique, une transition est définie sur l'intervalle ouvert $]t_0, t_1[$ uniquement) :

$$T_{t_0} = I_{t_0} \tag{3.5}$$
$$T_{t_1} = J_{t_1} \tag{3.6}$$

L'équation 3.4 exprime donc l'état de transition (cf. section 3.1.2) des pixels. Un fondu enchaîné se caractérise ainsi par l'absence de modèle géométrique, et le passage de tous les pixels par un même état de transition, pendant la durée du fondu.

La figure 3.3 propose une représentation visuelle des fondus sous la forme de masques de transition et un exemple de transition réelle.

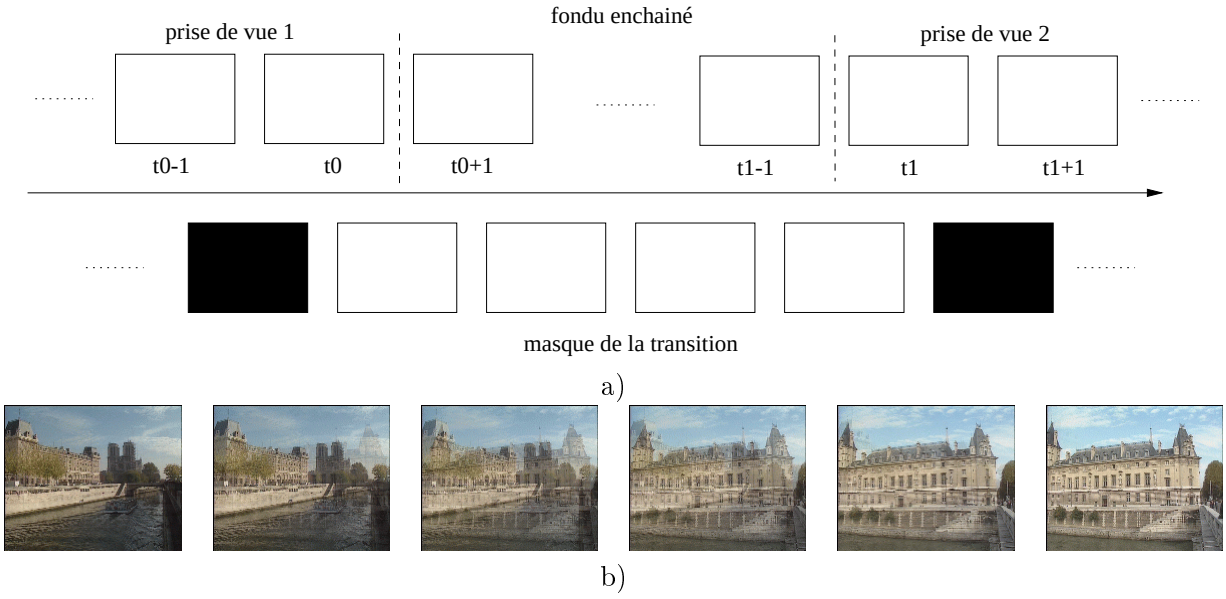


FIG. 3.3: Représentation d'un fondu entre les instants t_0 et t_1 : a) modèle du masque de transition. b) exemple d'un fondu réel.

A ce stade notons deux cas particuliers de fondus enchaînés. On appelle fondu (et non plus fondu enchaîné) le passage soit d'une prise de vue donnée à une image noire (ce que

nous noterons par la suite : $\forall t, J_t = 0$), soit d'une image noire ($\forall t, I_t = 0$) à une prise de vue donnée. Dans la pratique, les fondus enchaînés sont sensiblement plus courants que les fondus. Par la suite nous parlerons indistinctement de fondus ou de fondus enchaînés pour désigner l'ensemble de ces transitions.

3.1.2.3 Autres transitions

Après la description des transitions de types coupure et fondu, comme exemples des groupes 1 et 4, nous détaillons ici les deux classes de transitions restantes (groupes 2 et 3) et nous explicitons leurs modèles.

Les transitions du groupe 2, dites géométriques, (de durée non nulle, avec un modèle géométrique mais sans état de transition des pixels) s'apparentent de près aux coupures (groupe 1) qui n'en sont en fait qu'un cas particulier.

Localement on peut en effet considérer que chaque pixel subit une transition de type coupure. Sur toute l'image par contre, ces coupures locales n'ont pas lieu au même instant, mais suivant un modèle géométrique temporel. L'exemple le plus simple de cette catégorie de transitions est sans aucun doute le balayage horizontal dans sa forme la plus triviale, dont on donne une définition ci-dessous.

Définition 12. Balayage horizontal *Lors d'une transition de type balayage horizontal entre les prises de vue (I) et (J), une ligne verticale se déplace horizontalement dans l'image au cours du temps. En deçà de cette ligne, on conserve la prise de vue (I) ; au-delà la nouvelle prise de vue (J) apparaît.*

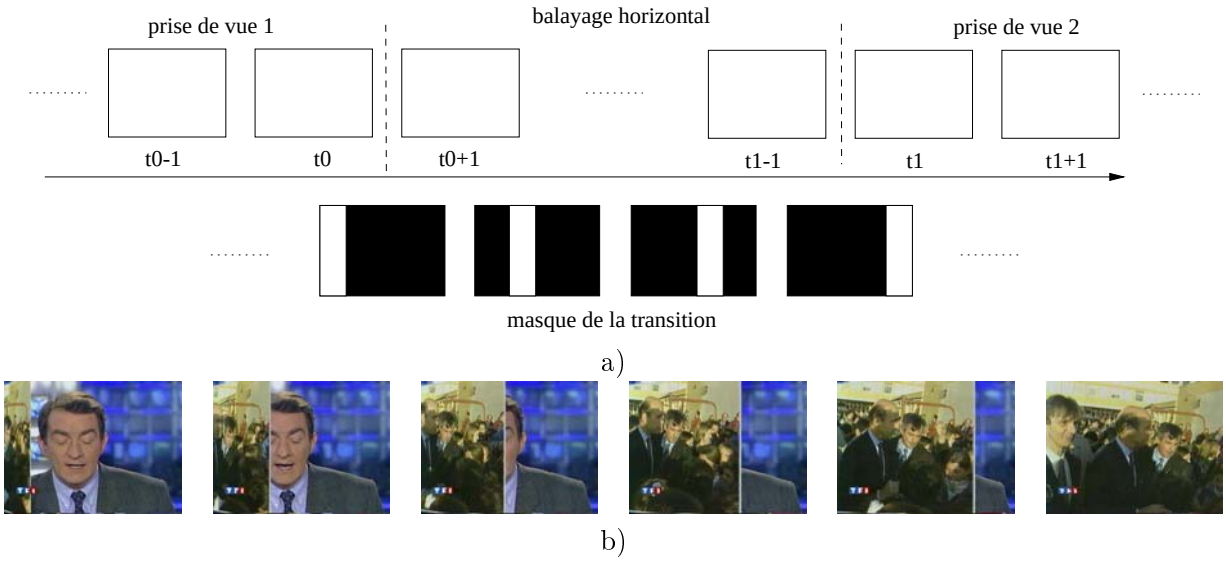


FIG. 3.4: Représentation d'un balayage horizontal entre les instants t_0 et t_1 : a) modèle du masque de transition. b) exemple d'un balayage réel.

A partir de cette définition, on accède directement à la visualisation sous forme de masque de transition de la figure 3.4.

Pour chaque transition géométrique, chaque pixel p de l'image ne connaît que deux états : avant transition, il appartient à la prise de vue (I), et après transition, il appartient à la prise

de vue (J) : l'état de transition, de durée nulle, n'apparaît donc pas :

$$\forall p(x,y), T_{t_i}(x,y) \quad = \quad \begin{cases} I(t_i,x,y) & \text{si} \quad \forall t \in]t_0,t_i], (x,y) \notin G_t \\ J(t_i,x,y) & \text{si} \quad \exists t \in]t_0,t_i], (x,y) \in G_t \end{cases} \tag{3.7}$$

Quant aux transitions du groupe 3, elles possèdent également un modèle géométrique de transition, mais dans ce dernier cas, les couleurs des pixels sont également affectées lors de l'état de transition pendant une durée non nulle. Chaque pixel de l'image passe donc par trois états : avant la transition, il appartient à la première prise de vue, pendant la transition, il connaît une évolution conforme au modèle de la transition (état de transition), après la transition, il appartient à la deuxième prise de vue. Ceci se résume par les équations suivantes :

$$\forall p(x,y), T_{t_i}(x,y) \quad = \quad \begin{cases} I(t_i,x,y) & \text{si} \quad \forall t \in]t_0,t_i], (x,y) \notin G_t \\ F(I,J,x,y) & \text{si} \quad (x,y) \in G_{t_i} \\ J(t_i,x,y) & \text{si} \quad \forall t \in [t_i,t_1], (x,y) \notin G_t \end{cases} \tag{3.8}$$

où F est la fonction caractéristique de l'état de transition (i.e. la fonction de modification de la couleur des pixels dans l'état de transition).

La "page tournée" est un exemple de transition de type 3 (une illustration du masque de transition dans ce cas est proposée dans la figure 3.5) :

Définition 13. Page tournée *On appelle page tournée l'ensemble des transitions entre deux prises de vue dont l'effet consiste à donner l'impression qu'on passe d'une prise de vue à la suivante comme si on tournait une page d'un livre.*

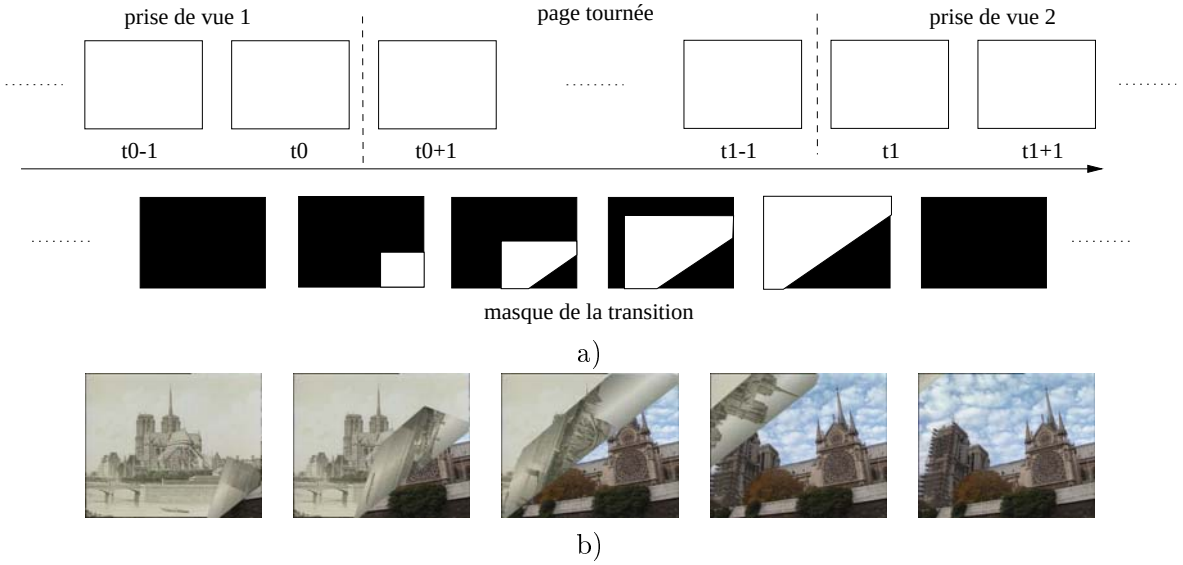


FIG. 3.5: Représentation d'une page tournée entre les instants t_0 et t_1 : a) modèle du masque de transition. b) exemple d'une page tournée réelle.

Il est important de remarquer dès à présent que pour une transition de type 1 ou 2, il n'y a pas de recouvrement des masques de transition au cours du temps (i.e. $\forall (t_i,t_j) \in]t_0,t_1]^2, G_{t_i} \cap G_{t_j} = \emptyset$). Par contre, pour une transition du groupe 3, des recouvrements ont

lieu, avec dans le cas extrême du fondu, un recouvrement total des masques pendant toute la durée de la transition ($\forall t \in]t_0, t_1], G_t = cste = Support(T_t)$).

Les modèles des quatre grands groupes principaux de transitions que nous avons distingués étant à présent définis, nous proposons, dans la section suivante, un état de l'art succinct des techniques existantes en détection de transitions.

3.1.3 Etat de l'art

En détection des transitions, deux grandes catégories de techniques existent en fonction de leur type de données d'entrée : comprimées ou non comprimées.

3.1.3.1 Images non comprimées

Cette première catégorie comprend de multiples méthodes toutes basées sur une comparaison entre deux ou trois [85] images successives.

Différents états de l'art existent à l'heure actuelle dans ce domaine [20, 3], aussi notre énumération des diverses techniques sera-t'elle succincte :

- **Différence de luminance / distance couleur calculée pixel à pixel** [105] Il s'agit ici d'un calcul global moyen sur toute l'image. Une coupure est détectée lorsque cette différence moyenne est supérieure à un seuil donné. Une version plus complexe basée sur des changements d'échelle dans les valeurs de luminance est donnée dans [4], permettant entre autres de détecter des transitions plus sophistiquées telles que les fondus.
- **Comparaison bloc par bloc** [74, 105] La dissemblance n'est plus calculée pixel à pixel sur toute l'image mais sur des sous-parties de deux images consécutives, sous forme d'une différence de luminance classique, ou bien sous forme d'un critère de probabilité, fonction de la luminance moyenne et de la déviation standard de l'histogramme de luminance. Lorsqu'une certaine proportion de blocs possède une valeur de dissemblance supérieure à un certain seuil, une coupure est détectée.
- **Comparaison d'histogrammes** [74, 105, 89] de façon à limiter la sensibilité du critère aux mouvements dans la prise de vue. On distingue dans cette catégorie les techniques basant leur détection sur une simple différence d'histogrammes couleur ou luminance, de celles utilisant une mesure plus complexe, toujours sur les histogrammes, telle qu'une mesure de χ^2, ou le test de Kolmogorov-Smirnov [85], ou enfin une formule classique d'intersection d'histogrammes. Ces dernières mesures ont la particularité de fournir des valeurs de dissemblance plus aisées à extraire aux instants de coupures. Zhang *et al.* [105] proposent en outre une variante plus robuste de mesure sur les histogrammes grâce à leur technique intitulée *twin-method* : un seuillage dual est appliqué sur l'histogramme permettant ainsi une meilleure efficacité lors du mouvement d'objets dans la scène. Cette technique comprend en outre une étape de suppression des mouvements simples de la caméra par flot optique. Fisher *et al.* [45] procède également par différence d'histogrammes couleur, précédée d'une étape de calcul et de suppression de ce qu'il nomme l'**énergie de mouvement**, simple différence d'images successives, filtrée par une gaussienne.
- **Estimation de mouvement** [11, 19, 49] La segmentation en plans repose ici soit sur l'analyse du mouvement des objets de la scène, soit sur celle du mouvement dominant de la scène elle-même par estimation robuste. Cette classe de techniques, qui assimile les ruptures de modèles de mouvement aux changements de scènes, possède bien sûr l'avantage non négligeable d'être robuste aux mouvements d'objets.

- **Modèles théoriques explicites des transitions** [53, 57, 54, 55] Hampapur *et al.* modélisent les prises de vue et les différents types de transitions comme étant des éléments de leur *video edit model*. A l'aide de leurs caractéristiques mathématiques théoriques, les transitions de type coupures, ou fondus sont donc extraites.
- **Extraction de caractéristiques** [104, 103, 26] En fonction du nombre de bords entrants et sortants dans la scène au cours du temps, les transitions sont extraites du fichier vidéo. D'après les auteurs, cette technique offre une robustesse non négligeable aux mouvements d'objets, mais reste sensible aux changements d'illumination de la scène. Cette méthode permet en outre une extraction des transitions de type fondus et balayages, grâce à une étude de la localisation dans l'image des bords entrants et sortants.
- **Méthode des plateaux** Enfin Yeo [101] propose une comparaison non plus entre deux images successives d'un même fichier vidéo, mais entre deux images à t et $t + k$ de façon à détecter les transitions progressives. Ces transitions passent en effet quasiment inaperçues, les valeurs de dissemblance étant très faibles d'une image à une autre. Au contraire entre deux images non plus successives mais à une certaine durée l'une de l'autre, la valeur de dissemblance augmente. Les transitions progressives se traduisent ainsi sous la forme de plateau dans la courbe d'évolution temporelle du critère, qu'il s'agit alors de détecter.

Terminons la catégorie des techniques travaillant sur des séquences d'images non comprimées par une remarque concernant les approches à passes multiples [105]. Le temps de traitement est alors réduit par un premier examen du fichier vidéo à un pas d'échantillonnage élevé de façon à détecter des zones éventuelles de transitions, zones qui sont par la suite traitées plus précisément pour sélectionner les véritables transitions.

3.1.3.2 Images comprimées

Dans le but justement de réduire ces temps d'exécution, par exemple en évitant un décodage des séquences, une deuxième classe de techniques de détection de coupures s'est développée, traitant directement le flot codé. Il est en effet possible d'exploiter le résultat du codage MPEG ou M-JPEG pour l'extraction des transitions.

Zhang *et al.* [106] et Arman *et al.* [8] utilisent ainsi le fait que le retour à un codage DCT direct d'une image donnée du flot MPEG, sans compensation de mouvement par prédiction, signifie un changement tel dans la continuité des images, qu'il s'agit certainement d'une transition. D'autres travaux tels que ceux Feng [44] et Yeo [100] sont également très proches de ces techniques, Yeo ayant bâti un algorithme permettant d'extraire les transitions graduelles.

Deardoff *et al.* [33] se basent quant à eux sur la taille des différentes images une fois codées, une augmentation soudaine de cette taille signifiant également la présence d'une transition. Cette technique est utilisée à la fois pour la détection de coupures et de fondus, mais dépend d'un seuil fixé manuellement. Enfin Ardizzone *et al.* [6] construisent un "modèle statique implicite" à partir d'un réseau de neurones 3 couches, prenant en entrée des images de luminance sous-échantillonées directement issues du flot MPEG. Cette méthode ne fonctionne cependant que pour les transitions de type coupure et n'utilise pas du tout l'information couleur contenue dans les images.

En conclusion, citons deux articles fournissant une étude comparée des performances de ces diverses techniques [18, 27]. Notons que les meilleurs taux de détection atteignent des valeurs de $95 - 98\%$ pour des taux de fausses alarmes souvent très élevés ($150 - 500\%$),

nécessitant une phase de vérification ultérieure. Enfin, malgré le grand nombre de techniques disponibles actuellement, la détection de transitions, et notamment des transitions graduelles, est loin d'être parfaite, puisqu'elle reste souvent encore sensible aux mouvements de caméra et d'objets présents dans la scène, ou aux changements d'illuminations, tels que les flashes par exemples.

Cet état de l'art étant terminé, nous proposons à présent d'analyser les différents choix que nous nous sommes imposés pour bâtir nos propres algorithmes de détection de transitions et les conséquences que ces choix impliquent.

3.1.4 Choix adoptés - Méthodes proposées

Au regard du choix déjà exprimé dans le chapitre introductif de ce mémoire de ne bâtir que des outils d'un niveau sémantique peu élevé, mais qui combinés entre eux et associés à des règles de décision, permettent d'atteindre un plus haut niveau, les algorithmes de détection de transition proposés ont la qualité d'être extrêmement simples. Cette caractéristique essentielle trouve un argument de poids en sa faveur dans le fait que le découpage temporel, et donc la détection de transitions, ne sont et restent qu'une toute première étape de l'indexation d'un document vidéo. Dans l'objectif final d'obtenir un processus complet d'indexation en temps réel, cette première étape se doit d'être rapide, ce qui est réalisé si les algorithmes sont simples à mettre en œuvre (tout en restant bien sûr efficaces). Cet objectif "temps réel" fait partie de nos buts primordiaux à atteindre pour les outils d'indexation proposés.

Expérimentalement deux transitions, parmi toutes celles existantes, sont essentielles quel que soit le type de document vidéo : la coupure et le fondu. Les autres transitions présentent en effet une fréquence d'apparition moins élevée, pour l'ensemble des documents vidéo. Nous avons donc fait le choix de développer deux algorithmes différents pour chacune de ces deux transitions ; nous rejoignons ainsi la décision de construire des outils qui soient applicables indifféremment à une grande diversité de documents vidéo. Nous étudierons en outre, dans la section 3.2.4.3, comment l'algorithme de détection des coupures fournit également des informations supplémentaires sur la nature des transitions géométriques simples telles que les balayages ou les pages tournées.

Pour chacun des deux algorithmes de détection, des critères de dissemblance les plus simples possibles ont donc été sélectionnés ; nous verrons alors que la combinaison d'une détection locale et d'un filtrage morphologique permet de contrebalancer efficacement ce choix et d'atteindre un taux de détection maximal, supérieur ou égal à celui obtenu par des techniques plus complexes et d'une rapidité moindre.

En faisant ce choix de suivre la dissemblance entre deux images successives du document vidéo par l'étude d'un critère, nous éliminons du même coup toute technique plus onéreuse basée par exemple sur l'analyse et l'étude du mouvement de la scène [19] ou l'extraction de contours [104, 103]. Ces méthodes requièrent en effet la capacité de modéliser le mouvement ou du moins d'en calculer une expression, par flot optique par exemple. Or la modélisation, comme l'étude du mouvement, sont deux problèmes complexes et souvent non encore résolus, sauf dans des cas bien précis, comme le prouve l'énorme production de littérature dans ce domaine.

De même, nous avons fait le choix de travailler sur des données non comprimées en entrée, malgré l'engouement récent pour des techniques de détection directement sur le flot MPEG [8, 44, 33, 106, 69]. Ceci est toujours motivé par la volonté d'être le plus général possible, tous les documents vidéo existants n'étant pas tous à l'heure actuelle sous forme comprimée. De

plus, au vu de l'état de l'art réalisé dans la section précédente, il semble qu'il soit moins aisé d'extraire du flot comprimé des informations de transitions graduelles.

Les données couleur contenant plus d'information que la seule luminance, il serait dommage de se priver de cette richesse supplémentaire, aussi le critère de dissemblance est-il appliqué à des images couleur. Par contre, afin d'atteindre notre objectif de rapidité, c'est-à-dire au moins le temps réel, la fréquence de comparaison des images est fixée à 5Hz. Nous reviendrons sur cette valeur dans le paragraphe 3.2.7, mais notons dès à présent qu'il s'agit du meilleur compromis entre les deux objectifs : être le plus rapide possible (comparer le plus petit nombre d'images possible) et ne pas perdre de transitions. Enfin les images sont soumises à un échantillonnage spatial par 2 dans les deux directions horizontale et verticale.

3.2 Détection des coupures

Les coupures apparaissent comme de brusques ruptures de modèle dans la succession des images. Intuitivement, si on calcule une dissemblance quelconque sur une paire d'images successives, cette dissemblance aura des valeurs faibles à l'intérieur d'une même prise de vue et au contraire, prendra une valeur élevée à l'instant de la coupure, entre les deux images de part et d'autre, i.e. appartenant à deux prises de vue différentes.

Une fois le critère de dissemblance et sa technique de calcul choisis, détecter les coupures revient donc à sélectionner, ou seuiller, les hautes valeurs de critère.

Nous proposons donc un algorithme en deux temps, pour lequel on effectue d'abord une détection locale des changements entre deux images successives, puis une extraction temporelle par filtrage morphologique [83, 82] des pics parmi les valeurs du critère. Mais détaillons tout d'abord notre choix de critère de dissemblance.

3.2.1 Critère de dissemblance

3.2.1.1 Présentation des divers critères

Comme cela a déjà été explicité dans l'état de l'art, section 3.1.3, rappelons qu'un grand nombre de méthodes de détection de coupures sont basées sur le calcul d'un critère de dissemblance entre deux images successives. L'objectif de ce paragraphe est donc de donner les définitions mathématiques des critères de dissemblance utilisés le plus couramment, afin d'en faire une comparaison dans la section suivante, cette comparaison débouchant sur notre propre choix de critère pour la détection de coupures.

Parmi les plus critères de dissemblance les plus courants [74, 20], citons donc :

– la différence absolue d'histogrammes entre deux images successives :

$$C(\text{diff\_hist}) \quad = \quad \frac{\sum_{i=1}^{N} |H_t[i] - H_{t+1}[i]|}{N} \tag{3.9}$$

avec N le nombre de classes de l'histogramme H_t à l'instant t.

– l'intersection d'histogrammes :

$$C(\text{inter\_hist}) \quad = \quad \sum_i \frac{\min(H_{t+1}[i], H_t[i])}{\max(H_{t+1}[i], H_t[i])} \tag{3.10}$$

– le calcul de mesures statistiques toujours sur les histogrammes, comme par exemple le χ^2, pour lequel deux formulations sont possibles :

$$C(\chi_2) = \frac{1}{N}\sum_{i=1}^{N}\frac{(H_t[i]-H_{t+1}[i])^2}{H_{t+1}[i]} \tag{3.11}$$

$$C(\chi_2) = \frac{1}{N}\sum_{i=1}^{N}\frac{(H_t[i]-H_{t+1}[i])^2}{H_{t+1}[i]+H_t[i]} \tag{3.12}$$

avec N le nombre de classes de l'histogramme H_t à l'instant t.
La somme n'est évidemment effectuée que sur les niveaux de gris représentés dans au moins l'une des images, pour la deuxième formulation, et sur les niveaux de gris de l'image $t+1$ pour la première formulation.

– la moyenne des différences de luminance pixel à pixel :

$$C(\text{diff\_lumi}) = \frac{\sum_{x,y}|I_t(x,y)-I_{t+1}(x,y)|}{\sum_{x,y}1} \tag{3.13}$$

avec I_t l'image à l'instant t. $I_t(x,y)$ est dans ce cas une valeur de luminance, i.e. un niveau de gris dans l'intervalle $[0,255]$.

– la moyenne des distances, toujours pixel à pixel, dans un espace de couleur, RGB par exemple :

$$C(\text{dist\_RGB}) = \frac{\sum_{x,y}\text{dist}_{\text{RGB}}(I_t(x,y),I_{t+1}(x,y))}{\sum_{x,y}1} \tag{3.14}$$

avec I_t l'image à l'instant t. $I_t(x,y)$ est ici un vecteur couleur à trois composantes RGB, toutes trois prises dans l'intervalle $[0,255]$.

Notons que la différence d'histogrammes ou des mesures telles que le χ^2 s'appliquent indifféremment à des images à niveaux de gris ou couleur, alors que la différence moyenne de luminance et la distance moyenne dans RGB sont deux notions correspondantes suivant qu'on les applique à des images à niveaux de gris ou à des images couleur. Ces premiers critères statistiques ne font en outre pas du tout intervenir l'information spatiale de changement, au contraire de la différence de luminance ou de la distance couleur, pour lesquelles un calcul au niveau du pixel intervient.

Par ailleurs, si la formulation de ces critères apparaît dans cette section comme une mesure moyenne globale sur l'ensemble de l'image, il est tout à fait envisageable de n'effectuer ces calculs que bloc par bloc dans les images, au prix d'une légèrement modification de la sommation effectuée dans les équations. Il s'agit là d'une mesure plus locale sur laquelle nous reviendrons dans la section 3.2.2.

Dans le cadre de ce mémoire seuls les trois critères 3.9, 3.12, 3.14 appliqués à des images couleur ont été étudiés, puisqu'on recherchait un critère simple et peu coûteux en temps de calcul. Nous en présentons par la suite une comparaison.

3.2.1.2 Comparaison et choix

La comparaison des trois critères retenus précédemment a été plus précisément basée sur l'étude des courbes d'évolution temporelle de chacun d'eux (i.e. l'évolution de leur valeur entre deux images successives au cours du temps). Ces courbes doivent en théorie présenter des pics aux instants de transition (valeur élevée du critère de dissemblance), émergeant d'un ensemble de valeurs de critère plus faibles, correspondant au niveau de bruit. Typiquement l'extraction de pics, i.e. de transitions, se fait par l'application d'un seuil sur ces courbes temporelles.

De l'étude expérimentale de ces trois critères sont ressorties les caractéristiques suivantes :

- le rapport signal sur bruit augmente avec l'utilisation de la différence d'histogrammes, puis le calcul du χ^2, par rapport à la distance couleur ;
- le seuil à appliquer pour l'extraction des pics est donc plus aisé à placer dans le cas d'une différence d'histogrammes et du χ^2 ;
- le calcul de la distance couleur est légèrement plus simple à mettre en œuvre et plus rapide que les deux autres critères ;
- la différence d'histogrammes et surtout le χ^2 étant des mesures statistiques, elles nécessitent d'être appliquées globalement sur toute l'image pour rester significatives.

De toutes ces remarques découle notre choix de conserver la distance couleur moyenne dans l'espace RGB comme critère de dissemblance entre deux images successives. Il ressort en effet des remarques précédentes que cette distance reste la plus simple à calculer, même si l'évaluation des deux autres critères est elle-aussi relativement aisée. En ce sens nous confortons notre volonté de ne bâtir que des algorithmes de mise en œuvre simple et rapide. L'obligation d'appliquer les mesures statistiques sur les images entières a également guidé notre choix vers la distance couleur dans la mesure où nous estimons nécessaire de conserver le plus longtemps possible une information locale de dissemblance et non globale à toute l'image. Mais ceci sera amplement détaillé dans la section suivante. Seule la première remarque concernant la croissance du rapport signal sur bruit pour les deux mesures d'histogrammes (remarque dont découle le deuxième point de placement d'un seuil plus aisé) invalide notre choix. Pour cette raison nous proposons dans la section 3.2.5 une étape de filtrage temporel des courbes d'évolution du critère de dissemblance qui compensera avantageusement ce rapport signal sur bruit plus faible, comme nous le verrons.

Au final, nous avons donc conservé notre choix de distance couleur comme critère de dissemblance entre deux images successives. La section suivante propose alors un calcul par blocs de ce critère de manière à conserver l'information locale de changement dans les images.

3.2.2 Détection locale

Au cours de l'état de l'art établi dans la section 3.1.3, deux groupes de méthodes ont été dégagés, suivant leur application du critère de dissemblance globalement ou localement à toute l'image. Ce dernier groupe, dont les travaux les plus anciens sont certainement ceux de Nagasaka et Tanaka [74], présente l'avantage de conserver l'information spatiale de la transition, c'est-à-dire le lieu précis des changements intervenant dans l'image. En outre, le calcul d'un critère globalement sur toute l'image courante fait intervenir au final une moyenne de l'ensemble de mesures ponctuelles sur l'image. L'information de dissemblance est donc affaiblie, puisque moyennée.

Dans l'exposé des différentes catégories de transitions (cf. section 3.1.2), il est de plus apparu que certaines transitions (groupe 2) pouvaient localement apparaître comme des coupures, l'enchaînement de ces coupures locales pendant la durée de la transition s'effectuant suivant ce que nous avons appelé le modèle géométrique de la transition.

Afin, d'une part de ne pas moyenner directement les valeurs de critère sur la totalité de l'image, et d'autre part de conserver l'information spatiale contenue dans la transition, information essentielle dans le cas des transitions possédant un modèle géométrique, nous avons choisi d'effectuer une détection locale des lieux de dissemblance dans l'image. En ce sens, cette partie de l'algorithme est proche de celui développé dans [74].

On effectue sur chaque image du document vidéo une partition en damier (typiquement des rectangles de taille 20×20 pixels, pour une image sous-échantillonée spatialement). Le critère de dissemblance est alors évalué sur chacune des régions obtenues après découpage. On compare ainsi deux rectangles correspondants dans deux images successives et on obtient pour chaque paire de rectangles, une distance moyenne pixel à pixel dans l'espace des couleurs (cf. le choix de critère de dissemblance détaillé dans la section 3.2.1).

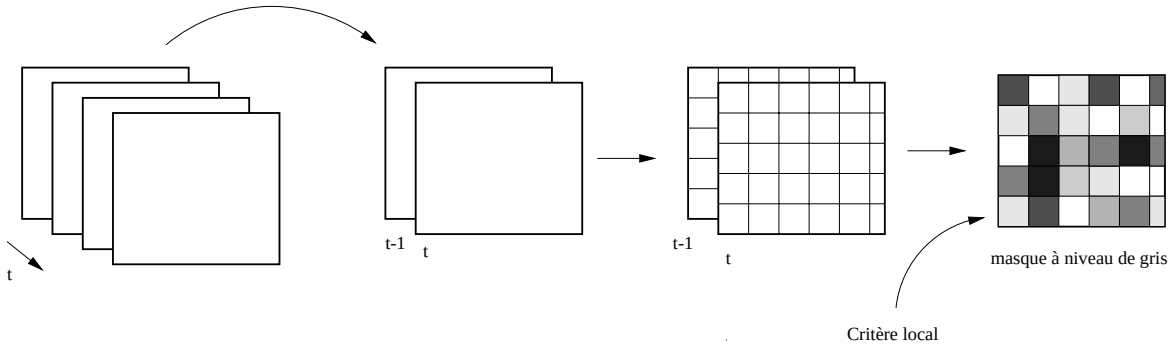


FIG. 3.6: Algorithme de détection de coupures, première étape : calcul local d'un critère de dissemblance.

La figure 3.6 illustre les différentes étapes de cette première partie de l'algorithme, que constitue la détection locale de coupures entre deux images successives du document vidéo. Le résultat de cette étape est obtenu sous la forme d'un masque de transition (notion introduite dans la section 3.1.2.1) à niveaux de gris, un niveau de gris par rectangle. Ces niveaux de gris correspondent à la valeur moyenne du critère de dissemblance sur chacun des rectangles. Plus le niveau de gris est sombre, plus la valeur du critère est faible, i.e. plus les deux rectangles comparés sont similaires. Un exemple d'un masque réel de transition entre deux images successives d'un document donné est fourni dans la figure 3.7.

image 46 image 47 masque de transition

FIG. 3.7: Exemple de masque de transition obtenu pour une coupure.

Ces masques et leur utilisation seront étudiés plus avant par la suite (section 3.2.4), mais pour l'instant nous proposons d'effectuer une comparaison de cette détection locale avec deux autres types de détection.

3.2.3 Comparaison avec des détections globale ou mosaïque

Ainsi que cela a déjà été évoqué dans le paragraphe précédent, le calcul d'un critère de dissemblance globalement entre deux images successives, suivant par exemple l'expression de la distance moyenne dans un espace couleur (cf. section 3.2.1.1), présente l'inconvénient conséquent de fournir des mesures de dissemblance moyennées sur l'ensemble de l'image et de perdre la dépendance spatiale de la mesure avec les changements locaux.

Or être capable de localiser les zones de changement dans les images renseigne sur la nature propre de ces changements : il est ainsi envisageable de caractériser les différentes transitions par l'étude de leur géométrie, certaines transitions plus progressives telles que les balayages n'étant pas perceptibles, une fois moyennées sur toute l'image (cf. section 3.2.4.3), ou même d'extraire les zones de changements que sont les incrustations d'images et/ou de texte (cf. sections 6.2 et 6.3 et figure 3.8), ou les objets en mouvement (figure 3.9). Là encore une étude simple de la géométrie des masques de transitions fournit un outil puissant de classification des diverses causes de changements locaux. Il est alors plus aisé de lancer par la suite, et sur ces zones locales uniquement, des outils plus sophistiqués tels que la détection et la reconnaissance de texte ou un estimateur de mouvement.

image 161 image 162 masque de transition

FIG. 3.8: Exemple de masque de transition binaire obtenu dans le cas de l'apparition d'un bandeau de texte (séquence *seq9*).

Si conserver l'information spatiale du changement dans les images apparaît comme primordial, il reste la question du découpage de l'image à effectuer : que choisir entre un partitionnement régulier, géométrique, totalement arbitraire et un partitionnement respectant le contenu des images, i.e. les contours des objets présents et les mouvements éventuels ?

Afin d'évaluer la faisabilité de cette dernière solution, nous avons choisi pour partitionnement la segmentation, aussi appelée mosaïque, obtenue par des outils morphologiques tels que ceux développés dans les travaux de Cichosz et Meyer [25, 72]. Le critère de dissemblance est alors calculé entre deux régions correspondantes dans deux images successives, segmentées indépendamment l'une de l'autre.

Si cette méthode permet de tenir compte au mieux des objets présents dans la scène et de ne pas mélanger les informations spatiales, après réalisation, elle comporte cependant plusieurs inconvénients qu'il est bon de rappeler :

- Bien sûr elle sous-entend que toutes les images de la séquence ont été segmentées. Outre le coût en temps de calcul, le problème de l'exactitude de la segmentation reste posé.

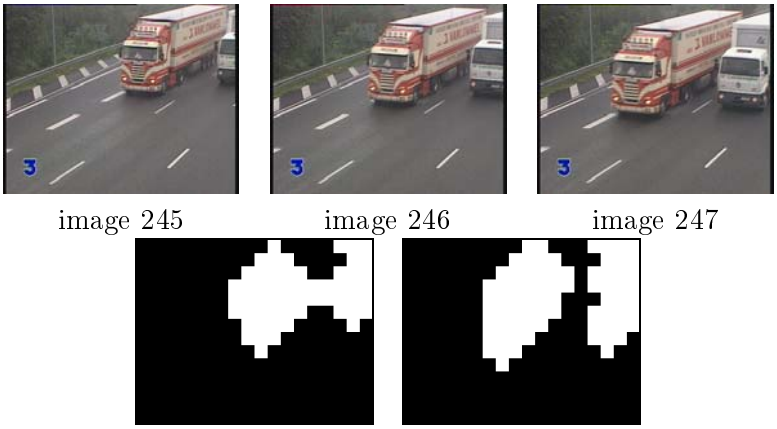


FIG. 3.9: Exemples de masques de transition binaires obtenus dans le cas du mouvement d'un objet.

- Une fois toutes les segmentations réalisées, il reste à mettre en correspondance les paires de régions dans deux images successives et dans le cas d'un mouvement, une compensation de ce mouvement est nécessaire, ce qui soulève à nouveau la question du temps de calcul.
- L'utilisation d'une segmentation et de l'estimation du mouvement pour la détection des transitions est contraire à l'ordre logique et à la complexité du problème de l'indexation qui ne peut se faire de la même façon sur l'ensemble des images. Ainsi les traitements appliqués à l'ensemble des données, c'est-à-dire des images, tels que la détection des transitions, se doivent d'être simples et rapides, ce qui permet de réserver plus de temps pour des traitements plus coûteux et plus sophistiqués, mais appliqués à certaines images ou portions d'images bien ciblées.
- Les trois arguments précédents vont dans le sens du principe de rapidité et de simplicité que nous nous sommes imposés pour la détection des transitions (cf. section 3.1.4).
- Enfin, toujours dans l'objectif de détecter des transitions ou des changements locaux tels que les incrustations ou bandeaux, il faut garder à l'esprit que les géométries impliquées sont alors de type parallélépipédiques et ne nécessitent donc pas véritablement de segmentations poussées. Ce dernier point est à moduler dans le cas où la raison d'apparition de changement local est due à un objet de forme quelconque. Dans ce cas toutefois, un partitionnement en damier donne une première approximation de la zone qu'il conviendra de segmenter plus finement par la suite.

Pour toutes ces raisons, le partitionnement en damier, solution intermédiaire entre un calcul global sur l'image et l'utilisation d'une mosaïque, a été retenu.

Rappelons que le processus de détection des transitions par mosaïque peut cependant être simplifié en remplaçant la compensation de mouvement nécessaire à la mise en corrélation de deux segmentations successives, par une simple projection de la segmentation au temps $(t-1)$ sur l'image t : on calcule le critère de dissemblance entre les images $(t-1)$ et t dans les régions de la segmentation au temps $(t-1)$. Ceci est évidemment soumis à conditions : les objets de la scène ne doivent pas bouger trop vite, ils doivent être rigides et visibles en

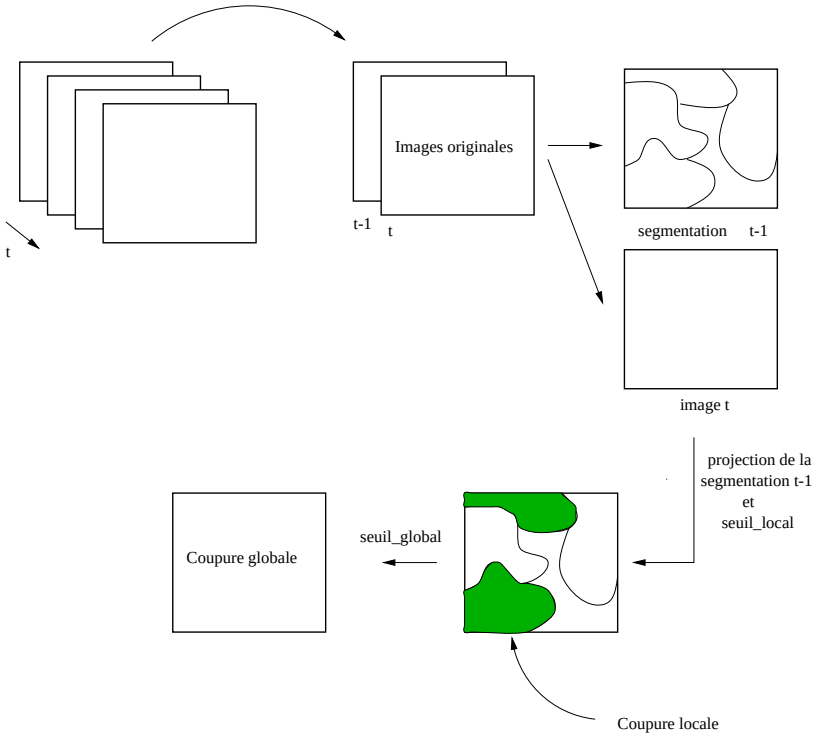


FIG. 3.10: Algorithme général de détection locale de transitions à partir d'une segmentation.

totalité. Dès qu'une de ces conditions n'est pas respectée, on réintroduit de l'arbitraire dans le découpage. La figure 3.10 fournit une représentation des grandes étapes de cet algorithme et on donne, dans la figure 3.11, deux exemples de masques de transitions obtenus dans le cas d'une coupure : les régions blanches ou grises dans l'image masque correspondent aux zones de changements locaux, les régions noires sont considérées comme sans changement.

3.2.4 Obtention de masques de transition

Le calcul local du critère de dissemblance débouche sur la construction d'un masque de transition à niveaux de gris, chaque niveau étant proportionnel à la valeur du critère. Plus le niveau de gris est sombre, plus la valeur du critère est faible, i.e. plus les deux rectangles comparés sont similaires.

Ainsi que cela a déjà été formulé, ces masques contiennent l'information spatiale, ou géométrique de la transition. Dans le cas d'une coupure, on s'attend ainsi à ce que le masque ait de fortes valeurs de niveaux de gris sur toute sa surface à l'instant de la coupure (figure 3.7). La classe de transitions du groupe 2, possédant un modèle géométrique, mais pas d'état de transition des pixels, s'apparente en fait à des transitions de type coupures locales dans les images. Leur modèle géométrique est la principale caractéristique qui va alors permettre de distinguer par exemple un balayage gauche-droite, d'un balayage haut-bas.

Pour extraire l'information spatiale de ces masques de transition, nous proposons un filtrage spatial suivi d'une étude de l'évolution de leur géométrie au cours du temps. Le suivi de cette géométrie permet une classification et une reconnaissance du type des transitions rencontrées dans un document vidéo (section 3.2.4.2). Mais avant de chercher à reconnaître à quel type

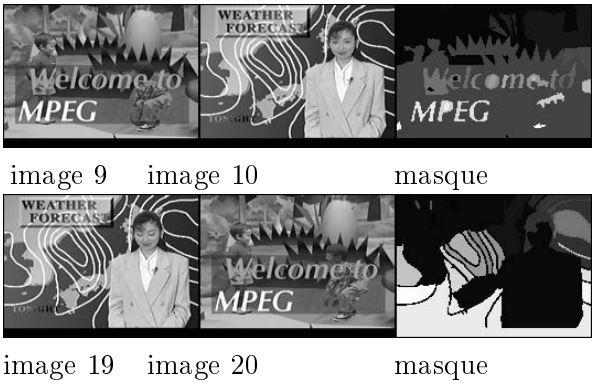


FIG. 3.11: Exemples de coupures détectées par différence locale de luminance. La partition est une segmentation. On présente successivement l'image au temps $(t-1)$ et l'image au temps t et le masque (toutes les zones grises ou blanches sont considérées comme des zones de transition).

de transition on a affaire, il est primordial, toujours à partir du masque de transition, de résumer l'information de dissemblance qui est toujours locale dans le masque, pour déterminer si globalement il y a présence ou non d'une transition sur l'image. Pour cela, deux méthodes différentes sont envisageables, suivant qu'on conserve un masque à niveaux de gris, ou qu'on le transforme en un masque binaire.

3.2.4.1 Masques binaires ou à niveaux de gris

Nous étudions ici les apports respectifs de l'utilisation d'un masque binaire, ou bien d'un masque à niveaux de gris. Le calcul local de dissemblance entre images successives fournit un masque de transition à niveaux de gris, un niveau de gris par rectangle. La première utilisation de ces masques est de répondre à la question "Y a t'il eu transition ?". Il faut donc résumer toutes les valeurs numériques locales de dissemblance en une réponse binaire de type "transition ou non".

Pour ce faire, deux procédés différents sont envisageables :

- soit on décide déjà en chaque rectangle si on a eu une coupure locale ou non (le passage à une information binaire a lieu ici). Il faut alors condenser par la suite n informations binaires de coupures locales en une information binaire de coupure globale.
- soit on conserve le plus longtemps possible l'information numérique et on repousse au maximum le passage à un masque binaire.

Cette distinction "niveaux de gris / binaire" retrouve également toute son importance pour le choix du type de masque en entrée du filtrage spatial proposé dans la section suivante (cf. section 3.2.4.2). Toutefois dans ce contexte, et avec l'utilisation des filtres morphologiques, nous verrons qu'il est équivalent de filtrer des masques binaires ou de filtrer des masques à niveaux de gris, puis de les seuiller ensuite.

Revenons cependant sur les avantages et inconvénients des deux types de masques, pour la détection de transitions. L'inconvénient principal de l'utilisation de masques binaires réside dans la nécessité d'ajuster un premier seuil, noté *seuil local*, de binarisation du masque. On rajoute donc un paramètre supplémentaire à l'algorithme.

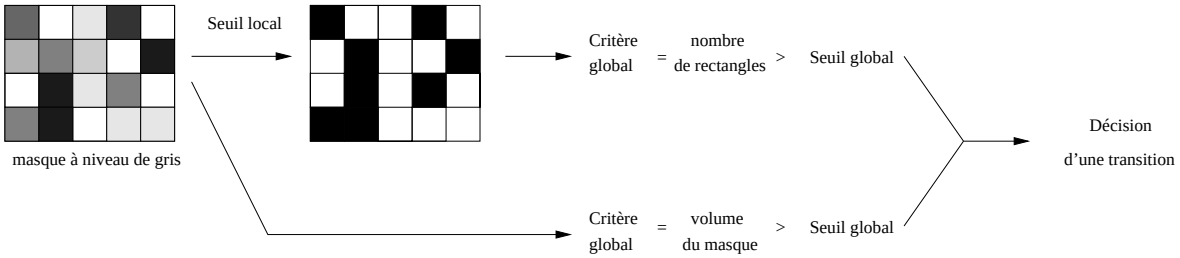


FIG. 3.12: Représentation des deux modes possibles de décision de la présence d'une transition à partir du masque à niveaux de gris.

Les deux options possibles pour, à partir de l'image à niveaux de gris, accéder à l'information globale de présence ou non d'une transition, sont résumées dans la figure 3.12. Dans le cas d'une binarisation des masques (application d'un seuil local), la décision d'une coupure globale se fait naturellement par un seuillage du pourcentage de rectangles blancs (i.e. ayant subi une transition locale). Si on conserve un masque à niveaux de gris, on seuille cette fois-ci le volume du masque, i.e. on effectue la somme des niveaux de gris des différents pixels.

L'option consistant à garder le masque à niveaux de gris le plus longtemps possible permet de s'affranchir du seuil local. Cependant dans le cas idéal et théorique où les rectangles ont tous la même taille et où il a été possible de découper l'image en un nombre entier de rectangles, l'algorithme revient à calculer un critère global, comme cela se démontre aisément de la façon suivante :

Soit I_t l'image à l'instant t, N_r le nombre de rectangles contenus dans I_t, on note $card(I)$ le nombre de pixels dans une image I, $card(r)$ le nombre de pixels dans un rectangle r. $card(I)$ et $card(r)$ sont constants au cours du temps et pour tous les rectangles. On a alors :

$$\begin{aligned}
C_{\text{local niv. de gris}} &= \frac{1}{255 \times N_r} \sum_r \frac{\sum_{(x,y)\in r} |I_t(x,y) - I_{t+1}(x,y)|}{card(r)} \\
C_{\text{local niv. de gris}} &= \frac{1}{255 \times N_r \times card(r)} \sum_r \sum_{(x,y)\in r} |I_t(x,y) - I_{t+1}(x,y)| \\
C_{\text{local niv. de gris}} &= \frac{1}{255 \times card(I)} \sum_{(x,y)\in I} |I_t(x,y) - I_{t+1}(x,y)| \\
C_{\text{local niv. de gris}} &= C_{\text{global}}
\end{aligned} \tag{3.15}$$

En pratique, notre algorithme de découpage de l'image fournit des rectangles de bord plus grands que la taille imposée par l'utilisateur (cf. figure 3.13).

Sur chaque rectangle on procède en outre à un seuillage de la valeur locale du critère, si celle-ci dépasse 255. On calcule en effet des distances entre couleurs représentées par des vecteurs à trois composantes prises dans l'intervalle $[0, 255]$. La distance maximale théorique possible, dans le cas d'une norme L^2, est obtenue pour les deux vecteurs $(0, 0, 0)$ et $(255, 255, 255)$, soit $255.\sqrt{3}$. Plutôt que de normaliser l'échelle des valeurs de critère pour la ramener de $[0, 255.\sqrt{3}]$ à l'intervalle $[0, 255]$, de façon à représenter aisément les valeurs par des niveaux de gris sur le masque de transition, nous avons préférer seuiller toutes les valeurs plus grandes que 255 pour les ramener à cette valeur maximale, avec pour effet d'obtenir des masques plus uniformément blancs lors de transitions globales de type coupures.

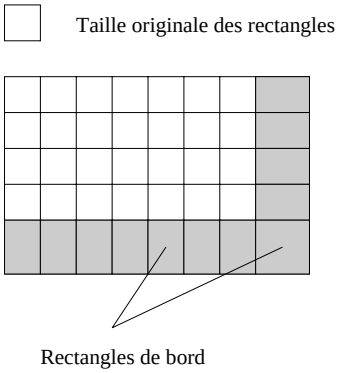


FIG. 3.13: Diverses tailles de rectangles dans le partitionnement.

Ainsi, en pratique, du fait des tailles différentes de certains rectangles, du seuillage à 255 des valeurs élevées de critère, et si en outre - et c'est surtout ce dernier point proposé dans la section 3.2.4.2 qui fait la différence - une étape de filtrage spatial est ajoutée, les deux critères local niveaux de grix et global diffèrent.

La deuxième option d'utilisation d'un masque binaire (figure 3.12) est, elle, de façon évidente différente du critère global, qu'un filtrage spatial ait lieu ou non.

Notons dès à présent que, si l'utilisation de masques binaires nécessite un paramètre supplémentaire, ce seuil local est cependant relativement peu dépendant des données et qu'il se fixe en outre aisément à une valeur stable quant aux résultats, au vu des tests effectués sur les documents vidéo dont nous disposons (cf. section 3.2.7). La même valeur (70) a ainsi été conservée pour tous les documents vidéo, tout en aboutissant à des résultats d'un niveau tout à fait satisfaisant.

En conclusion, poussés par la volonté de nous affranchir du plus grand nombre de paramètres possibles, nous avons finalement retenu l'option : découpage local et conservation des masques à niveaux de gris le plus longtemps possible (de façon à conserver l'information spatiale et accèder à tout moment à la géométrie des transitions et aux zones locales de changement) et seuillage du volume du masque à niveau de gris.

Cependant gardons à l'esprit que, de par la suppression du seuil local, les deux critères local à niveaux de gris et global se rapprochent, avec l'inconvénient déjà cité dans la section 3.2.3 de l'obtention de valeurs de dissemblance moyennées, et donc plus faibles. Ce retour vers un critère global sera contrebalancé avantageusement par une étape de filtrage temporel développée dans la section 3.2.5.

Nous proposons à présent d'étudier l'intérêt d'un filtrage spatial des masques, comme étape supplémentaire, avant la décision de coupure globale.

3.2.4.2 Prétraitement spatial

Que les masques de transition obtenus soient binaires ou à niveaux de gris, leur appliquer un filtrage de type morphologique peut avoir deux conséquences intéressantes dans le cadre de la détection des coupures, mais aussi des transitions géométriques du groupe 2 (section 3.1.2), pour l'étude de leur géométrie (cf. section 3.2.4.3).

La morphologie mathématique [83, 84, 82] met à la disposition du traiteur d'images de nombreux outils efficaces tout particulièrement lorsqu'il s'agit de filtrage et de segmentation

d'images ou de séquences. Basée sur la théorie des ensembles, cette technique de traitement d'images a déjà prouvé son efficacité dans de nombreux problèmes.

Il s'agit ici de mettre en place un filtrage très simple sur les masques de transition. Ce prétraitement spatial a pour but dans un premier temps d'améliorer la distinction entre la présence ou l'absence de transition. Notre choix de filtres morphologiques s'est porté sur la succession d'une fermeture, suivie d'une ouverture, opérateurs dont nous rappelons les définitions en annexe B.

La fermeture a pour effet de combler les trous dans une image ; au contraire, l'ouverture élimine les pics. Ce sont ces deux filtres très simples que nous appliquons successivement aux masques de transition, de façon à faire abstraction, en chaque rectangle du masque, des réponses erronées, tant en positif (détection d'une fausse coupure), qu'en négatif (oubli d'une coupure locale), pour la décision finale à chaque instant. En effet, même dans le cas d'une coupure très franche et brutale dans la scène entière, il est rare que le masque soit entièrement blanc. Certaines zones restent malgré tout très similaires. S'il restera toujours de telles zones dans les masques, elles sont quand même très sporadiques et disséminées parmi de plus larges plages de niveaux de gris élevés. Leur caractère ponctuel ne comporte pas d'indication d'appartenance à une plus grande zone d'un objet en mouvement par exemple. Ces rectangles peuvent donc être supprimés, dans l'optique d'une détection globale de transition, sans nuire à l'information sémantique de la scène.

La figure 3.14 présente le résultat d'une succession ouverture-fermeture par un élément structurant de taille 1, sur un masque binaire et sur un masque à niveaux de gris. En fait les deux filtres morphologiques ne sont pas directement appliqués sur les images des masques de transition, composées de petits rectangles de taille donnée. Afin d'accélerer cette étape, on sous-échantillonne le masque de façon à ne conserver qu'un pixel par rectangle, de niveau de gris, le niveau de gris du rectangle. On ne filtre donc réellement qu'une image de taille $n_x \times n_y$ pixels, où $n_x \times n_y$ correpond au nombre de rectangles du masque original. Pour une image CIF (384×288) et des tailles de rectangles 20×20, on obtient donc 19×14 rectangles, soit une image finale de taille 19×14. C'est sur cette dernière image qu'on applique la succession ouverture-fermeture par un élément structurant carré de taille 1.

A ce stade, notons que les deux opérations filtrage et seuillage du masque permutent [82, 83] :

$$\text{Seuil}((I \oplus \check{B}) \ominus B) \quad = \quad (\text{Seuil}(I) \oplus \check{B}) \ominus B \tag{3.16}$$

où I est l'image originale, B l'élément structurant et Seuil l'opérateur de seuillage. Il est donc équivalent de seuiller un masque à niveaux de gris filtré ou de filtrer le masque seuillé.

L'algorithme 1 résume les diverses étapes mises en œuvre depuis la création du masque de transition à niveaux de gris contenant les valeurs du critère de dissemblance, jusqu'à l'étape finale de filtrage spatial.

L'effet de lissage du masque dû au filtrage permet d'approcher le modèle idéal de la géométrie de la transition. On tend donc dans le cas des coupures vers la succession théorique, illustrée dans la figure 3.2, d'un masque noir lorsqu'on est dans la première prise de vue, puis d'un masque blanc à l'instant de la coupure et enfin à nouveau d'un masque noir dans la deuxième prise de vue. Certaines régions jugées similaires avant le filtrage spatial, telles que la zone de ciel dans la figure 3.7, sont à présent marquées comme appartenant à la transition. Ceci permet d'augmenter le contraste entre les valeurs de critère de dissemblance en dehors

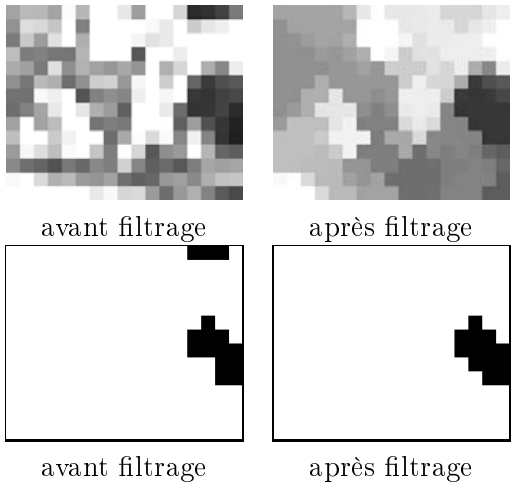


FIG. 3.14: Prétraitement spatial du masque de transition : on présente dans la première ligne un masque à niveau de gris avant et après filtrage, et dans la deuxième ligne sa version binaire également avant et après filtrage. Le masque correspond à la transition de type coupure précédemment utilisée dans la figure 3.7.

Algorithme 1 Algorithme de filtrage spatial du masque de transition.

Pré-condition : M_t masque de transition à niveaux de gris à l'instant t.

Pré-condition : n_x nombre de rectangles contenus en X dans M_t.

Pré-condition : n_y nombre de rectangles contenus en Y dans M_t.

Sous-échantillonnage de M_t de façon à se ramener à une image de taille $n_x \times n_y$.

Eventuellement binarisation de M_t.

Filtrage morphologique de M_t par ouverture-fermeture de taille 1.

et pendant une coupure, i.e. les valeurs de pic sont plus hautes au niveau des transitions et le niveau de "bruit" baisse à l'intérieur d'une prise de vue.

Outre le cas particulier de détection des coupures, le filtrage spatial des masques de transition permet également une étude plus solide de la géométrie des autres transitions classifiées dans le groupe 2. Cette étape est essentielle dans cette deuxième utilisation, pour une classification correcte des transitions géométriques apparaissant dans un document vidéo, que nous détaillons dans la section 3.2.4.3.

Pour la détection des coupures, le filtrage spatial proposé augmente le rapport pic/bruit, sans véritable amélioration des résultats finaux, déjà d'un niveau élevé. Pour cette raison, le filtrage spatial n'a été retenu que dans le cadre d'une étude véritable et approfondie d'une transition donnée, et non dans le processus complet de détection des coupures. Cette étape de filtrage n'est donc pour l'instant pas incluse dans l'algorithme de détection de coupures. Elle est par contre utilisée dans un outil d'étude de la géométrie des transitions autres que les coupures, que nous détaillons maintenant. En ce sens, la section suivante constitue une disgression dans l'exposé linéaire des différentes étapes de l'algorithme de détection des coupures, qui se poursuit véritablement avec l'étape de filtrage temporelle, détaillée au paragraphe 3.2.5, auquel le lecteur uniquement intéressé par la détection de coupures pourra directement se reporter. Dans le processus bâti jusqu'à présent, c'est cependant à cet instant, après un filtrage spatial des masques, qu'une retombée annexe, utile à la classification de transitions plus complexes (groupe 2), apparaît.

3.2.4.3 Géométrie des transitions

Cette section a pour but de prouver qu'il est possible de classifier les transitions apparaissant dans un document vidéo en fonction de simples caractéristiques extraites de la géométrie du masque de transition. Quant à l'intérêt d'une telle classification dans le cadre de l'indexation de documents vidéo, l'apport sémantique supplémentaire fourni par l'appartenance d'une transition à telle ou telle classe n'est généralement pas primordial. Cependant on peut tout de même souligner dès à présent un exemple où l'appartenance des transitions à une catégorie particulière contient une information sémantique intéressante dans la construction hiérarchique du document. Cet exemple, qui sera repris de façon plus approfondie dans le chapitre 7, section 7.4.1, concerne l'utilisation des fondus dans certains journaux télévisés, comme délimitation des reportages. La détection de ces transitions et leur classification en tant que fondu permet donc indirectement d'établir des liens entre les prises de vue d'un même reportage et d'obtenir un découpage cette fois-ci en scènes (cf. définition 2, section 2.3.1).

Comme nous l'avons déjà évoqué, les masques de transition donnent accès à la géométrie de la transition, une fois l'étape de filtrage spatial terminée (cf. section 3.2.4.2). En fonction des caractéristiques que l'on est capable d'extraire de cette géométrie, il est possible de classifier les diverses transitions rencontrées dans un document vidéo. Dans cette section, nous mettons donc en œuvre quelques algorithmes très simples d'extraction de caractéristiques géométriques et de comparaison de ces caractéristiques avec des modèles théoriques des transitions. Il ne s'agit ici que de fournir un aperçu de l'intérêt du masque de transition dans la classification de quelques transitions. Dans une poursuite éventuelle de ce travail, l'amélioration et l'étendue de cet algorithme à d'autres catégories de transitions fourniraient un thème de recherche intéressant.

L'étude de la géométrie de la transition, comme le pré-filtrage morphologique d'ailleurs, peuvent s'appliquer sur des masques binaires ou bien à niveaux de gris (cf. section 3.2.4.1).

Nous proposons ici un modèle d'étude sur des masques binaires. Bien sûr ce choix simplifie l'algorithme et pose toujours le problème supplémentaire qui est l'étape de seuillage et le choix de la valeur du seuil. Cependant notons dès à présent que dans les deux exemples que nous prenons dans cette section, la même valeur de seuil a été conservée pour la binarisation.

En outre, il s'agit d'une étude de la géométrie du masque de transition, dans l'hypothèse où les limites de la transition ont déjà été extraites, ce qui, bien sûr, nous ramène au problème de la détection des transitions.

Si on suppose qu'une transition a été détectée entre deux instants donnés et que les masques de transition ont été binarisés, nous proposons d'extraire un ensemble de mesures des masques et de comparer ces mesures avec une base de courbes théoriques des modèles idéaux de transitions. On classifie ainsi la transition à étudier en fonction de la corrélation de ses mesures avec celles des modèles idéaux. Un inconvénient majeur, mais inhérent à cette technique, est la nécessité de modéliser le maximum de transitions existantes, tout en conservant le risque de ne pas réussir à classifier la transition à étudier si son modèle n'est pas dans la base a priori.

La classification proposée est pour l'instant restreinte puisque seuls les différents modèles de balayages horizontaux, verticaux et diagonaux et pages tournées sont explicités, avec toutefois la restriction suivante : aucune distinction ne sera faite entre le balayage diagonal et la page tournée, ces deux transitions étant confondues du fait de la modélisation adoptée. Toute autre transition "géométrique" est donc placée dans une catégorie "autres". Cependant une fois le principe de comparaison bâti, seul l'établissement des modèles théoriques reste à faire pour ajouter une nouvelle classe de transition.

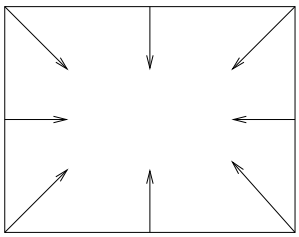


FIG. 3.15: Ensemble des directions de mesure des traversées pour une phase donnée, dans le cadre de l'étude de la géométrie du masque de transition : quatre directions principales (horizontal et vertical), et quatre directions supplémentaires éventuelles (diagonal).

Avant d'énumérer les mesures extraites des masques, précisons que, dans un but de simplification des modèles de transition, on ne calcule pas ces mesures sur le masque filtré à un instant t, mais sur l'union des masques de transition entre l'instant t_0, début de la transition, et l'instant t. L'utilisation de ce supremum sur les masques a deux actions. Il permet tout d'abord de conforter l'information géométrique, puisqu'on cumule cette information dans le temps. On espère ainsi s'affranchir des erreurs locales et ponctuelles (au sens instantanées) de détection dues au bruit. En outre, cette opération d'union des masques permet de classifier des transitions un peu plus sophistiquées. Prenons à nouveau l'exemple du balayage gauche-droite. Sa forme la plus simplifiée et les masques correspondants ont déjà été décrits : en amont d'une ligne verticale, on conserve la première prise de vue et en aval, la deuxième prise de vue apparaît. Les masques sont alors illustrés dans la figure 3.4. Un balayage un peu plus sophistiqué consiste toujours à conserver la première prise de vue en amont, mais à faire apparaître la seconde prise de vue avec un effet supplémentaire de mouvement, comme si la nouvelle prise

de vue était plaquée sur la face d'un cube en rotation (cf. figure 3.16).

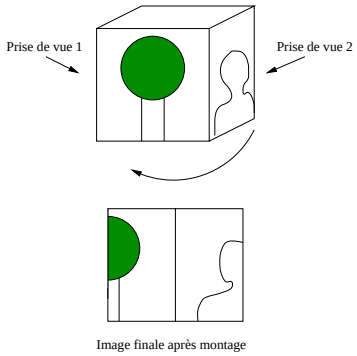


FIG. 3.16: Balayage avec effet de cube en rotation.

Les masques ne sont plus alors simplement représentés par une bande blanche de largeur constante se déplaçant de gauche à droite, mais par une bande blanche apparaissant à gauche et croissant de façon à couvrir toute l'image. En réalité cette nouvelle transition fait plutôt partie du groupe 3, puisque les pixels passent par un état de transition. Cependant son modèle géométrique s'apparente à celui d'un balayage. Le fait de prendre le supremum permet de la classifier dans la catégorie des balayages, pour identifier son modèle géométrique.

L'utilisation de l'union des masques a cependant l'inconvénient de conduire tôt ou tard à un masque entièrement blanc qui n'évolue plus. Pour cette raison il est entre autres primordial de n'étudier la géométrie que sur la durée de la transition, cette durée ayant été déterminée auparavant.

Afin de décrire certaines mesures calculées sur le masque, nous donnons la définition de la notion de traversée.

Définition 14. Traversée (*intercept) La traversée dans une image binaire correspond à la plus grande distance qu'il est possible de parcourir en ligne droite dans une direction donnée, en partant du bord de l'image, tout en restant dans une même phase.*

Une illustration de cette notion est fournie dans la figure 3.17.

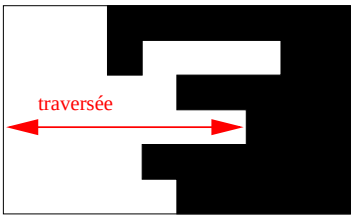


FIG. 3.17: Exemple de traversée horizontale gauche pour une image binaire.

Les mesures effectuées sur les masques sont d'une grande simplicité et suffisent à caractériser les diverses directions de balayage. On calcule ainsi :

- l'évolution de la surface du masque (% de rectangles blancs) au cours de la transition ;
- l'évolution de la traversée horizontale en partant de la gauche du masque pour chacune des deux phases (rectangles blancs et noirs) ;

– idem pour les traversées horizontales droite, verticales haut et bas toujours pour les deux phases.

Eventuellement d'autres traversées telles que les diagonales en haut à gauche, en haut à droite, en bas à gauche, en bas à droite peuvent être ajoutées, au prix d'un temps d'exécution légérement plus important. Dans ce qui suit ces dernières mesures n'ont pas été prises en compte dans la mesure où les tests effectués sur la base de données ont prouvé qu'elles n'apportaient pas d'amélioration de la reconnaissance des modèles de balayages et de pages tournées déjà suffisamment discriminés par les mesures ci-dessus.

Dans chacune des phases, quatre traversées sont donc calculées (cf. figure 3.15), ce qui fournit au total pour un masque donné 9 mesures à suivre pendant l'évolution d'une transition.

Le calcul des traversées pour chaque masque se trouve à nouveau accéléré du fait de l'utilisation des masques sous-échantillonnés. Quatre balayages d'images sous-échantillonnées (typiquement 19×14 pixels dans le cas d'images CIF au départ) suffisent pour obtenir ces 9 mesures.

Sur la durée de la transition, on obtient au final 9 courbes d'évolution. En parallèle, les courbes idéales de chacun des modèles à tester sont construites et on calcule la corrélation moyenne entre ces courbes idéales et les courbes expérimentales. La figure 3.18 contient un échantillon des courbes idéales pour un balayage horizontal de gauche à droite et les courbes expérimentales correspondantes d'une transition réelle. Cette transition est présentée dans la figure 3.19 sous la forme, d'une part, des images originales, et d'autre part, des unions des masques de transition.

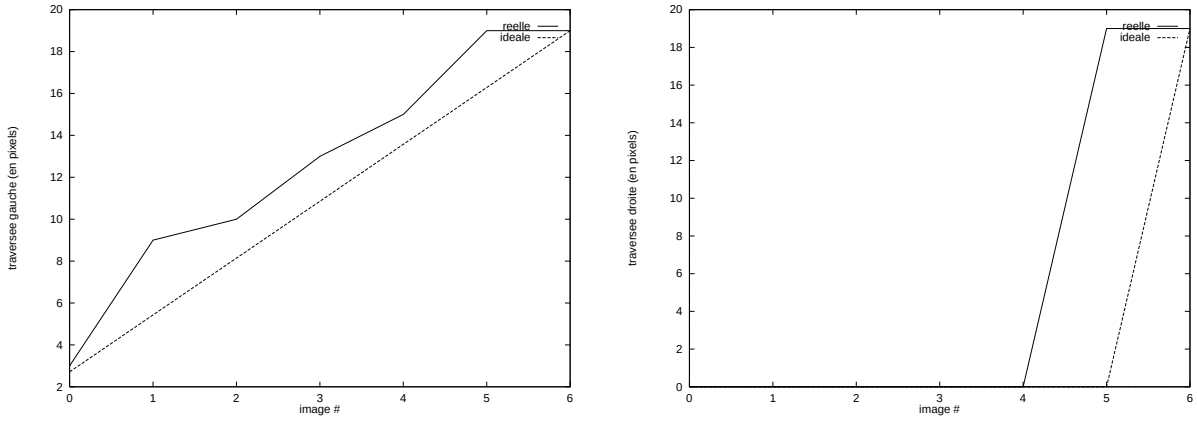


FIG. 3.18: Exemples de courbes d'évolution des mesures caractéristiques de la géométrie d'une transition de type balayage horizontal de gauche à droite (cf. figure 3.19 pour la séquence originale). On présente sur chaque figure la courbe expérimentale et la courbe idéale, des traversées gauche et droite, dans la phase blanche.

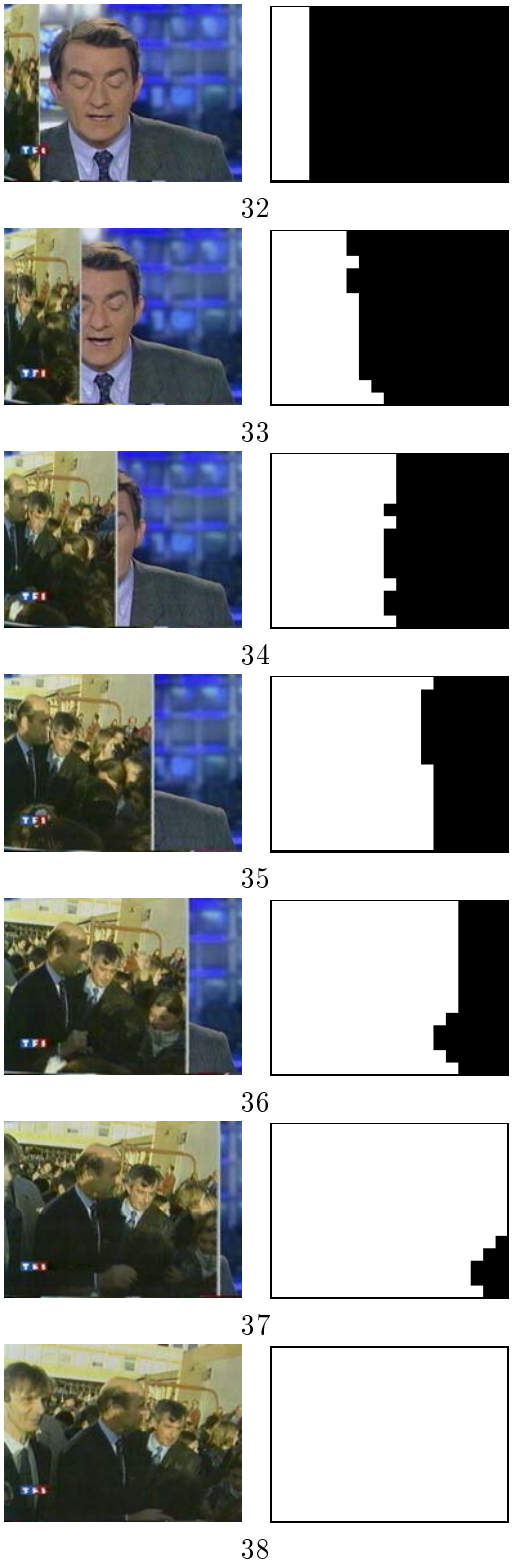


FIG. 3.19: Exemple d'un balayage réel horizontal de gauche à droite et des unions des masques de transition correspondants. La transition a lieu sur l'intervalle ouvert $]31, 38[$.

Le critère de corrélation choisi est le critère ZNCC (*zero-mean normalized cross-correlation*) dont on rappelle la définition :

Définition 15. Mesure de corrélation ZNCC *Soit deux courbes de longueur L (i.e. contenant L points), on note :*

$$S_1 = \sum_{t=0}^{L} C_1(t) \qquad S_{11} = \sum_{t=0}^{L} C_1(t) \times C_1(t)$$
$$S_2 = \sum_{t=0}^{L} C_2(t) \qquad S_{22} = \sum_{t=0}^{L} C_2(t) \times C_2(t)$$
$$S_{12} = \sum_{t=0}^{L} C_1(t) \times C_2(t)$$

La mesure de corrélation ZNCC entre les deux courbes C_1 et C_2 est donnée par :

$$Corr(C_1, C_2) = \frac{L \times S_{12} - S_1 S_2}{\sqrt{(L \times S_{11} - S_1 S_1)(L \times S_{22} - S_2 S_2)}} \tag{3.17}$$

Ce critère fournit en général de bons résultats et possède en outre l'avantage de mettre facilement en correspondance deux courbes similaires mais décalées l'une par rapport à l'autre.

La meilleure corrélation (i.e. la corrélation maximale) entre les courbes expérimentales et les courbes idéales de chaque modèle n'est retenue, et son modèle de transition sélectionné, que si sa valeur dépasse le seuil de 0.5. Dans le cas contraire, aucun modèle n'est retenu.

L'algorithme 2 résume par ailleurs la technique mise en œuvre de façon complète.

Comme nous le précisions en début de section, il s'agit ici de proposer des idées de classification des transitions rencontrées dans un document vidéo. La méthode présentée dans ce paragraphe, bien qu'à ses prémices, a cependant prouvé son efficacité pour les deux uniques tests effectués de reconnaissance, l'un sur le balayage présenté dans la figure 3.19 et l'autre sur une transition de type page tournée. Ces transitions ont été correctement classifiées, avec des mesures de corrélation respectives de 0.85 et 0.92. Le tableau 3.20 résume en outre l'ensemble des corrélations obtenues pour chacune de ces deux transitions avec l'ensemble des modèles idéaux existants dans l'algorithme. Notons les valeurs véritablement discriminantes des corrélations dans chaque cas, ainsi que leur symétrie : deux modèles idéaux symétriques par rapport à la transition réelle (par exemple, les balayages de haut en bas ou de bas en haut pour le balayage gauche-droite réel) ont des valeurs de corrélation proches (0.29 et 0.31).

Bien sûr l'échantillon de test est faible mais ces résultats sont malgré tout très encourageants, pour encore une fois un algorithme très simple et rapide.

Notre disgression dans l'enchaînement des étapes de l'algorithme de détection des coupures étant terminée, nous continuons par l'exposé de l'étape suivante de préfiltrage temporel.

3.2.5 Prétraitement temporel

A ce stade de l'algorithme, c'est-à-dire après la construction du masque de transition à niveaux de gris et après une étape éventuelle de filtrage spatial de ce masque, on dispose donc d'une valeur de critère unique par masque, correspondant au volume du masque.

Il est alors possible de tracer la courbe d'évolution temporelle de ce volume, que nous appellerons également *critère global de dissemblance*. Quel que soit le critère local de dissemblance choisi, cette courbe présente des valeurs élevées, ou pics, aux instants de transition.

Algorithme 2 Algorithme de classification d'un modèle géométrique de transition.

Pré-condition : $]t_0, t_1[$ transition

Pré-condition : M_t masque de transition sous-échantillonné et filtré à l'instant t.

pour t entre t_0 et t_1 **faire**

Calcul des 9 mesures de surface et de traversées sur M_t.

fin pour

Construction des 9 courbes d'évolution $\{E_i\}_{i\in[1;9]}$ sur $[t_0, t_1]$.

pour tout m, modèle de transition idéal **faire**

Construction des 9 courbes d'évolution idéales $\{I_i\}_{i\in[1;9]}$ sur $[t_0, t_1]$.

$C_m = \frac{1}{9}\sum_{i\in[1;9]} Corr(E_i, I_i)$

fin pour

$M = argmax_{m\in\text{modèles}} C_m$

si $C_M > 0.5$ **alors**

retourner M

sinon

retourner aucun modèle

fin si

		Transitions réelles	
		Balayage →	Page tournée ↖
Modèles idéaux	Balayage →	**0.85**	0.39
	Balayage ←	0.70	0.51
	Balayage ↓	0.29	0.44
	Balayage ↑	0.31	0.49
	Page tournée ↘	0.55	0.80
	Page tournée ↗	0.54	0.85
	Page tournée ↙	0.51	0.87
	Page tournée ↖	0.50	**0.92**

FIG. 3.20: Mesures de corrélation obtenues pour les deux transitions réelles balayage gauche-droite et page tournée du coin en bas à droite, avec les huit modèles idéaux de balayages et de pages tournées.

Pour la distance moyenne pixel à pixel dans l'espace couleur RGB (cf. section 3.2.1), la courbe d'évolution temporelle présente des pics d'une hauteur parfois similaire au niveau de bruit moyen, c'est-à-dire aux valeurs de critère hors transition. Extraire ces pics et par conséquent les transitions qu'ils représentent n'est pas immédiat sur la courbe brute. Pour cette raison, une étape de filtrage, toujours morphologique mais cette fois-ci temporel, est ajoutée, afin d'augmenter le rapport signal à bruit et d'extraire les pics plus facilement. L'algorithme complet de détection est alors schématisé dans la figure 3.21, avec la possibilité de rajouter l'étape de filtrage spatial du masque.

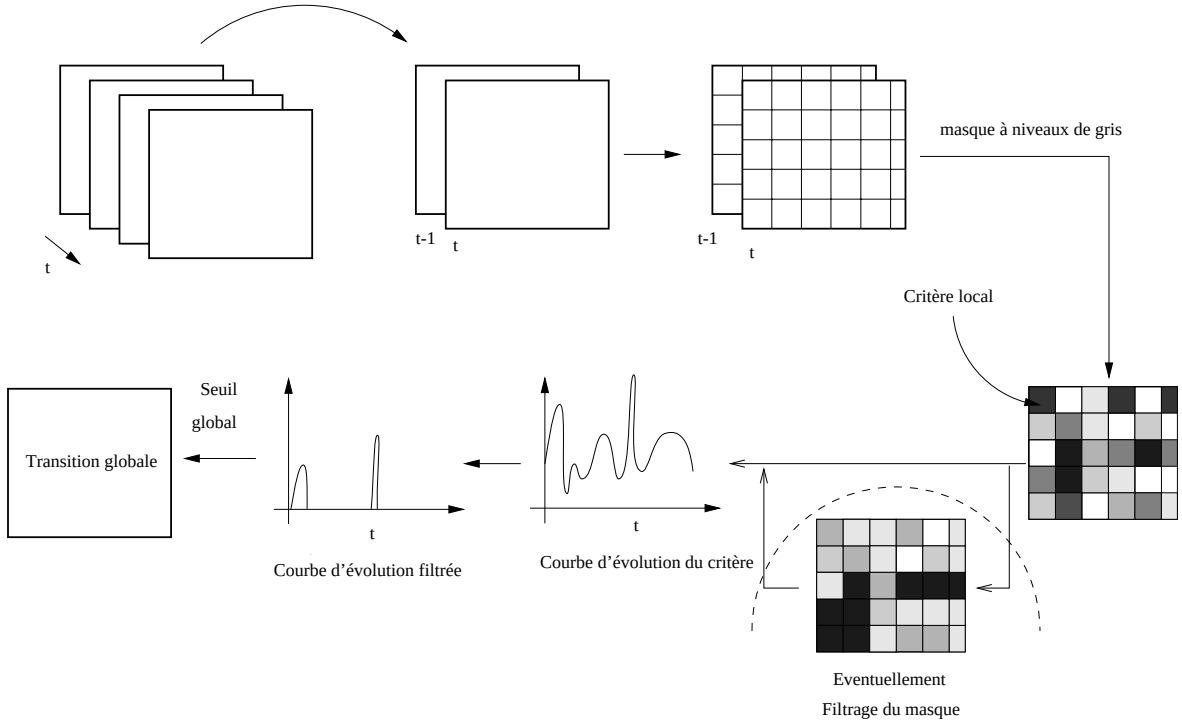


FIG. 3.21: Algorithme complet de détection de coupures.

Nous ne reviendrons pas sur le choix du critère local de dissemblance (section 3.2.1.2) ; cependant rappelons rapidement l'existence d'autres critères aboutissant à des courbes d'évolution plus discriminantes pour les pics de transition telles que la mesure de χ^2. Notre volonté de bâtir un outil de détection extrêmement simple et rapide nous a toutefois conduit au choix de la distance couleur pixel à pixel, moins discriminante, mais contrebalancée par cette étape de filtrage temporel. Avant d'étudier l'amélioration apportée aux courbes, nous présentons les outils morphologiques utilisés.

3.2.5.1 Chapeau haut-de-forme et chapeau haut-de-forme inf

En annexe B, nous proposons la définition du *chapeau haut-de-forme blanc* comme étant le résidu le plus simple qu'il est possible d'extraire de l'opérateur d'ouverture, par simple différence avec l'image originale.

Puisque l'ouverture consiste à éliminer tous les pics et points isolés d'une image, ce nouvel opérateur a pour résultat de ne conserver que ces résidus. Sur une image à niveaux de gris,

le chapeau haut-de-forme extrait tous les détails fins et blancs ; sur une courbe monodimensionnelle, il a pour action de sélectionner les pics plus étroits que l'élément structurant. Un exemple de résultat d'un tel opérateur est proposé dans la figure 3.22.

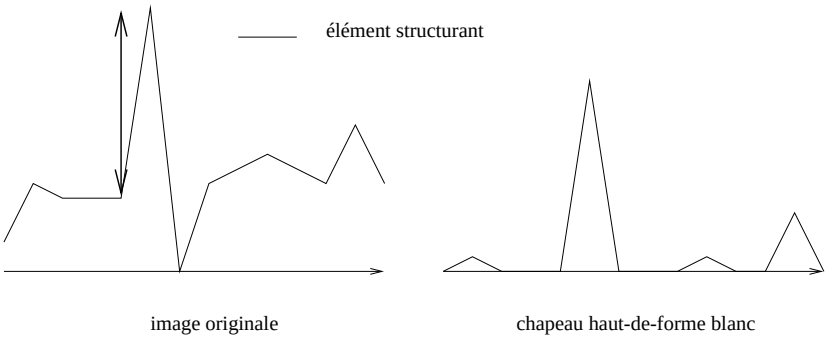


FIG. 3.22: Effet du chapeau haut-de-forme blanc sur une courbe monodimensionnelle. L'élément structurant est un segment de taille trois pixels.

Le chapeau haut-de-forme blanc restitue les pics avec une hauteur correspondant exactement au résidu de l'ouverture. Sur une courbe monodimensionnelle, un pic a pour support un intervalle de trois pixels : un point de valeur élevée, entouré de deux points aux extrémités de l'intervalle, de valeurs plus faibles. Dans ce cas, le chapeau haut-de-forme blanc correspond en fait à la hauteur minimale entre le point le plus haut sous le pic et l'une des deux extrémités, dans le cas d'une courbe monodimensionnelle.

On obtient alors la formulation mathématique suivante, qui correspond exactement à la différence de la courbe originale et de son ouvert :

$$\forall x \in f,\ \mathrm{TH}\ (x) = f(x) - max \begin{pmatrix} min(f(x-2), f(x-1), f(x)), \\ min(f(x-1), f(x), f(x+1)), \\ min(f(x), f(x+1), f(x+2)) \end{pmatrix} \tag{3.18}$$

où f est une courbe monodimensionnelle, B l'élément structurant linéaire de taille 3 pixels et d'origine centrée :

Dans le cas extrême d'un fort déséquilibre entre les deux valeurs des extrémités de l'intervalle, le résultat du chapeau haut de forme blanc est faible. Cette remarque conduit à l'introduction d'un nouvel opérateur morphologique, *le chapeau haut-de-forme inf* qui restitue, pour un pic, la hauteur non pas entre le point le plus haut et la plus haute des extrémités, mais entre le point le plus haut et la plus faible des extrémités. Cet opérateur, illustré dans la figure 3.23, se formalise de la façon suivante, toujours pour le même élément structurant B :

$$\forall x \in f,\ \mathrm{THI}\ (x) = \begin{cases} 0 \text{ si } f(x) \text{ est atteint par l'élément structurant} \\ f(x) - min(f(x-1), f(x), f(x+1)) \text{ sinon} \end{cases} \tag{3.19}$$

Cet opérateur est en essence proche de celui, bidimensionnel, développé dans [15]. Notons que, comme pour le chapeau haut-de-forme classique, $f(x)$ n'est pas atteint par l'élément structurant lorsque le pic en x de la courbe originale est plus étroit que l'élément structurant.

Ces deux opérateurs étant définis, étudions à présent leur effet sur les courbes de critères calculées à partir des masques de transition.

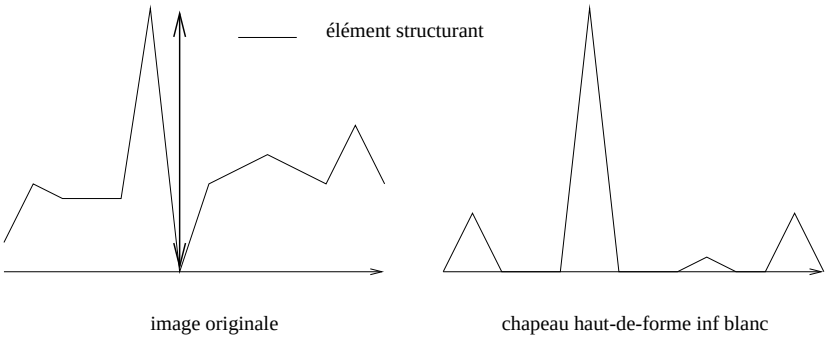


FIG. 3.23: Effet du chapeau haut-de-forme inf blanc sur une courbe monodimensionnelle. L'élément structurant est un segment de taille trois pixels.

3.2.5.2 Effet sur les courbes

Le calcul du critère global de dissemblance, sous forme de volume du masque de transition, fournit des courbes d'évolution temporelle du type de celles présentées dans la figure 3.25. L'une comme l'autre présentent des pics aux instants de transition, mais ces pics sont plus ou moins élevés et, s'il est envisageable d'en extraire la majorité à l'aide d'un simple seuillage (tout particulièrement pour la courbe de gauche), cette technique ne permet pas d'atteindre un taux de détection de 100%, du fait du bruit dû à des mouvements rapides d'objets en gros plan, ou à la mauvaise qualité d'acquisition des images, comme c'est particulièrement le cas de la courbe de droite ; le document vidéo correspondant, de très mauvaise qualité, représente une course de kart, contenant beaucoup de mouvement et des changements entre des scènes similaires. On fournit un exemple d'image de cette séquence dans la figure 3.24.

FIG. 3.24: Image de la séquence *course de kart*.

En outre, que ce soit pour la courbe de gauche ou pour celle de droite, la variation dans le choix du seuil à appliquer est restreinte si on désire conserver une qualité de détection stable. En d'autres termes, une faible variation du seuil peut introduire une chute importante du nombre de transitions détectées. Pour cette raison, l'application du chapeau haut-de-forme inf sur les courbes est essentielle.

Pour les deux exemples précédents, on fournit les courbes filtrées temporellement dans la figure 3.26. L'effet de réhaussement des pics dû au chapeau haut-de-forme inf apparaît tout particulièrement au niveau du troisième pic de la courbe de droite.

Cette étape de filtrage temporel terminée, il apparaît de façon évidente que la marge de placement du seuil global sur la courbe, de façon à conserver un même niveau de résultats

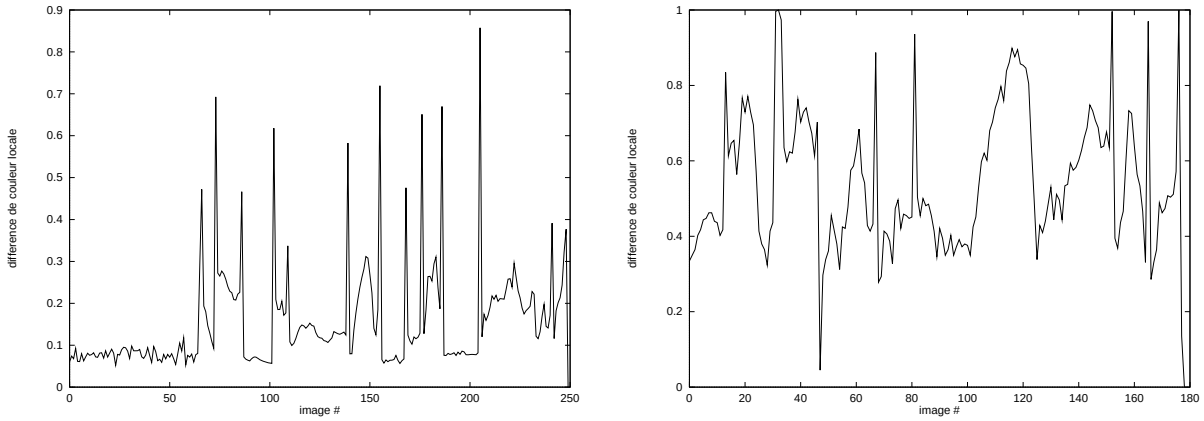


FIG. 3.25: Exemples de courbes d'évolution temporelle du critère de dissemblance, avant filtrage temporel. Courbe de gauche : séquence *travaux*, 40 s, format CIF, fréquence de 5 Hz. Courbe de droite : séquence *course de kart*, 30 s, format CIF, fréquence de 5 Hz.

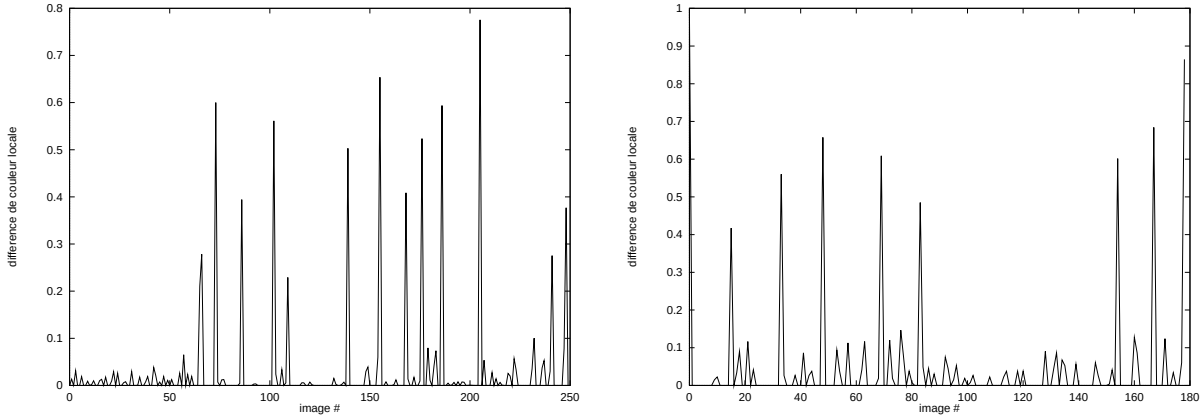


FIG. 3.26: Exemples de courbes d'évolution temporelle du critère de dissemblance, filtrées par un chapeau haut-de-forme inf blanc. Courbe de gauche : séquence *travaux*. Courbe de droite : séquence *course de kart*.

(un même nombre de bonnes et fausses détections), est augmentée. Mais discutons à présent le choix de la valeur de ce seuil.

3.2.6 Seuillage

Nous nous étendrons peu sur cette étape somme toute triviale : une fois le seuil global choisi, tous les pics au-dessus du seuil sur la courbe d'évolution sont détectés comme étant des transitions de type coupure. Grâce au filtrage temporel, de petites variations du seuil n'entraînent pas ou peu de modification des résultats. La valeur du seuil est fixée expérimentalement à 0.2 (20% du volume total du masque), à partir des documents vidéo test à notre disposition. Pour tous ces documents, cette même valeur est conservée avec des résultats, détaillés dans la section 3.2.8, très satisfaisants. On a donc une fois de plus affaire à un paramètre de notre algorithme peu dépendant des données.

Cependant, il est malgré tout toujours positif de chercher à supprimer le maximum de paramètres d'un algorithme. Dans cette optique, la mise au point d'une technique de sélection automatique du seuil global représente une piste d'amélioration et de simplification de l'algorithme. Un choix de seuil adaptatif, déterminé en fonction d'une étude des valeurs du critère de dissemblance (valeurs maximale, minimale, moyenne, médiane, etc.), permettrait ainsi d'ajuster sa valeur au contenu de chaque séquence (présence de mouvement, bruit). Le fait qu'il soit déjà possible de sélectionner une seule et même valeur pour toute séquence, tout en obtenant des taux de détection de transitions très élevés, laisse en outre supposer que cette amélioration par sélection automatique du seuil pour chaque séquence devrait être aisément réalisable.

Dans l'attente de cette amélioration, la section suivante est l'occasion de faire le point sur l'ensemble des paramètres dont dépend actuellement l'algorithme de détection de coupures proposé.

3.2.7 Paramètres de l'algorithme

En plus des seuils local et global, il faut rajouter trois paramètres à cet algorithme que sont le pas d'échantillonnage spatial, la fréquence de comparaison des images et la taille des rectangles des masques de transition. On obtient ainsi un total maximum de cinq paramètres, dont dépendent les performances de l'algorithme.

Seuil local. Lors de la comparaison entre les masques à niveaux de gris et les masques binaires, ces derniers nécessitant un seuil local, il est apparu qu'on pouvait conserver l'information numérique sous forme de différentes valeurs de critère (différents niveaux de gris) dans chaque rectangle. On substitue alors le volume des masques à niveaux de gris au pourcentage de rectangles blancs des masques binaires. On s'affranchit ainsi d'un paramètre sur cinq. Rappelons toutefois que cette transformation consiste à se rapprocher au final d'un calcul global du critère sur toute l'image. D'autre part, dans le cadre de l'étude de la géométrie des transitions (section 3.2.4.3), ou de l'extraction de changements locaux (section 6.2), le passage à un masque binaire simplifie les procédures mises en œuvre.

Dans le cas d'une binarisation du masque de transition (et plus particulièrement pour ces deux derniers traitements) plusieurs valeurs de seuil ont été testées ; les résultats de ces tests sur une base assez importante de documents vidéo ont prouvé que ce seuil était peu dépendant des données. Une seule et même valeur fixée expérimentalement à 70 a

finalement été utilisée pour toute la série de documents. De petites variations autour de ce seuil n'entraînent pas de changement notable de résultats, comme le montre la figure 3.27.

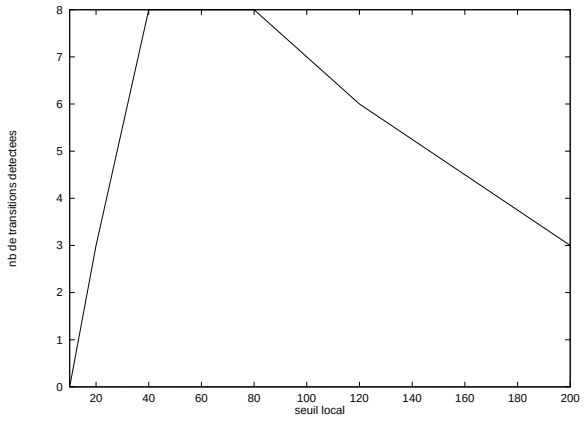


FIG. 3.27: Courbe d'évolution du nombre de prises de vue correctes détectées (sans fausses alarmes) en fonction du seuil local, dans le cas de masques binaires. Séquence originale *course de kart*, comprenant 9 coupures.

Seuil global. De même, une seule et même valeur de seuil global (0.2) a été conservée pour toutes les séquences test, avec la même conclusion de peu de dépendance avec les données et de possibilités de petites variations du seuil sans modification des résultats. La valeur de seuil choisie est extraite sans ambiguïté au vu des courbes d'évolution obtenues après filtrage (cf. figure 3.26, par exemple) : le bruit est de l'ordre de 0.1 tandis que les pics sont d'un niveau plus élevé.

Pas d'échantillonnage spatial. Le critère de dissemblance est calculé entre deux images sous-échantillonnées par deux dans les deux directions X et Y. Aucun évaluation du comportement de l'algorithme pour des pas d'échantillonnage plus grand n'a été menée. Nul doute qu'il s'agisse là d'une piste possible dans la diminution du temps de calcul de l'algorithme. Il s'agirait alors de trouver le pas d'échantillonnage fournissant le meilleur compromis entre la réduction du temps d'exécution et l'étude de la géométrie des masques.

Taille des rectangles. En ce qui concerne la taille des rectangles, elle a été fixée, toujours de façon empirique, à 20×20 pixels pour toutes les séquences, c'est-à-dire à environ 5% de la taille des images si elles sont au format CIF. Une fois encore des tailles s'échelonnant entre 10×10 et 140×140 pixels ont été testées avec les résultats suivants :

- avec une utilisation des masques à niveaux de gris, peu de variations des résultats avec l'augmentation de la taille des rectangles, pas d'augmentation du nombre de fausses alarmes, pas d'augmentation du nombre de transitions perdues. Ceci confirme en outre la très proche similarité entre les deux critères local à niveaux de gris et global, dans le cas où aucun filtrage spatial n'est effectué ;
- avec une utilisation des masques binaires, augmentation importante du nombre de fausses alarmes avec la taille des rectangles ;
- légère augmentation du temps de calcul pour les deux critères avec la diminution de la taille des rectangles.

- meilleure classification des transitions géométriques pour des petites tailles de rectangles ;
- obtention de taux de détection de vraies transitions très légèrement plus élevés avec les masques binaires qu'avec les masques à niveaux de gris, quelle que soit la taille des rectangles ;
- meilleure sélection des images représentatives de chaque prise de vue en niveaux de gris qu'en binaire (le choix des images clés sera détaillé dans le chapitre 4, section 4.3.3) : un nombre beaucoup plus important d'images clés est sélectionné en binaire sans apport d'information pertinente supplémentaire ;

Cette dernière remarque, non liée à la taille des rectangles, appuie par ailleurs notre choix de conserver les masques à niveaux de gris, dans le processus de détection de transitions.

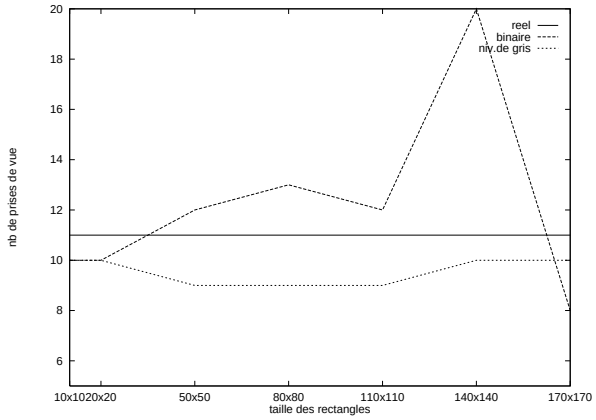


FIG. 3.28: Courbe d'évolution du nombre de prises de vue détectées (fausses alarmes comprises) en fonction de la taille des rectangles du critère global, soit binaire, soit à niveaux de gris. Séquence originale *course de kart*, comprenant 11 prises de vue réelles, séparées par 9 coupures et un fondu. Pour le critère local binaire, le nombre de fausses alarmes croît avec la taille des rectangles.

Les deux premiers points sont illustrés par les courbes des figures 3.28 et 3.29, établies pour la séquence *kart*, sans aucun doute une des plus difficiles de la base de données.

Au vu de ces courbes, mais également pour tenir compte des tailles d'événements tels que l'apparition d'incrustations ou de bandeaux, de l'étude de la géométrie des masques et des temps de calcul, la taille de 20×20 pixels est le meilleur choix possible, ce qui est confirmé sur les autres séquences. La figure 3.30 propose des exemples des courbes d'évolution obtenues pour diverses tailles de rectangles, dans le cas de masques binaires.

Fréquence. La fréquence d'échantillonnage et de comparaison des images du fichier vidéo constitue le quatrième et dernier paramètre de notre algorithme.

Dans le but de limiter les temps de traitement, il est en effet tout à fait envisageable de ne pas comparer les images au rythme de 25 images/seconde mais à une fréquence plus faible, étant entendu qu'à 25 images/seconde un grand nombre de comparaisons sont inutiles, i.e. ne conduisent pas à la détection de transition, puisque les prises de vue ont en moyenne des durées minimales de 2 secondes (selon le type de document

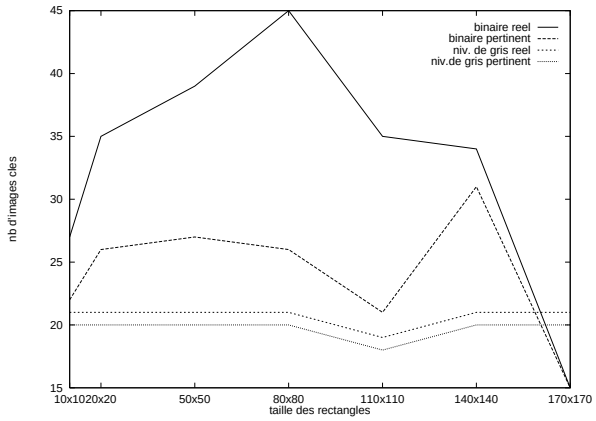


FIG. 3.29: Courbe d'évolution du nombre d'images clés sélectionnées en fonction de la taille des rectangles du critère global, soit binaire, soit à niveaux de gris. Pour chaque critère, on étudie le nombre d'images clés extraites par l'algorithme (réel) et, parmi ces images extraites, le nombre d'images clés contenant une information supplémentaire (pertinent). Séquence originale *course de kart*, comprenant 11 prises de vue réelles, séparées par 9 coupures et 1 fondu. Pour le critère local binaire, le nombre d'images clés extraites augmente avec la taille des rectangles, sans apport d'information supplémentaire.

vidéo traité). Quelques remarques permettent alors de guider le choix de la fréquence de comparaison :

- Plus la fréquence est faible, plus le temps de calcul nécessaire à l'évaluation du critère de dissemblance entre images sur la séquence entière est faible (moins de comparaisons sont effectuées).
- Les artéfacts que l'on pouvait craindre à partir d'une fréquence plus basse apparaissent : des faux positifs sont détectés aux zones de fort mouvement, amplifié par la fréquence ; trop de transitions sont perdues lorsque la fréquence est trop faible.
- Plus la fréquence est faible, plus les masques de transition ont tendance à être blancs. Pour de faibles fréquences, la nature des transitions n'est en effet pas conservée et elles apparaissent toutes au final comme des coupures, ce qui peut apparaître comme un avantage : un seul algorithme de détection de coupures est nécessaire ; ou un inconvénient : on perd toute information de début/fin réels de la transition, ainsi que sur son modèle géométrique.
- A partir du moment où on ne traite plus toutes les images successivement, la position précise des coupures est perdue.
- Pour être corrigés, les deux points précédents (perte de la nature et de la position précise des transitions) nécessitent de réexaminer les passages où il y a eu détection de transitions. Il reste donc à établir comment cette deuxième étape de raffinement peut être menée à bien en pratique. D'autre part il est nécessaire d'évaluer si la perte occasionnée en temps de calcul est plus ou moins importante que le gain réalisé par l'utilisation d'une fréquence plus basse.
- Enfin, l'algorithme proposé a des temps de calcul très faibles pour la fréquence élevée de 5 images/seconde. Dans ces conditions, il serait dommage de ne pas garder

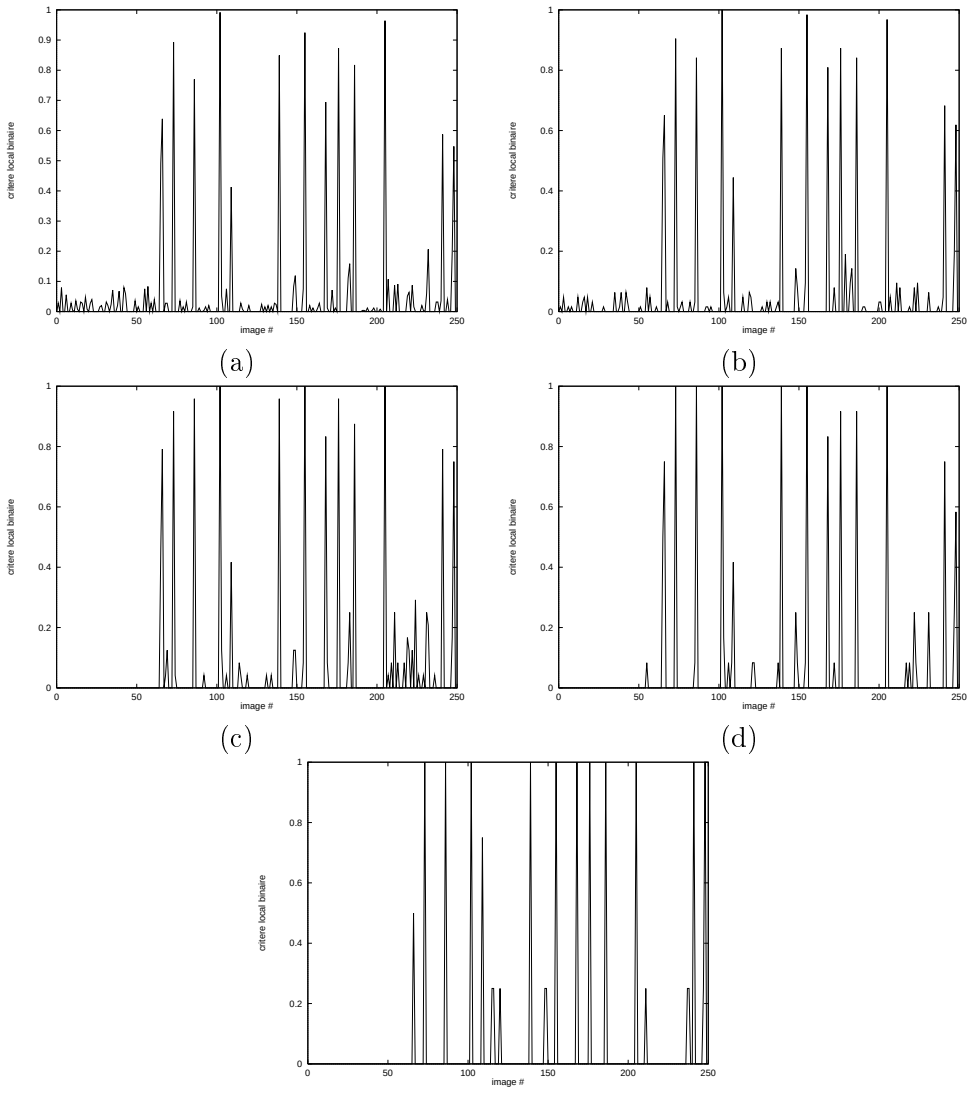


FIG. 3.30: Courbes d'évolution du critère de détection de coupures, dans le cas de masques binaires pour diverses tailles de rectangles : (a) 20×20 ; (b) 40×40 ; (c) 60×60 ; (d) 80×80 ; (e)140×140. Séquence originale *travaux*, seuil global $= 0.2$; seuil local $= 70$.

cette fréquence, qui fournit le plus d'informations dès le premier traitement. Il serait même envisageable, pour des applications nécessitant une localisation très précise des instants de transition, d'effectuer les traitements à 24 images/seconde, la simplicité de l'algorithme permettant certainement, au prix de quelques optimisations, de conserver un traitement en temps réel dans ce cas.

Pour toutes ces raisons, une fréquence de 5 images/seconde a été choisie. La nature des transitions est alors conservée. En ce qui concerne leur localisation précise dans le temps, aucune étape supplémentaire n'est ajoutée dans le cadre de nos travaux. Notons cependant que, pour la fréquence choisie, l'imprécision n'est que de 5 images ; on peut donc imaginer qu'une étape de raffinement directe et triviale suffise (deuxième passe de calcul du critère sur ces 5 images, par exemple).

En conclusion, l'algorithme de détection des coupures que nous avons bâti repose sur quatre ou cinq paramètres suivant le choix de travailler à partir d'un masque binaire ou à niveaux de gris. Ces paramètres sont tous peu dépendants des données d'entrée, et pour tous une seule et même valeur a pu être choisie pour l'ensemble des séquences de test, avec des taux de détection que nous détaillons dans la section 3.2.8, mais dont on peut déjà dire qu'ils sont très élevés.

En outre, il est envisageable de s'affranchir du choix d'au moins un de ces paramètres, le seuil global, grâce à la mise en œuvre d'une sélection automatique de cette valeur de seuil, adaptée à la séquence d'entrée.

3.2.8 Performances - résultats

3.2.8.1 Description de la base de données

La base de données à notre disposition contient 22 séquences de provenance et de caractéristiques différentes résumées dans le tableau 3.1.

3.2.8.2 Résultats

L'algorithme développé a été testé sur chacune des séquences de la base, aboutissant aux résultats du tableau 3.2 fournis sous la forme de pourcentage de détection de transitions quelle que soit leur classe d'appartenance, pourcentage de détection de coupures et taux de fausses alarmes (détection de transition inexistante).

Pour chacun de ces taux, deux résultats brut et corrigé sont disponibles, ces deux termes, que nous explicitons ci-dessous, étant directement liés aux commentaires du tableau 3.3.

On fournit par ailleurs un exemple de représentation de la structure obtenue, sur la séquence test *travaux*, dans la figure 3.31, sous forme des images de départ et de fin de chacune des prises de vue extraites.

Le terme de *résultat brut* a été employé à dessein pour signifier que les pourcentages fournis sont souvent sous-estimés, i.e. pour quelques séquences, certaines fausses détections ou certains échecs de détection sont comptabilisés, alors qu'ils correspondent à des situations improbables dans des séquences réelles, et que l'algorithme ne peut de façon inhérente les éviter.

C'est le cas par exemple des plusieurs coupures successives ; on retrouve cette situation dans de nombreuses séquences issues du CCETT (nantes, toulouse, paris, senghor, acadie, abidjan). Ce cas est improbable dans des séquences réelles ; il signifie que des prises de vue ne contiennent qu'une image. Et au delà de deux coupures à suivre, l'algorithme ne peut les

Séquence	Provenance	Nb d'images	Fréquence (Hz)	Durée (s)	Tailles d'image
hard rock (seq1) - A2	CMM	180	6	30	384×288
broderie (seq2) - A2	CMM	180	6	30	384×288
kart (seq3) - TF1	CMM	181	6	30	384×288
tennis (seq4) - TF1	CMM	181	6	30	384×288
vieux tennis (seq5) - TF1	CMM	181	6	30	384×288
secte (seq6) - TF1	CMM	181	6	30	384×288
lille (seq7) - TF1	CMM	180	6	30	384×288
interview (seq8) - A2	CMM	181	6	30	384×288
6 minutes (seq9) - M6	CMM	181	6	30	384×288
jtv1 - FR3	CMM	1713	5	342.6	384×288
travaux - FR3	CCETT	250	5	50	360×288
affaire - FR3	CCETT	250	5	50	360×288
procès - FR3	CCETT	250	5	50	360×288
visite - FR3	CCETT	250	5	50	360×288
colère - FR3	CCETT	200	5	40	360×288
brèves - FR3	CCETT	200	5	40	360×288
abidjan	CCETT	1425	5	285	352×288
nantes	CCETT	66	5	13.2	352×288
paris	CCETT	426	5	85.2	352×288
acadie	CCETT	869	5	173.8	352×288
senghor	CCETT	356	5	71.2	352×288
toulouse	CCETT	36	5	7.2	352×288

TAB. 3.1: Base de données de séquences vidéo.

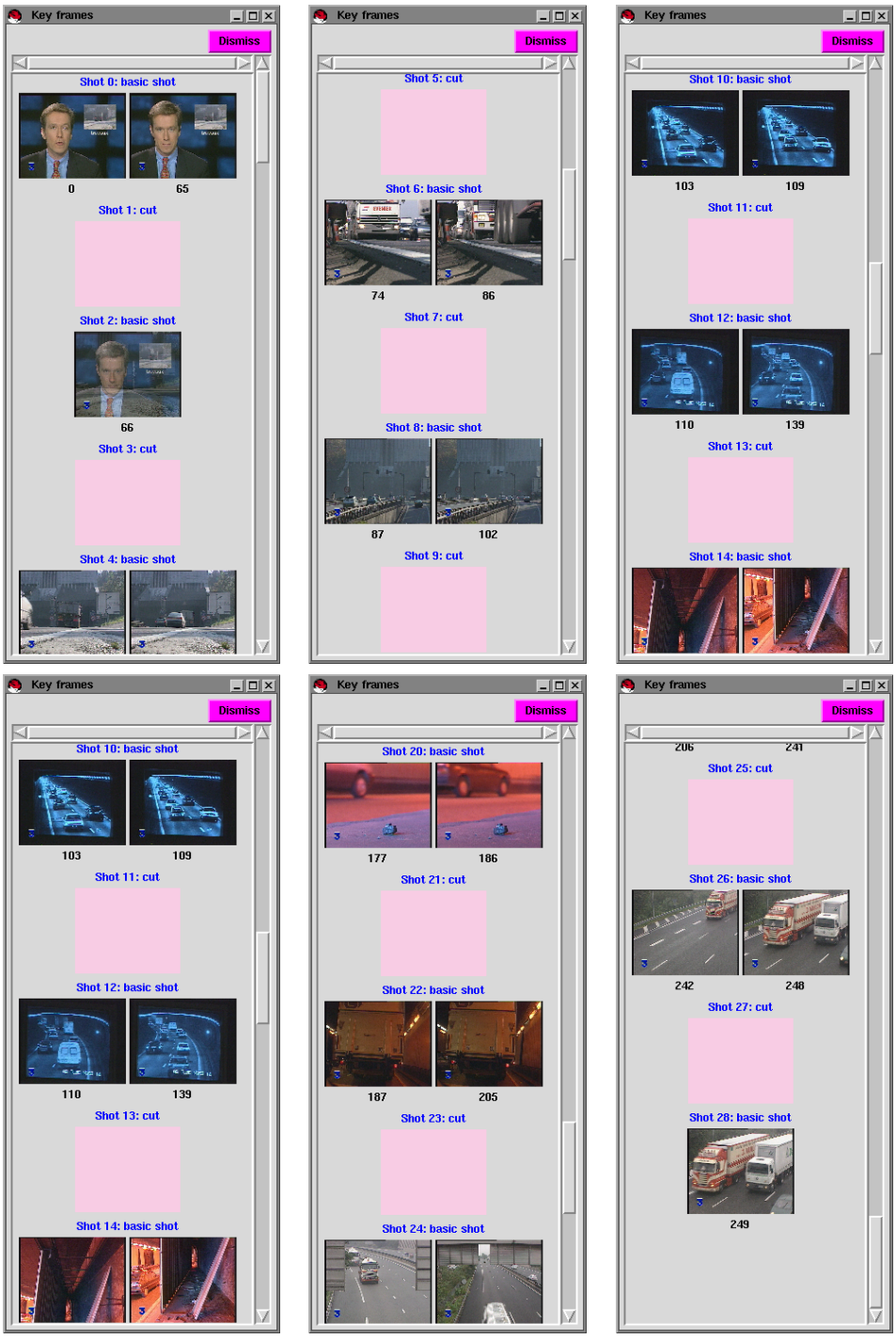


FIG. 3.31: Prises de vue extraites pour la séquence *travaux*.

détecter, du fait de l'utilisation de l'outil morphologique *chapeau haut de forme* de taille fixe et fine [35].

Séquence	% **coupures détectées**		% **total détectés**		% **fausses alarmes**	
	brut	**corrigé**	**brut**	**corrigé**	**brut**	**corrigé**
hard rock	100	100	87.5	87.5	0	0
broderie	100	100	83.3	83.3	0	0
kart	88.8	88.8	80	80	11.1	11.1
tennis	100	100	100	100	0	0
vieux tennis	100	100	100	100	20	0
secte	100	100	88.9	88.9	0	0
lille	100	100	66.6	66.6	0	0
interview	100	100	100	100	0	0
6 minutes	100	100	100	100	44.4	28.5
travaux	100	100	100	100	14.2	0
affaire	100	100	100	100	0	0
procès	100	100	100	100	33.3	0
visite	100	100	100	100	54.5	37.5
colère	100	100	100	100	0	0
brèves	100	100	100	100	0	0
abidjan	100	100	100	100	3.1	1.5
nantes	100	100	83.3	83.3	0	0
paris	80.9	100	73.1	88.5	5.5	0
acadie	96.8	100	83.8	86.5	0	0
senghor	100	100	61.5	61.5	0	0
toulouse	37.5	100	37.5	100	0	0
jtv1	100	100	100	100	9.0	2.4
total	**95.6**	**99.5**	**88.4**	**92.1**	**8.8**	**3.7**

TAB. 3.2: Résultats de la détection de transitions sur la base de données de séquences.

D'autre part, il arrive assez fréquemment que des fausses alarmes aient lieu entre l'avant-dernière et la dernière images de la séquence : il s'agit là d'un effet de bord de l'algorithme qui amplifie d'autant plus les différences entre images successives (dues à des mouvements principalement) que ces images sont les dernière et avant-dernière. C'est par exemple le cas de la séquence *vieux tennis*.

Outre ces effets de bord, les fausses alarmes sont dues pour beaucoup soit à des flashes détectés comme deux coupures successives (donc deux fausses détections par flash), soit à des fondus qui, lorsqu'ils sont détectés, sont souvent extraits comme des prises de vue à part entière. Dans ce dernier cas, nous les comptabilisons encore comme des erreurs de détection (résultats bruts), à raison d'une fausse alarme par fondu. Or, l'extraction d'un fondu comme une prise de vue propre se justifie pleinement dans un modèle où toutes les transitions sont considérées comme des prises de vue, y compris la coupure, prise de vue de longueur nulle. Cette modélisation permet en outre de stocker dans chaque prise de vue - transition, son image de début, son image de fin, des images clés éventuelles et une étiquette labélisant le type de transition. Tous ces champs ont déjà été définis pour la prise de vue classique (cf. section 2.4)

et leur réutilisation facilite grandement la manipulation de la structure de document vidéo.

Pour toutes ces raisons, les pourcentages sont volontairement plus bas que les résultats théoriques réels, fournis dans les colonnes "corrigées" du tableau 3.2.

L'utilisation de l'algorithme pour la détection de coupures simples fournit malgré tout des résultats tout à fait honorables, même en cas de mouvement rapide soit de la caméra, soit de gros objets en premier plan. Par contre, l'ensemble des autres transitions (fondus, balayages, etc.) ont des taux de détection beaucoup plus bas et sont d'ailleurs généralement responsables de la baisse des taux globaux. Pour cette raison, il apparaît indispensable d'étudier une autre solution de détection de ces transitions, et c'est l'objet de la section 3.3.

Face à ces résultats, une fois tous les cas "parasites" éliminés, les taux de détection atteignent des pourcentages élevés de 92.1%, et ceci pour un taux de fausses alarmes de 3.7%. Le pourcentage de détection des coupures seules atteint, quant à lui, 99.5%. Les quelques détections ratées sont alors souvent récupérables par le choix d'un seuil légérement plus bas, au détriment des quelques fausses alarmes supplémentaires. Ces fausses alarmes peuvent à leur tour être rapidement éliminées grâce à l'établissement d'un graphe de relations entre prises de vue, technique que nous développons au chapitre 8, section 8.2. Pour introduire dès à présent et en quelques mots cette nouvelle notion, remarquons que, si on est capable de déterminer que deux prises de vue adjacentes ont des contenus colorimétriques proches (i.e. sont en relation), on est en droit d'émettre un doute sur le bien-fondé de la transition intermédiaire.

L'établissement de ces relations entre prises de vue permettra également de s'affranchir des fausses détections dues à des flashes (événements courants dans les reportages de journaux télévisés) : les prises de vue de part et d'autre sont similaires, donc en relation. Mais cette extension de l'algorithme sera également détaillée par la suite (cf. section 8.2).

Enfin, en ce qui concerne les temps d'exécution de notre algorithme, l'objectif "temps réel" est atteint et même dépassé puique le temps d'exécution moyen est de 0.6 fois le temps réel sur un pentium II, 400MHz. Il convient de noter que ces temps, excessivement faibles, sont obtenus sans aucune optimisation. De tels résultats sont tout à fait compétitifs avec les méthodes existantes que nous avons citées dans la section 3.1.3, pour des taux de détection égaux, voire meilleurs. La simplicité de l'algorithme, en $O(n)$, autorise de plus la possibilité d'envisager une implantation hardware dédiée, capable de diminuer encore les temps d'exécution.

3.2.9 Conclusion - Améliorations

Afin de satisfaire les contraintes de rapidité (égaler au minimum le temps réel) et d'efficacité (atteindre au moins les taux de détection proposés dans la littérature) posées en introduction de ce chapitre, l'algorithme développé repose sur des techniques très simples de comparaison d'images, contrebalancées par des filtrages morphologiques efficaces, et tout particulièrement par un nouvel opérateur, le chapeau haut-de-forme inf. Cette combinaison aboutit à des taux de détection de coupures de plus de 99% pour des taux de fausses alarmes de 3.7% que l'on se propose de réduire par la suite et ceci pour un temps d'exécution moyen de 0.6 fois le temps réel, malgré une implantation non optimale. L'objectif fixé est donc atteint.

Le choix d'une détection locale s'est en outre avéré fournir des éléments de réponse pour une première classification des transitions géométriques et pour la détection de changements locaux, de type incrustations de texte, de bandeaux, etc.

Deux points restent cependant à mettre en œuvre : tout d'abord la construction de relations entre prises de vue, dont nous avons déjà évoqué plusieurs objectifs, parmi lesquels celui de détecter les fausses alarmes, et d'autre part l'élaboration d'un algorithme supplémentaire de

Séquence	Nb de transitions	Commentaires
hard rock	7 coupures - 1 fondu	1 fondu non détecté
broderie	5 coupures - 1 fondu	1 fondu non détecté
kart	9 coupures - 1 fondu	1 fondu et 1 coupure non détectés, 1 fausse détection due à un flash, séquence difficile
tennis	6 coupures	
vieux tennis	4 coupures	1 fausse alarme sur la dernière image (effet de bord)
secte	8 coupures - 1 fondu	1 fondu non détecté
lille	3 coupures - 3 balayages	2 balayages non détectés
interview	4 coupures	
6 minutes	5 coupures	2 fausses détections dues à des flashes
travaux	11 coupures - 1 fondu	1 fondu détecté comme séquence à part entière 1 fausse alarme sur la dernière image
affaire	2 coupures - 1 fondu	
procès	3 coupures - 1 fondu	1 fondu détecté comme séquence à part entière
visite	5 coupures	2 fausses détections dues à un flash 2 fausses alarmes dont un effet de bord
colère	14 coupures - 1 fondu	
brèves	13 coupures - 1 fondu	
abidjan	62 coupures	2 fausses alarmes dont un effet de bord
nantes	5 coupures - 1 fondu	1 fondu non détecté
paris	21 coupures - 3 fondus - 1 page tournée - 1 transition plus - sophistiquée	4 coupures, 2 fondus, 1 page tournée non détectés. 1 seule fausse alarme due à un fondu détecté comme séquence à part entière. Les coupures non détectées correspondent au cas extrême de coupures successives non traitées par l'algorithme Un seuil un peu plus bas (0.05) donne 1 fondu et 4 coupures non détectés, mais plus de fausses alarmes : 2 fondus détectés comme séquence à part entière et objet en mouvement et en gros plan (2)
acadie	32 coupures - 5 fondus	5 fondus et 1 coupure non détectés. La coupure se trouve dans la situation improbable de 2 coupures successives.
senghor	8 coupures - 5 fondus	5 fondus non détectés
toulouse	8 coupures	4 coupures non détectées (successives)
jtv1	40 coupures	Il s'agit d'un document créé non réel. Certaines prises de vue commencent par des bouts de fondus 3 des 4 fausses alarmes sont dues à des fondus pris comme séquence à part entière.

TAB. 3.3: Commentaires sur la détection de transitions sur la base de données de séquences, après correction de quelques problèmes.

détection des transitions de type fondus (groupes 3 et 4), les transitions à modèle géométrique étant partiellement traitées (cf. section 3.2.4.3).

La détection et la suppression des fausses alarmes, quant à elles, feront l'objet de la section 8.2 du chapitre *Applications*. Nous proposons à présent un deuxième algorithme, en essence proche de celui que nous venons de détailler, mais adapté à la détection des transitions de type fondus.

3.3 Détection des fondus

La deuxième classe de transitions la plus utilisée après les coupures est sans aucun doute celle des fondus et fondus enchaînés, dont une description est disponible à la section 3.1.2.2.

L'inconvénient majeur de l'algorithme de détection de transitions que nous venons d'exposer consiste en la non-détection des fondus. Nous proposons donc ici un second algorithme, tout spécialement dédié à la détection de cette classe de transitions, correspondant au groupe 4 dans notre classification (cf. section 3.1.2).

Mais avant d'expliciter et de bâtir ce nouvel algorithme, commençons tout d'abord par une courte analyse des raisons de la non-détection des fondus dans l'algorithme de détection des coupures.

3.3.1 Analyse de l'algorithme de détection de coupures

Rappelons encore une fois que les coupures sont des transitions brutales de durée nulle (pas d'état de transition) et affectant toute l'image en même temps. Même dans le cas où les deux prises de vue successives sont relativement similaires, la coupure apparaît sous la forme d'un pic, plus ou moins haut, dans la courbe d'évolution du critère de dissemblance choisi.

Les fondus ont des caractéristiques opposées : ils possèdent une durée propre ; tous les pixels passent par un même état de transition pendant lequel leur niveau de gris est modifié, et la fonction de modification est progressive, de la forme de l'équation 3.4.

Du fait de cette modification progressive du niveau de gris des pixels, les valeurs du critère de dissemblance sont faibles sur la durée du fondu. On n'observe donc pas d'effet de pic comme pour les coupures dans la majorité des cas. Dans les situations, relativement rares, où un pic apparaît quand même, sa hauteur est beaucoup plus faible que celles des pics correspondant à des coupures. D'autre part, ce pic résiduel est localisé entre deux images à l'intérieur du fondu, mais, en aucun cas, ne donne accès à la totalité des informations du fondu, à savoir :

- début de la transition, $t_0 + 1$;
- fin de la transition, t_1 ;
- durée = fin - début + 1 = $t_1 - t_0$.

De par l'absence de modèle géométrique (tous les pixels sont affectés en même temps et pendant la durée du fondu), il est également impossible de se baser sur une étude de la géométrie du masque de transition pour classifier des transitions de ce type.

Une solution conduisant à la détection des fondus toujours à l'aide de ce premier algorithme, consisterait à adopter une technique multi-échelles temporelles, grâce au choix d'une fréquence de comparaison des images plus basse (cf. section 3.2.7). De cette façon, toute transition, quelle que soit sa nature, finit par apparaître, pour une valeur suffisamment basse de fréquence, comme une coupure, avec l'inconvénient de devoir revenir par la suite sur chaque

coupure détectée, de façon à déterminer la nature exacte de la transition ayant eu lieu et sa durée.

Ainsi que cela a déjà été discuté dans la section 3.2.7, le choix de fréquences plus basses complique l'algorithme, et dans le cas des fondus, ne donne toujours pas de solution pour détecter l'endroit exact de la transition, ni sa nature.

Suite à ces observations, la solution adoptée consiste à sélectionner un nouveau critère (section 3.3.3), basé sur une étude théorique des fondus (section 3.3.2), pour lequel cette classe de transitions apparaît sous une forme géométrique propre sur la courbe d'évolution temporelle. De la même façon que les pics représentant les coupures sont extraits par chapeau haut de forme inf, la forme géométrique correspondant aux fondus sera par la suite extraite par l'utilisation d'outils morphologiques (section 3.3.4).

3.3.2 Etude théorique des transitions

Nous proposons ici une étude théorique de transitions de type fondu, en comparaison avec deux autres situations : l'absence de transition et la coupure.

Nous reprenons les notations introduites dans la section 3.1.2, avec les ajouts suivants.

On note (F) un fichier vidéo et F_t l'image à l'instant t de ce fichier vidéo. La quantité ΔF_{t+1} symbolise alors la différence entre deux images successives t et $(t+1)$ du fichier vidéo :

$$\Delta F_{t+1} \quad = \quad F_{t+1} - F_t \tag{3.20}$$

ΔF_{t+1} n'est autre qu'une approximation classique de la dérivée temporelle d'ordre 1 de F : $\frac{\partial F}{\partial t}$.

De même, on note $\Delta^2 F_{t+1}$ la différence d'ordre 2 (approximation discrète de la dérivée d'ordre 2, $\frac{\partial^2 F}{\partial t^2}$) entre deux images successives :

$$\Delta^2 F_{t+1} = \Delta F_{t+1} - \Delta F_t \tag{3.21}$$

Cette différence est elle-aussi porteuse d'information, lors d'une transition entre deux prises de vue.

On retrouve ici, par le calcul des deux quantités ΔF et $\Delta^2 F$, un problème très similaire à celui de la détection de contours [43], pour lequel le calcul des dérivées discrètes d'ordres 1 et 2 sont à la base de nombreux algorithmes [92, 21, 39, 68].

Nous avons déjà souligné qu'à chaque transition, il était bien souvent possible de relier une géométrie et un modèle mathématique de modification des niveaux de gris. En pratique, du fait du bruit et du mouvement présent dans les deux prises de vue avant et après transition, la géométrie et le modèle mathématique ne sont pas parfaitement vérifiés. Le filtrage spatial des masques de transition permet alors de se rapprocher du modèle géométrique théorique.

En ce qui concerne le modèle mathématique de modification des niveaux de gris, même imparfait, il mérite quand même que l'on s'y arrête. L'étude de la différence ΔF permet alors d'en avoir une première approximation. Nous calculons ainsi cette quantité dans le cas général d'une seule et même prise de vue, puis pour une coupure et un fondu, lorsque les deux prises de vue de part et d'autre sont fixes, et dans le cas plus réel où ces deux prises de vue contiennent du bruit et/ou du mouvement.

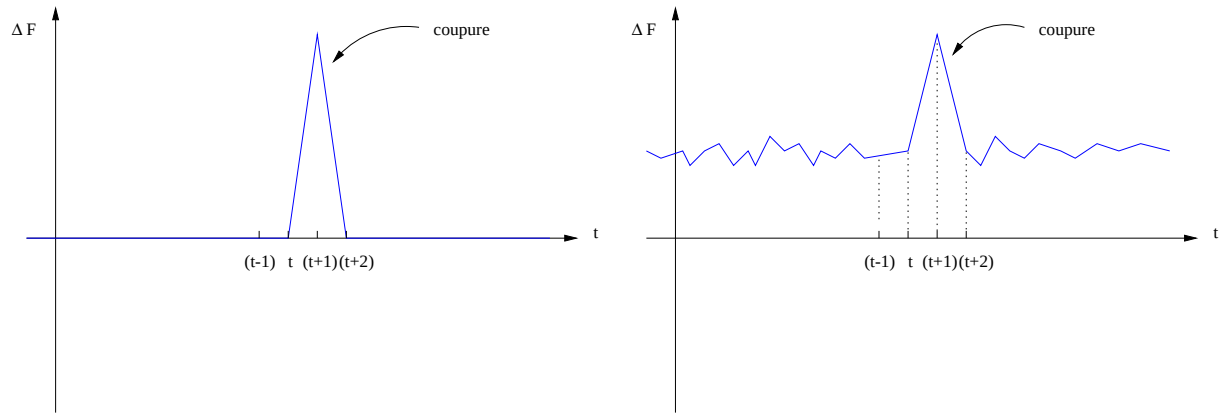
Absence de transition ou de mouvement de caméra. A l'intérieur d'une même prise de vue, les deux quantités ΔF et $\Delta^2 F$ sont toutes les deux nulles, en absence de bruit, de mouvement d'objets ou de la caméra et de changement d'éclairage :

$$\Delta F_{t+1} = 0 \quad (3.22)$$
$$\Delta^2 F_{t+1} = 0 \quad (3.23)$$

Cas d'une coupure. L'expression d'une coupure à l'instant $(t+1)$ entre les prises de vue (I) et (J) s'écrit trivialement :

$$\Delta F_{t+1} = J_{t+1} - I_t \quad (3.24)$$

Dans le cas où les prises de vue I et J sont fixes et sans bruit, on obtient une courbe d'évolution de ΔF très simple (cf. figure 3.32, (a)). Si les prises de vue contiennent du mouvement, l'allure globale de la courbe reste la même, avec un niveau de bruit plus élevé en dehors de l'instant de transition (cf. figure 3.32, (b)). Dans les deux cas, la transition se traduit par un pic étroit sur la courbe.



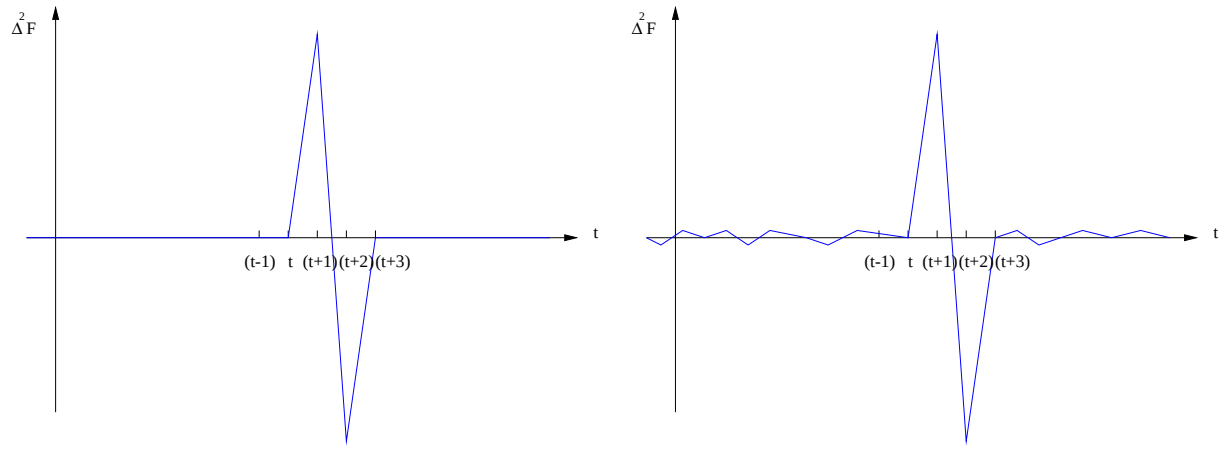
(a) (b)

FIG. 3.32: Courbes d'évolution de ΔF dans le cas d'une coupure entre deux prises de vue (a) fixes ; (b) avec mouvement.

On obtient également une courbe d'évolution de la différence d'ordre 2 telle que celle de la figure 3.33, (a), pour deux prises de vue fixes, ou celle de la figure 3.33, (b), pour deux prises de vue contenant du mouvement, la transition se traduisant par un passage par zéro de la dérivée d'ordre 2.

Ces courbes ΔF et $\Delta^2 F$ sont caractéristiques d'une coupure entre deux prises de vue.

Cas d'un fondu. L'équation 3.4 donne l'expression d'un fondu comme la somme pondérée à chaque instant des deux images correspondantes dans chacune des prises de vue.



(a) (b)

FIG. 3.33: Courbes d'évolution de $\Delta^2 F$ dans le cas d'une coupure entre deux prises de vue (a) fixes ; (b) avec mouvement.

La différence d'ordre 1 devient alors :

$$\begin{aligned}
\Delta F_{t+1} &= F_{t+1} - F_t \\
\Delta F_{t+1} &= \left(1 - \frac{t+1-t_0}{t_1-t_0}\right) I_{t+1} + \frac{t+1-t_0}{t_1-t_0} J_{t+1} - \left(1 - \frac{t-t_0}{t_1-t_0}\right) I_t - \frac{t-t_0}{t_1-t_0} J_t \\
\Delta F_{t+1} &= \frac{1}{t_1-t_0}(J_{t+1} - I_t) + \frac{t-t_0}{t_1-t_0}(J_{t+1} - J_t) + \left(1 - \frac{t+1-t_0}{t_1-t_0}\right)(I_{t+1} - I_t)
\end{aligned}$$

Soit, en utilisant les notations $\Delta J = J_{t+1} - J_t$ et $\Delta I = I_{t+1} - I_t$:

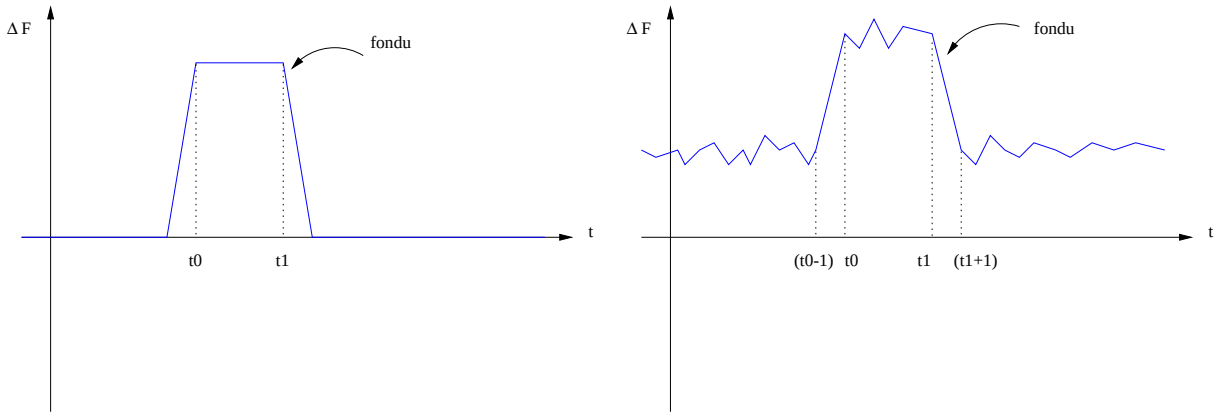
$$\Delta F_{t+1} = \frac{1}{t_1-t_0}(J_{t+1} - I_t) + \frac{t-t_0}{t_1-t_0}\Delta J + \left(1 - \frac{t+1-t_0}{t_1-t_0}\right)\Delta I \tag{3.25}$$

Dans le cas de deux prises de vue fixes, ΔI et ΔJ sont nulles ; l'expression se simplifie en une valeur constante pendant toute la durée du fondu :

$$\Delta F_{t+1} = \frac{1}{t_1-t_0}(J_{t+1} - I_t) \tag{3.26}$$

On obtient ainsi la courbe 3.34, (a), pour un fondu entre deux prises de vue fixes, et la courbe 3.34, (b), si les deux prises de vue contiennent du mouvement. La constance de ΔF pendant le fondu, se traduit à l'ordre 2 par une valeur nulle de la courbe sur la durée du fondu, séparée par deux pics, l'un positif et l'autre négatif (cf. figure 3.35), toujours dans le cas de deux prises de vue fixes.

Un fondu entre deux prises de vue fixes se définit donc par une valeur de ΔF constante et élevée et par un passage par zéro avec changement de signe de la quantité $\Delta^2 F$.



(a) (b)

FIG. 3.34: Courbes d'évolution de ΔF dans le cas d'un fondu entre deux prises de vue (a) fixes ; (b) avec mouvement.

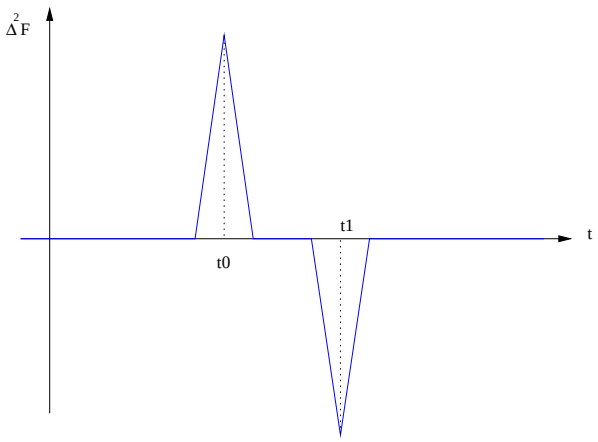


FIG. 3.35: Courbe d'évolution de $\Delta^2 F$ dans le cas d'un fondu entre deux prises de vue fixes.

Dans le cas de deux prises de vue quelconques, donc a priori avec mouvement, cette remarque doit bien sûr être modulée. En fait, on considère qu'elle est tout de même vérifiée globalement sur toute l'image dans la mesure où, dans le cas d'un mouvement peu important, le changement qui en découle ne se traduit qu'aux frontières des objets. A l'intérieur de ces mêmes objets, la prise de vue peut être considérée fixe.
On cherche donc à maximiser dans les images le nombre de points vérifiant $\Delta F \neq 0$ (il y a un changement en ce point) et $\Delta^2 F = 0$ (i.e. ΔF est constant).

En pratique l'extraction des courbes d'ordre 2, des pics entourant une zone où $\Delta^2 F = 0$, est délicate : les pics sont généralement de hauteur assez faible et ne sont pas toujours présents (cf. figure 3.36, pour un exemple réel parfait, de faible niveau de bruit). Cette constatation est encore une fois bien connue dans le domaine de l'extraction de contours : le calcul des dérivées d'ordres 1 et 2 d'une fonction déjà bruitée a tendance à augmenter encore l'effet du bruit [43], d'où la conclusion, à laquelle aboutissent beaucoup d'auteurs de ce domaine, de la nécessité d'une étape de pré-filtrage. Dans notre cas, loin de chercher à améliorer le calcul de la dérivée d'ordre 2, nous avons préféré nous restreindre à la dérivée d'ordre 1, i.e. aux courbes d'évolution d'ordre 1, desquelles, par filtrage morphologique, il est possible d'extraire les dômes caractéristiques des fondus.

Il est cependant intéressant à ce stade de remarquer la grande similitude des deux problèmes de détection des transitions et d'extraction de contours. Une telle remarque mène à une nouvelle interprétation du problème de détection des changements de scène, qui peut être réinterprété comme une segmentation en régions (les différentes prises de vue) par extraction de leurs contours (les transitions). Les courbes d'évolution temporelles extraites ne sont alors pas autres que les pendants du gradient et du laplacien spatiaux.

Cette similitude étant établie, nous détaillons à présent le critère de dissemblance choisi, i.e. l'expression de ΔF, pour construire les courbes d'évolution présentant ces caractéristiques.

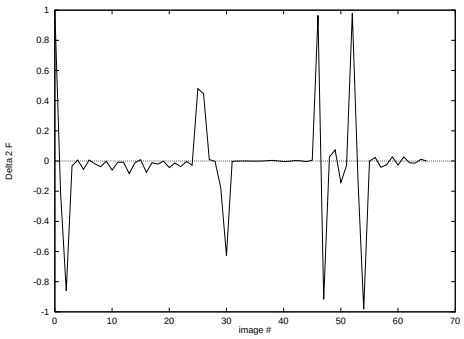


FIG. 3.36: Courbe d'évolution de $\Delta^2 F$ obtenue pour la séquence réelle *nantes*. Cette séquence présente un fondu dans l'intervalle $[26, 29]$ et quatre coupures aux instants 1, 47, 53 et 54. Cet exemple peu bruité est une exception, dans la mesure où les passages par zéro de $\Delta^2 F$ sont aisés à extraire, sauf toutefois pour la coupure 53, pour laquelle les pics sont d'un niveau nettement moins élevé.

3.3.3 Choix d'un nouveau critère

Toute la théorie mathématique précédente sous-entend le calcul d'une différence d'image à image, que nous avons notée ΔF, mais à aucun moment la façon dont cette différence était

concrètement calculée, n'a été explicitée. Or c'est à partir du choix primordial de ce nouveau critère de dissemblance entre images que les caractéristiques géométriques des fondus et autres événements sont susceptibles d'apparaître sur les courbes d'évolution ci-dessus.

Ce choix découle tout naturellement des observations concrètes des modifications dues aux transitions dans les images. Dans la section précédente, on a formulé l'hypothèse, idéalement vérifiée dans le cas de scènes fixes, qu'à l'intérieur d'une même prise de vue, les différences pixel à pixel sont nulles. Dans le cas d'une prise de vue contenant du bruit et du mouvement, les seules valeurs de différence non nulles vont donc correspondre à des points bruités ou bien situés sur les frontières d'objets en mouvement. En effet, si on suppose que les objets contenus dans l'image sont majoritairement d'une taille assez grande (plus de quelques pixels) et ont une texture relativement uniforme, seuls les bords de ces objets sont responsables de différences d'une image à l'autre.

La continuité d'une prise de vue se traduit donc par une constance de la majorité des pixels de l'image, hormis ceux situés aux bords des objets ou bruités. Une coupure, au contraire, correspond à une modification de chaque pixel ; le fondu modifie également sur toute sa durée chaque pixel de l'image.

On retrouve ainsi, pour la détection de fondus comme pour l'algorithme de détection de coupures, la nécessité de conserver une information locale de changement, de façon à conserver la différence de comportement des pixels à l'intérieur de régions homogènes de celui des pixels de bords. Mais cette fois-ci le caractère local est poussé à l'extrême puisqu'on calcule une distance point à point sans cette fois-ci faire suivre ce calcul d'une moyenne sur de petits blocs (cf. critère local, section 3.2.2) ou sur l'image totale (cf. critère global, section 3.2.3). Le critère de différence choisi est toujours une distance dans l'espace de couleur RGB, calculé point à point, nous permettant ainsi d'obtenir une image à niveaux de gris, un niveau de gris (i.e. une valeur de critère) par pixel.

En conséquence, le nombre de pixels ayant une différence non nulle entre deux instants doit être proche de zéro à l'intérieur d'une même prise de vue, égal au nombre de pixels de l'image pour une coupure (tous les pixels sont fortement modifiés) et d'un niveau constant non nul et non maximum pendant la durée d'un fondu.

On effectue ainsi un seuillage de l'image de distance pixel à pixel de façon à ne conserver que les pixels ayant une valeur de différence non nulle (en pratique supérieure à 5). La courbe finale de critère de dissemblance correspond donc à la proportion de pixels sur toute l'image satisfaisant à cette dernière contrainte au cours du temps. Cette première partie de l'algorithme est illustrée sous forme de schéma dans la figure 3.37.

Par le choix de ce critère de distance couleur pixel à pixel seuillée, la première contrainte caractéristique des fondus, $\Delta F \neq 0$, est vérifiée. Les courbes obtenues dont on présente un échantillon dans la figure 3.38 possèdent également les dômes caractéristiques de présence de fondus, bien que le critère représenté soit le nombre de pixels tels que $\Delta F \neq 0$, et non ΔF elle-même.

La deuxième contrainte, ΔF constante, i.e. $\Delta^2 F = 0$, sera transposée en une nouvelle contrainte (constance du nombre de pixels tels que $\Delta F \neq 0$), imposée lors de l'extraction des dômes, extraction qui ne sera optimale, comme nous le verrons dans la section suivante, que dans le cas de dômes relativement constants.

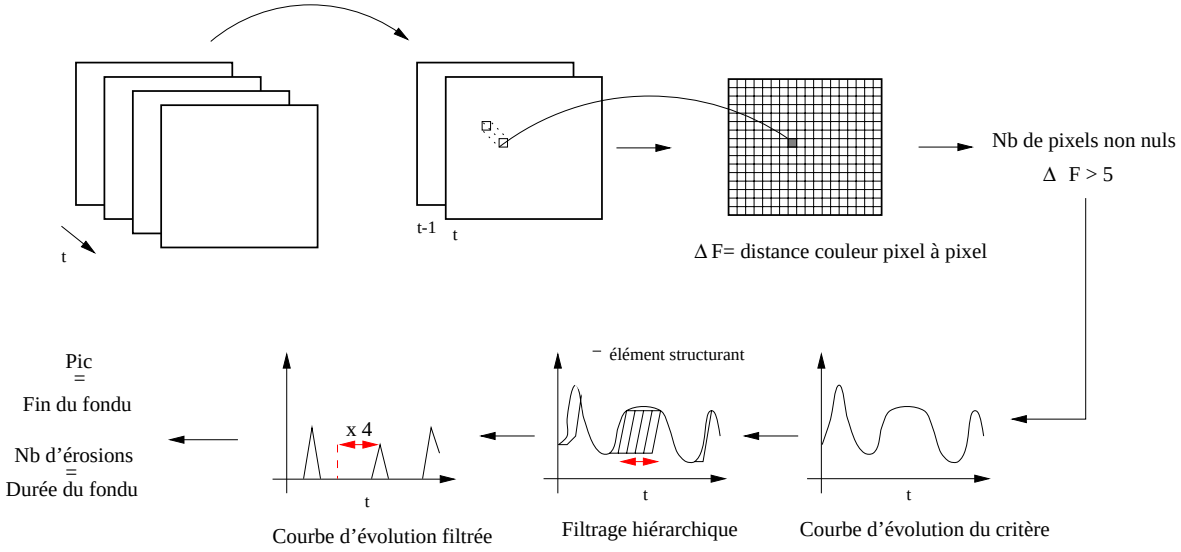


FIG. 3.37: Algorithme de détection des fondus dans un document vidéo. Cet algorithme se décompose en deux parties : le calcul d'un critère de dissemblance entre images successives débouchant sur une courbe d'évolution temporelle (partie supérieure du schéma) et le filtrage de cette courbe (partie inférieure) de façon à extraire les fondus et leurs caractéristiques (images de début et de fin, durée).

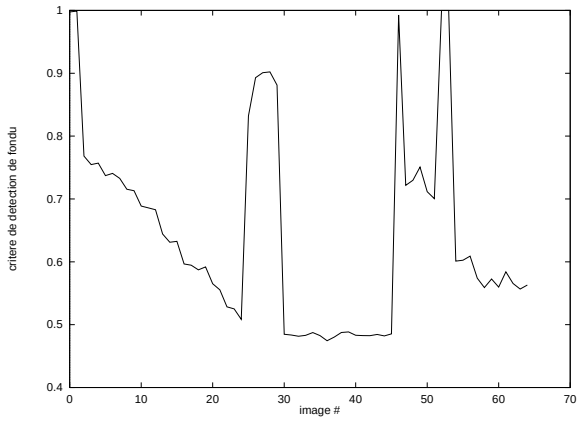


FIG. 3.38: Exemple de courbe d'évolution du critère de détection des fondus pour la séquence *Nantes*. Le fondu est situé entre les images 26 et 29.

3.3.4 Extension par hiérarchie de l'algorithme

Une fois la courbe d'évolution du critère obtenue, il s'agit d'en extraire les dômes signifiant la présence de fondus. Dans la continuité de ce qui a été réalisé pour la détection de coupures, cette extraction est à nouveau basée sur des filtrages morphologiques, mais avec la différence essentielle qu'au lieu d'un seul filtrage, plusieurs seront appliqués hiérarchiquement sur la courbe.

Nous avons émis l'idée, dans la section 3.1.1, d'extraire les fondus à partir du premier algorithme de détection des coupures grâce à l'utilisation d'une fréquence de comparaison des images plus basse, permettant la transformation de transitions longues et progressives telles que les fondus en transitions de type coupures. Cette piste a été abandonnée du fait de la nécessité par la suite de revenir sur chaque transition pour en déterminer sa nature exacte. Cette solution cependant a le mérite de mettre en avant le caractère différent des transitions aux différentes échelles de fréquence. C'est cette caractéristique particulière que nous désirons retrouver dans la courbe obtenue à la section précédente.

L'idée consiste donc en la transformation progressive, par des opérateurs morphologiques, de type érosion, des dômes de fondus en pics, qui sont alors extraits à nouveau par chapeau haut-de-forme.

Pour un choix d'élément structurant linéaire de longueur deux pixels et d'origine placée à l'extrémité droite, on supprime à chaque érosion une bande de la longueur de l'élément structurant sur la gauche du dôme. Ce processus, qui correspond à la notion d'érodé ultime temporel en morphologie mathématique, est illustré dans la figure 3.39, à partir de la courbe originale de la figure 3.38.

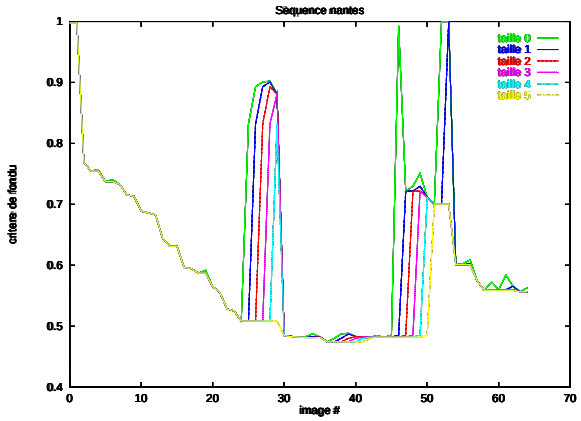


FIG. 3.39: Erosions successives appliquées à la courbe d'évolution du critère de détection de fondu, pour la séquence *Nantes*. L'érosion de taille 0 correspond à la courbe originale.

La notion de facteur d'échelle se retrouve donc dans le nombre d'érosions nécessaires à la transformation d'un dôme donné en un pic. Ce nombre d'érosions, propre à chaque fondu, est alors directement lié à sa durée. L'instant d'apparition du pic, extrait par chapeau haut-de-forme, correspond à l'instant, t_1, de fin du fondu. Il est alors trivial de déterminer la première image du fondu, située à $t_1 -$ durée, c'est-à-dire à $t_1 -$ nb. érosions. Cette deuxième partie de l'algorithme donne donc accès à toutes les caractéristiques des fondus extraits.

Une contrainte minime supplémentaire ajoutée à la décision de présence d'un fondu est à noter : seuls les dômes d'au maximum 3 secondes sont étiquetés "fondus". Au delà de cette

durée, une autre terminologie est utilisée qui sera détaillée par la suite dans le chapitre 4, section 4.2.2. Cette limite supérieure est légèrement au-delà de la durée moyenne d'un fondu qui dure rarement plus de 2 secondes.

Chaque dôme extrait correspond à ce stade à une structure de prise de vue, telle que nous l'avons introduite dans la section 2.4 et étiquetée en fonction de sa longueur, "fondu" ou "autre". Il s'agit ensuite d'insérer cette nouvelle prise de vue dans la structure de document vidéo. Pour avoir lieu, cette insertion doit répondre à certaines règles logiques que nous détaillons dans la section suivante.

3.3.5 Insertion des fondus dans la structure de document vidéo

Du fait de l'application hiérarchique des érosions sur la courbe d'évolution du critère, la détection d'un fondu de taille n en nombre d'images n'est réalisée qu'avec un retard de n images. En d'autres termes, un fondu de 5 images commençant en t et finissant en $t+5$ ne sera extrait qu'en $t+10$. Entre temps, tout fondu ou autre transition plus petite aura déjà été détectée et insérée dans la structure de document vidéo.

Pour cette raison, l'insertion des fondus ne s'effectue pas de façon linéaire comme dans l'algorithme de détection des coupures. L'extraction de ces dernières est en effet réalisée lors d'un parcours linéaire de la courbe d'évolution, à l'instant même où elles ont lieu. Chaque coupure est alors directement insérée à la fin de la structure déjà bâtie, sans qu'il soit besoin de revenir en arrière sur les coupures précédentes.

L'insertion doit cette fois-ci pouvoir avoir lieu en milieu de structure et tenir compte des diverses prises de vue et transitions déjà insérées.

Pour cela des règles logiques simples ont été écrites, constituant une sorte de grammaire d'insertion. Chaque nouveau fondu potentiel n'est alors inséré que si ces règles sont validées. A l'instant d'insertion du fondu, la structure de document vidéo peut déjà comporter des prises de vue étiquetées **normale**, **coupure**, **fondu**, etc. (cf. section 2.4).

Les règles d'insertion logiques sont donc les suivantes :

- Si le fondu potentiel se positionne directement avant ou après un fondu déjà existant, aucune insertion n'a lieu. Il est en effet irréaliste de rencontrer deux fondus successifs dans un fichier vidéo. Le premier fondu inséré, de durée moindre, prime sur le nouveau fondu. Cette règle permet de renforcer la détection de fondus courts et au contraire d'invalider celle de fondus plus longs et donc susceptibles d'être confondus avec des événements correspondant à des dômes plus larges.
- Aucun fondu n'est inséré s'il se positionne à cheval sur deux (ou plus) prises de vue, i.e. s'il n'est pas entièrement contenu dans une seule et même prise de vue. Cette règle joue le même rôle que la précédente de limitation de l'insertion de dômes plus larges, mais cette fois-ci correspondant à des artéfacts de l'algorithme de détection, qui tend parfois à extraire deux ou plusieurs prises de vue de niveau de bruit plus élevé que leurs voisines, sous un même dôme, alors très large.
- L'insertion du fondu potentiel directement avant ou après une coupure est autorisée, mais entraîne la suppression de la coupure. Ceci revient à remplacer une transition de type coupure par une transition de type fondu. Certains bords montants ou descendants des dômes sont en effet détectés comme coupures, fausses alarmes auxquelles on remédie donc par cette règle. Ceci étant, d'autres fausses alarmes sont alors introduites dans le cas particulier d'un fondu potentiel s'insérant entre deux coupures réelles. Le dôme

extrait ne correspond pas dans ce cas à un fondu mais à une courte prise de vue d'un niveau moyen de critère plus élevée que ses voisines (cf. figure 3.43).

– A condition de respecter les règles énoncées ci-dessus, il est possible de remplacer entièrement une prise de vue de type **normale** par une prise de vue **fondu** si elles sont de même longueur.

Notons qu'il est bien sûr possible, et à très peu de frais, de rajouter toute règle supplémentaire d'insertion. Cette grammaire de base étant énoncée, sa validation constitue la troisième phase de l'algorithme, dont nous donnons à présent les performances.

3.3.6 Performances - Résultats

Notre base de données test dont les caractéristiques sont résumées dans le tableau 3.1 contient 22 transitions de type fondu enchaîné et une transition plus complexe du groupe 3, selon la classification énoncée en 3.1.2. Une image intermédiaire de cette transition agissant sur les niveaux de gris et possédant son propre modèle géométrique est fournie dans la figure 3.40.

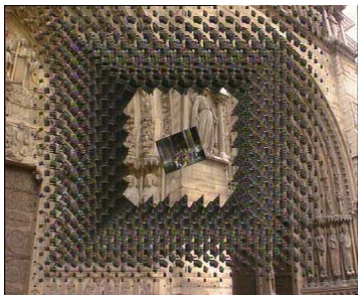
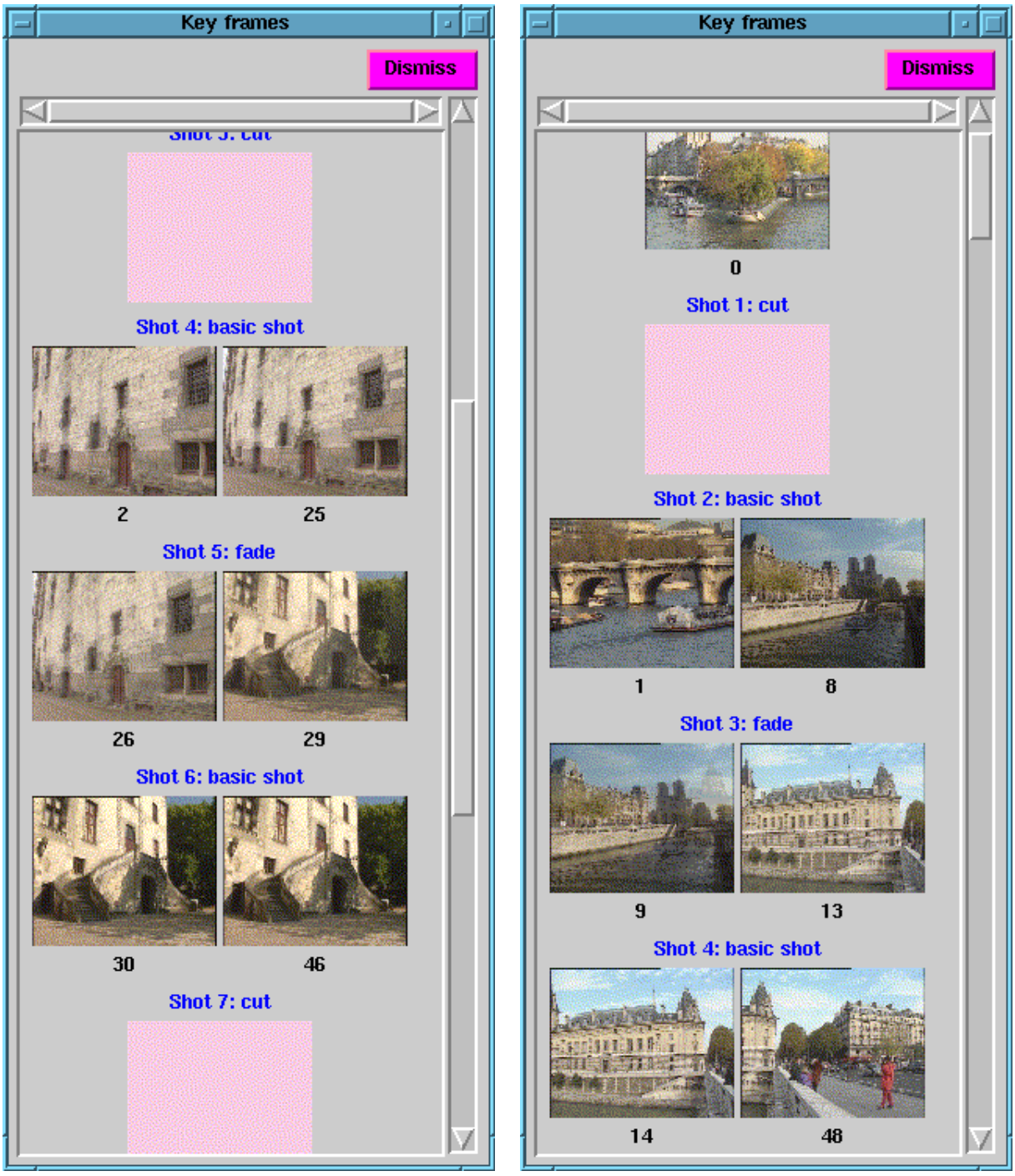


FIG. 3.40: Image intermédaire d'une transition de type 3.

Après lancement de l'algorithme sur les 22 séquences disponibles, nous aboutissons à un taux de détection de 78.3%, soit 18 sur 23 transitions. Le taux de fausses alarmes est élevé puisqu'il atteint 65.4%. En ce qui concerne le temps d'exécution, l'algorithme entier fournit la structure finale de document vidéo en 0.7 fois le temps réel, sans aucune optimisation de code, sur un pentium II, 400 MHz. Nul doute encore une fois qu'il soit possible de baisser ces temps de calcul du fait de la grande simplicité des traitements effectués sur les images. Le calcul du critère est en effet encore une fois en $O(n)$, avec n le nombre d'images de la séquence. Quant à la partie filtrage hiérarchique, elle est en $O(n \times n_e)$, avec n_e le nombre d'érosions effectuées. Nous fournissons des exemples de fondus détectés dans les figures 3.41 et 3.42, sous la forme des images de début et de fin des prises de vue avant, pendant et après transition. De par ces exemples, il apparaît que notre algorithme est capable de détecter les transitions de type fondu quelle que soit leur durée (de une à une petite dizaine d'images à la fréquence de 5Hz).

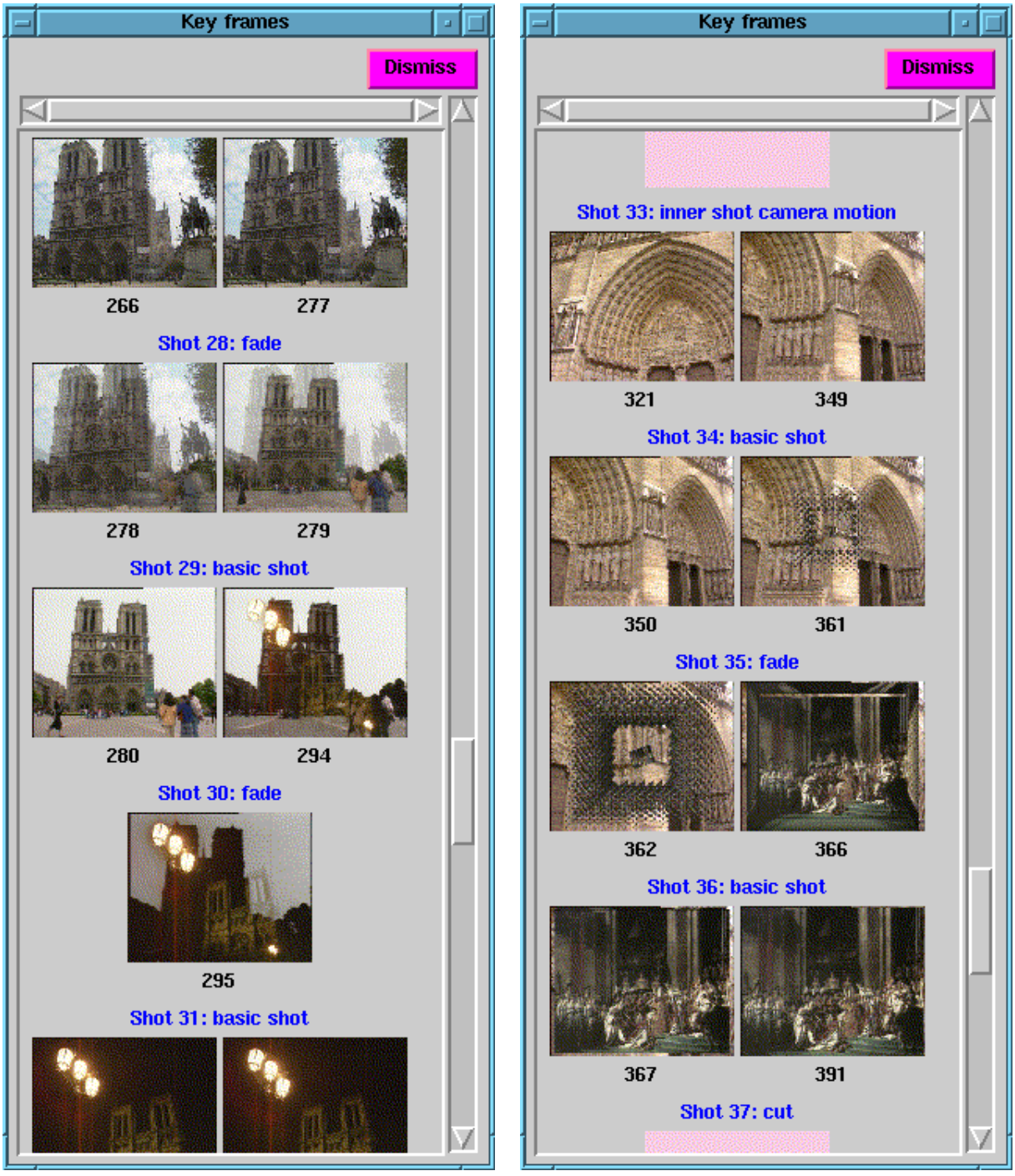
En outre, d'autres transitions chromatiques, i.e. agissant sur les niveaux de gris des pixels, sont détectées, comme le prouve l'extraction de la transition illustrée dans la figure 3.40. De même que l'algorithme de détection des coupures avait été étendu à l'ensemble des transitions du groupe 2 (possédant un modèle géométrique mais pas d'état de transition), cet algorithme permet également l'extraction de certaines transitions plus complexes, possédant un modèle géométrique et un état de transition.

L'extraction de dômes ou d'autres formes spécifiques à telle ou telle classe de transitions permet une localisation des segments du fichier vidéo à étudier plus précisément, par exemple



(a) (b)

FIG. 3.41: Exemples de fondus détectés : (a) séquence *Nantes*, prise de vue 5 ; (b) séquence *Paris*, prise de vue 3.



(a) (b)

FIG. 3.42: Exemples de fondus détectés : (a) séquence *Paris*, prises de vue 28 et 30 ; (b) séquence *Paris*, prise de vue 35 ; il s'agit ici d'une transition plus sophistiquée qu'un simple fondu, mais correctement détectée par l'algorithme.

grâce à la géométrie du masque de transition (cf. section 3.2.4.3). L'extraction des dômes donne en effet accès aux caractéristiques requises pour une étude du modèle géométrique, que sont les instants de début et de fin de transition.

Deux des fondus non détectés par l'algorithme appartiennent en outre à des séquences fort bruitées et de mauvaise qualité (séquences *broderie* et *kart*), pour lesquelles le niveau moyen du critère est extrêmement élevé, sur toute la durée de la séquence.

Le taux de fausses alarmes élevé (65.4%) pénalise les bons taux de détection de notre algorithme. Notons cependant que le calcul de ce taux T_f, tout particulièrement pour une séquence contenant peu de fondus, conduit naturellement à des taux élevés :

$$T_f = \frac{\text{nb. fausses alarmes}}{\text{nb. de fondus détectés} + \text{nb. fausses alarmes}} \tag{3.27}$$

Ces fausses alarmes sont dues à plusieurs situations différentes mais qui toutes correspondent à une structure sous-jacente de dôme dans les courbes d'évolution, ce qui tend à prouver que, loin de remettre en cause la partie extraction des dômes, qui est correcte, il s'agit de rajouter de nouvelles règles supplémentaires d'insertion, de façon à éliminer ces faux positifs.

Ainsi un cas fréquent de fausse alarme se produit lorsqu'une prise de vue entourée des deux transitions de type coupures possède des valeurs de critère plus hautes que les prises de vue voisines. On donne une illustration schématique de cette situation dans la figure 3.43. La courbe possède bien une structure de dôme caractéristique mais surmontée de deux pics parasites, représentant les deux coupures.

FIG. 3.43: Situation erronée de détection d'un dôme de fondu.

Tout comme pour les fausses alarmes issues de la détection de coupures, il est facilement envisageable d'utiliser la structure relationnelle entre prises de vue, que nous établirons dans le chapitre 7, pour valider la détection de fondus effectués et supprimer les cas erronés d'insertion de fondus. Nous ne détaillerons pas ici la technique utilisée alors, puisque c'est l'objet de la section 8.2, mais soulignons d'ores et déjà que la baisse du taux de fausses alarmes obtenue, réellement significative, fait de l'algorithme de détection de fondu que nous venons de présenter un outil puissant et rapide, d'un degré de robustesse équivalent à celui des méthodes existantes.

Nous terminons ce paragraphe par une dernière remarque sur le fait que seul un faible pourcentage de coupures sont extraites par l'algorithme de détection de fondus. Les deux

algorithmes de détection de transitions présentés jusqu'à présent sont donc complémentaires. Il reste cependant à les fusionner de façon à n'obtenir une seule et même structure de document vidéo contenant l'ensemble des résultats. C'est ce dernier point que nous proposons de réaliser dans le paragraphe suivant.

3.4 Fusion des résultats

Du fait de la similarité des processus pour chacun des deux algorithmes de détection de coupures et de fondus, leur fusion ne pose pas véritablement de problème théorique. En effet, tous deux possèdent le même corps de traitement par comparaison et calcul d'un critère de dissemblance sur deux images successives, puis filtrage des valeurs de critères et comparaison par rapport à un seuil. La partie filtrage hiérarchique ajoutée à la détection de fondus correspond en outre, dans son étape première, à une détection des pics avant tout filtrage, c'est-à-dire des coupures. De plus toute détection de fondus n'apparaît qu'avec un temps de retard sur l'instant traité courant de la séquence et obéit déjà à des règles d'insertion tenant compte d'une certaine structure du document, et notamment d'une détection des coupures.

Le processus de fusion mis en œuvre consiste donc à véritablement effectuer successivement les calculs de deux critères de dissemblance. Le critère utilisé pour la détection de coupures donne ainsi toujours lieu à la même structure de base sur laquelle viennent se greffer par la suite les fondus détectés par le deuxième critère. Le filtrage hiérarchique par succession d'érosions est toujours effectué comme précédemment sur ce deuxième critère mais le premier chapeau haut-de-forme calculé sur la courbe après une érosion de taille 0, i.e. sur la courbe originale, n'est plus réalisé.

En résumé, la suppression de cette première étape de chapeau haut-de-forme sur le deuxième critère de détection de fondu constitue la seule modification à apporter. Cette modification revient à remplacer la structure de coupures extraite lors de la détection de fondus, souvent incomplète (certaines coupures n'apparaissent pas sur le deuxième critère), par celle issue de la détection de coupures, beaucoup plus fiable, comme l'ont prouvé les taux de détection atteints (cf. section 3.2.8).

Au final, la structure de document obtenue n'est autre que la structure issue de la détection de coupures à laquelle ont été rajoutées tous les fondus détectés. Les taux de détection de l'un et l'autre des deux algorithmes sont donc conservés.

3.5 Conclusion - Apports

Ce chapitre clôt la partie "macro-segmentation temporelle" de la structuration linéaire. Deux algorithmes, en essence proches l'un de l'autre, puisque basés tous deux sur le calcul d'un critère de dissemblance locale entre images, de type distance couleur par blocs ou pixel à pixel, suivi d'une étape de filtrage morphologique de la courbe d'évolution temporelle obtenue, ont été présentés.

La combinaison d'un critère de dissemblance très simple et de l'étape de filtrage morphologique, outil puissant de traitement d'images, permet d'aboutir à deux algorithmes simples à mettre en œuvre et laissant envisager une implantation hardware, rapides puisque fonctionnant plus rapidement que le temps réel et atteignant de bons taux de détection. Le choix d'un calcul local de la dissemblance entre images est également un atout important qui conduit, par l'étude de la géométrie des transitions, à une classification de ces dernières.

Séquence	**Nb de fondus originaux**	**% Fondus réels détectés**	**% Fausses alarmes**	**Commentaires**
hard rock	1	100	50	La fausse alarme est due à une prise de vue contenant une seule image.
broderie	1	0	0	Séquence très bruitée.
kart	1	0	0	Séquence très bruitée.
tennis	0	-	0	
vieux tennis	0	-	0	
secte	1	100	0	
lille	0	-	0	
interview	0	-	100	
6 minutes	0	-	0	
travaux	1	100	0	
affaire	1	100	0	
procès	1	100	50	La fausse alarme correspond à un mouvement de caméra trop petit.
visite	0	-	100	La fausse alarme est due à un flash.
colère	1	100	50	La fausse alarme correspond à un mouvement de caméra trop petit.
brèves	1	100	50	
abidjan	0	-	1000	10 fausses alarmes
nantes	1	100	50	La fausse alarme est due à une image seule.
paris	4	100	42.8	
acadie	5	60	64.2	Deux fausses alarmes proviennent d'images seules.
senghor	4	75	44.4	
toulouse	0	-	100	La fausse alarme est due à une image seule.
total	**23**	**78.3**	**65.4**	

TAB. 3.4: Résultats de la détection de fondus sur la base de données de séquences. On rappelle que le taux de fausses alarmes correspond au rapport du nombre fausses alarmes détectées sur le nombre total de fondus détectés, fondus réels et fausses alarmes confondus.

La similarité des processus a en outre permis de mettre en œuvre une méthode de fusion de ces deux algorithmes, de façon à mixer les deux structures de documents obtenues par chacun d'entre eux.

Par la suite, nous verrons qu'il est également possible de réutiliser les critères déjà calculés pour obtenir d'autres informations sur la structure du document vidéo. L'obtention des images clés, une étude du changement interne aux prises de vue détectées, une extraction de mouvement de caméra et la construction de relations entre prises de vue, font ainsi partie des outils supplémentaires bâtis à partir de la base des critères de détection de transitions.

L'inconvénient majeur du second algorithme, i.e. son fort taux de fausses alarmes, sera ainsi en partie résolu dans le chapitre 8, section 8.2, grâce à l'établissement de relations entre les prises de vue. Mais pour l'instant, nous proposons de poursuivre la structuration du document vidéo par la deuxième partie de la segmentation temporelle linéaire : la micro-segmentation, passage de la prise de vue aux images clés.

Chapitre 4

Micro-découpage : de la prise de vue aux images clés

4.1 Introduction

La structuration temporelle d'un document vidéo telle qu'elle a été définie en 2.3.1.1 s'effectue en deux temps : la macro-segmentation, aboutissant à un découpage en prises de vue, et la micro-segmentation qui constitue l'objet de ce chapitre. Il s'agit ici de réaliser un sous-découpage des prises de vue extraites au chapitre 3 en morceaux cohérents du point de vue sémantique, puis d'extraire, pour chaque prise de vue ou morceau de prise de vue, des images représentatives de leur contenu sémantique et informationnel. Ce chapitre s'organise donc autour de ces deux étapes.

La micro-segmentation la plus naturelle, découlant directement du processus de montage du document vidéo, consiste en l'extraction des mouvements de caméra. Cette première passe possède en outre l'avantage de ne pas nécessiter une fois encore d'information sémantique trop élevée. Après un bref rappel des techniques existantes d'extraction des mouvements de caméra en vue de la structuration (cf. section 4.2.1), nous proposons donc notre propre méthode, qui est en fait une extension de l'étude de détection des fondus (cf. section 4.2.2).

Puis, après un rapide état de l'art, une méthode de sélection des images représentatives, cette fois fortement liée au critère de détection des coupures, est présentée (section 4.3.3), suivie d'un bref exposé de la méthode d'obtention d'une mosaïque d'images, dans la section 4.3.5.

Enfin, nous terminons ce chapitre de micro-segmentation temporelle par une étude du changement interne aux prises de vue, dans la section 4.4, permettant, entre autres, de réduire la redondance d'informations par la suppression de certaines images clés.

4.2 Micro-découpage temporel

Nous continuons ici le processus initié dans le chapitre précédent de découpage linéaire temporel, mais cette fois-ci appliqué aux prises de vue.

4.2.1 Extraction du mouvement de caméra

Ainsi que cela a été brièvement rappelé dans l'introduction, la première, et sans doute la plus naturelle segmentation temporelle d'une prise de vue correspond au processus inverse de

la création d'un document vidéo : avant même l'étape de montage, le réalisateur structure déjà chaque prise de vue, en jouant sur les mouvements de caméra dont il dispose pour filmer une même scène.

L'extraction des mouvements de caméra fournit donc une information importante pour l'analyse et la classification des prises de vue.

Davenport *et al.* [29] et Chandler [24] proposent une liste complète des mouvements de caméra existants, dont nous rappelons dans l'annexe A les principales définitions. Notons que la majorité de ces mouvements se décomposent essentiellement en mouvement de translation (travelling, crane, dolly) ou de rotation (pan, tilt). Quant aux effets de zoom, qui ne sont pas à proprement parler des mouvements de caméra, mais plutôt des modifications de la longueur focale, nous les traitons également ici comme faisant partie, au même titre que les mouvements de caméra, des effets utilisés au moment du tournage pour créer un certain contexte, à l'intérieur même de la prise de vue.

De nombreux travaux traitent de l'extraction de tels mouvements de caméra et la majorité d'entre eux reposent sur une étude du flot optique. Ainsi Smith *et al.* [89] examinent les propriétés géométriques du flot optique pour extraire les pannings et les zooms. Ces deux effets se distinguent d'une scène fixe, du fait des caractéristiques différentes des distributions angulaires de leurs vecteurs vitesse. Smith *et al.* utilisent en outre une structure multi-résolution pour réduire le coût en temps de calcul de leur méthode.

Zhang *et al.* [105] proposent également une analyse des schémas caractéristiques des vecteurs de mouvement issus du flot optique dans le but d'extraire les mouvements de caméra classiques. Par ailleurs, d'autres travaux de cette même équipe [106] étudient l'extraction des mouvements de caméra à partir des algorithmes de codage MPEG.

Dans [19, 49, 11], une estimation robuste du mouvement dominant dans la scène est utilisée pour caractériser par la suite des mouvements de caméra de type panning, travelling ou zoom. Les auteurs font donc l'hypothèse que le mouvement dominant de la scène correspond au mouvement de la caméra, et que ce dernier est représenté correctement par un modèle affine à 6 paramètres. Une fois le modèle estimé, l'étude de ces 6 paramètres conduit à une caractérisation des mouvements de caméra de type panning, travelling ou zoom.

D'autres classifications des diverses prises de vue, en fonction du mouvement qu'elles contiennent de façon plus générale, et notamment du mouvement d'objets de la scène, sont également disponibles dans la littérature. Pour un exemple de telle classification en prises de vue de mouvement homogène/hétérogène, localisé, global, etc., le lecteur pourra se référer aux travaux de Hampapur *et al.* [56]. Pour notre part, nous nous limitons dans ce mémoire aux mouvements de caméra.

Après ce survol des méthodes de détection de mouvement de caméra existantes, il est à noter que toutes ces techniques mettent en œuvre des algorithmes d'estimation de mouvement, qui, même s'ils sont implantés de façon à travailler à plusieurs niveaux de résolution, restent tout de même une étape assez lourde du processus de structuration de la vidéo. Nous décrivons donc à présent une retombée directe de l'algorithme de détection de fondus présenté dans le chapitre 3 donnant accès, sans aucun traitement supplémentaire, à une extraction de certains mouvements de caméra.

4.2.2 Utilisation de la détection de fondus

Lors de la section 3.3.2, l'étude des différences d'ordre 1 et 2 (ΔF et ΔF^2) entre images successives d'un document vidéo s'était révélée être source d'informations essentielles à la ca-

ractérisation des trois états *sans transition*, *coupure* et *fondu*. Nous proposons ici de poursuivre cette étude théorique dans le cas de certains mouvements de caméra de type translation, ou travelling.

Les mouvements de type panning ou tilting s'apparentent localement à des travellings, si la rotation mise en œuvre n'est pas trop importante et la scène filmée trop près de la caméra ; pour cette raison, les résultats développés ici s'appliquent également à ces autres classes de mouvement de caméra, même si, dans la suite de ce paragraphe, seul le cas du travelling est détaillé.

Cette fois-ci encore, les deux quantités ΔF et $\Delta^2 F$ conduisent à des courbes de formes géométriques particulières. Nul doute que des caractéristiques de même ordre puissent être extraites et exploitées dans le cas d'autres mouvements de caméra. Le cas du travelling comporte ceci de particulier que les courbes d'évolution obtenues sont très proches de celles d'un fondu, comme nous l'explicitons par la suite.

Pour simplifier encore l'exemple étudié, prenons le cas d'un travelling horizontal de gauche à droite et à vitesse constante, et d'une scène fixe (cf. figure 4.1).

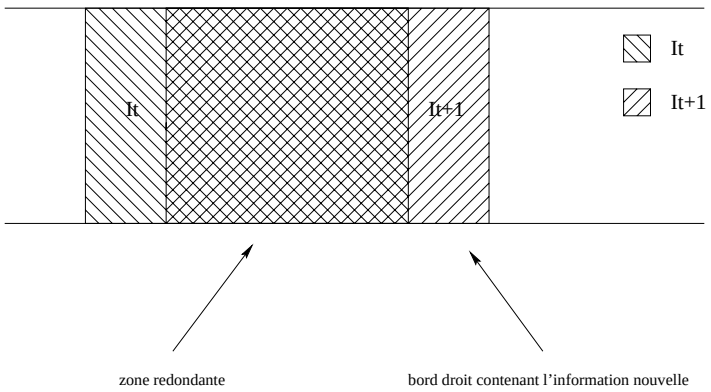


FIG. 4.1: Exemple d'un travelling simple de gauche à droite, à vitesse constante et pour une scène fixe (les deux images I_t et I_{t+1} se superposent en partie).

A l'instant t la caméra voit l'image I_t de la scène. L'image I_{t+1} est en grande partie similaire à l'image I_t (décalée à gauche), seul son bord droit contient de l'information nouvelle.

De même I_{t+2} est similaire à I_{t+1}, sauf sur son bord droit, et est aussi similaire à I_t, sauf sur un bord plus large et toujours placé à droite.

Soit $\vec{v}$ le déplacement apparent de la scène, modélisant le travelling, et m un pixel de l'image. L'équation ci-dessous se vérifie aisément :

$$I_{t+1}(m) = I_t(m + \vec{v}) \tag{4.1}$$

La quantité $\Delta F_{t+1}(m)$ a alors pour expression :

$$\begin{aligned} \Delta F_{t+1}(m) &= I_{t+1}(m) - I_t(m) \\ &= I_t(m + \vec{v}) - I_t(m) \end{aligned}$$

De même :

$$\Delta F_{t+2}(m) = I_{t+1}(m + \vec{v}) - I_{t+1}(m)$$

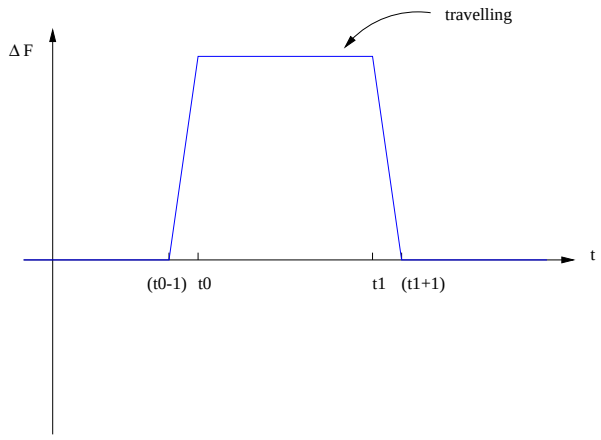


FIG. 4.2: Courbe d'évolution de ΔF dans le cas d'un travelling.

ce qui peut se réécrire sous la forme :

$$\begin{aligned} \Delta F_{t+2}(m) &= I_{t+1}(m+\vec{v}) - I_t(m+\vec{v}) \\ &= \Delta F_{t+1}(m+\vec{v}) \end{aligned} \tag{4.2}$$

Cette équation est vérifiée dans la zone commune aux trois images I_t, I_{t+1} et I_{t+2}. Pour un déplacement $\vec{v}$ de quelques pixels, cette zone couvre pratiquement la totalité de ces trois images. Globalement, dans le cas d'une scène fixe, on aura donc :

$$\Delta F_{t+1} \simeq \Delta F_{t+2}$$

la différence entre ces deux expressions provenant des différences aux bords entre deux images successives. La différence d'ordre 1 devra donc être non nulle et sensiblement constante pendant un travelling (cf. figure 4.2). Quant à la différence d'ordre 2, elle correspond justement aux mesures faites sur le bord des images.

$$\begin{aligned} \Delta^2 F_{t+1} &= \Delta F_{t+1} - \Delta F_t \\ \Delta^2 F_{t+1} &= \Delta(bords) \end{aligned} \tag{4.3}$$

On peut donc cette fois-ci encore extraire des caractéristiques mathématiques correspondant à une structure de travelling. Au vu des courbes 3.34 et 4.2, le fondu et le travelling ont des différences d'ordre 1 très similaires, la distinction se situant alors dans la durée des ces deux événements : un fondu est généralement plus court qu'un travelling.

Toujours avec le même choix de critère de distance couleur pixel à pixel effectué dans la section 3.3.3, les courbes obtenues, dont on fournit un exemple dans la figure 4.3, présentent des dômes, relativement larges comparés à ceux résultant de fondus, pendant la durée des mouvements de caméra.

L'importance d'un calcul pixel à pixel souligné lors de l'étude des fondus apparaît ici encore comme primordiale. L'étude du travelling, comme des autres mouvements de caméra du même type, s'appuie sur une constance de certaines parties d'images au cours du temps, alors que d'autres - les bords de part et d'autre de l'image - sont modifiées. Là encore les zones locales d'apparition des modifications dans l'image sont situées aux bords des objets.

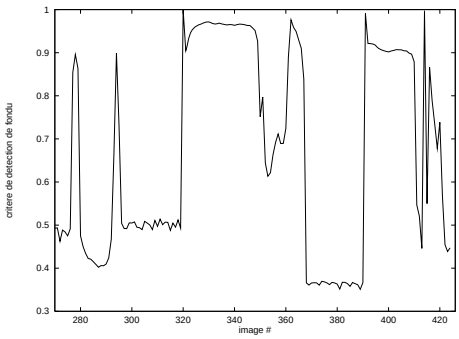


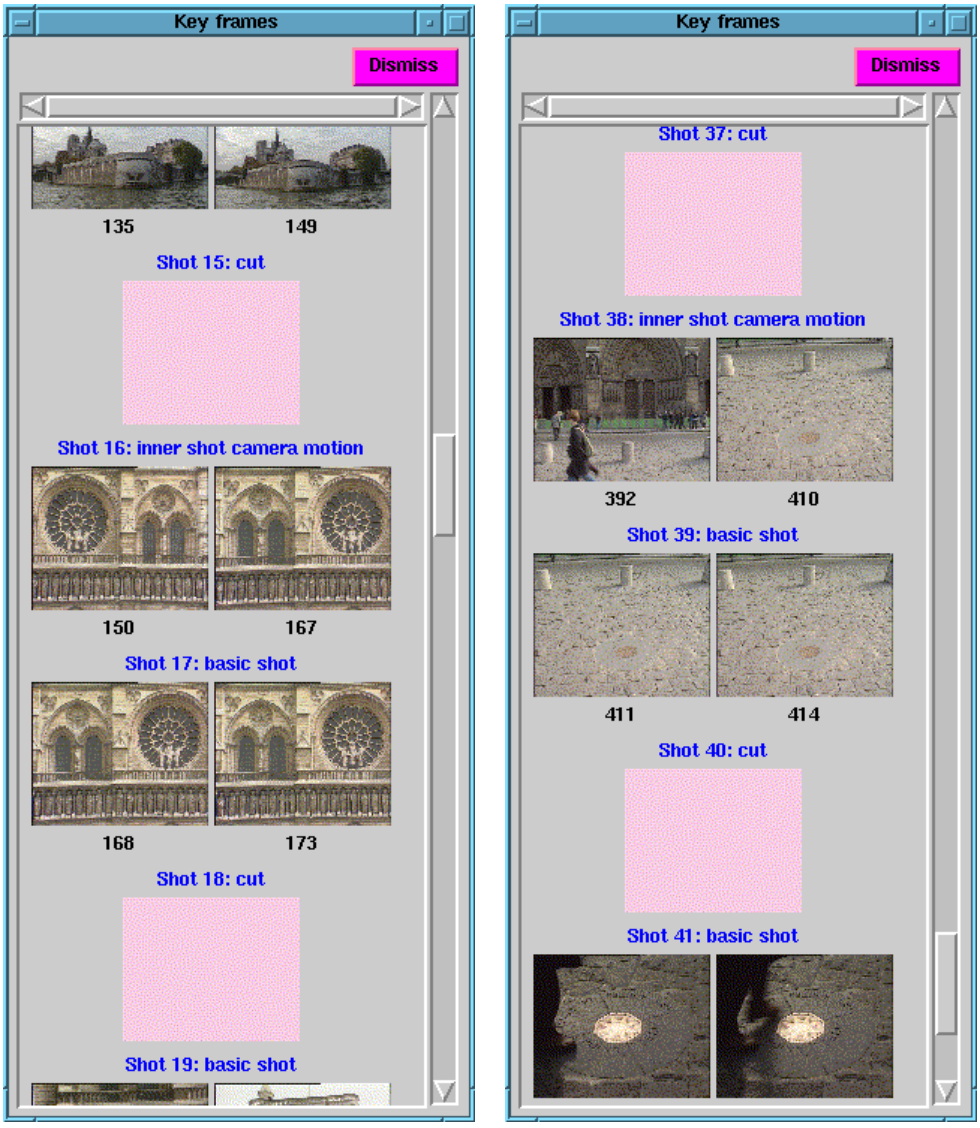
FIG. 4.3: Exemple de courbe d'évolution du critère de détection des fondus pour la séquence *Paris*. Les fondus sont situés sur les images 278-279, 295 et 362-366 (il ne s'agit ici pas exactement d'un fondu mais d'une transition plus compliquée du groupe 3) ; on notera également la présence de mouvements de caméra aux instants 321-349 (panning) et 392-410 (tilting).

Nous venons de démontrer que les mouvements de caméra de type travelling, ou apparentés, se caractérisaient également par une structure de dôme sur les courbes d'évolution du critère. Le même processus d'extraction de ces dômes par filtrage morphologique hiérarchique est donc réutilisé. Il reste alors, une fois tous les dômes de fondus ou de mouvements de caméra extraits, à les distinguer les uns des autres. L'exposé de la technique très simple mise en œuvre pour ce faire fait l'objet de la section suivante. Mais notons dès à présent qu'hormis cet outil de distinction, aucune modification ni ajout ne sont effectués par rapport à l'algorithme original de détection de fondus : il s'agit d'information supplémentaire disponible directement et gratuitement à la sortie de la détection de transitions. Ceci constitue de façon évidente un intérêt annexe majeur de l'algorithme de détection proposé dans ce mémoire, et nous obtenons ainsi une autre preuve que des outils très simples, tant au niveau syntaxique qu'au niveau de la réalisation concrète en traitement d'images, fournissent une quantité d'information non négligeable sur la structure d'un document vidéo, et ceci en contre-partie de faibles temps d'exécution et complexité.

4.2.3 Distinction fondu - mouvement de caméra

La distinction entre un dôme de fondu et celui d'un travelling repose sur la remarque triviale de leur différence de durée : par expérience, alors qu'un fondu ne dure que quelques secondes, 2 au maximum, un mouvement de caméra s'étend sur au moins 5 secondes, avec une moyenne plus proche de la dizaine de secondes. Munis de ces caractéristiques, nous avions déjà établi un seuil de durée maximale d'un fondu, permettant de rejeter les dômes trop larges dans l'algorithme de détection des fondus. En-deçà de ce seuil, placé sans ambiguïté entre la durée maximale d'un fondu (2 secondes) et la durée minimale d'un mouvement de caméra (5 secondes), chaque dôme est étiqueté fondu. Il suffit donc de rajouter une étiquette dans les types existants de prise de vue, pour que les dômes d'une durée supérieure soient étiquetés **mouvement de caméra** (*camera motion*).

La figure 4.4 propose des exemples de mouvements de caméra ainsi extraits. Un autre exemple est également disponible dans la figure 3.42, (b), dans la prise de vue numérotée 33.



(a) (b)

FIG. 4.4: Exemples de mouvements de caméra extraits pour la séquence *Paris* : (a) panning droite-gauche (morceau de prise de vue 16) ; (b) tilting bas-haut (morceau de prise de vue 38).

4.2.4 Bilan

L'extension de l'utilisation de la détection des fondus pour extraire les mouvements de caméra de type translation représente un coût nul en temps de calcul. Par la suite, seule l'insertion supplémentaire dans la structure de document vidéo des morceaux de prise de vue, étiquetés "mouvement de caméra", est consommatrice de ressources, qui cependant restent bien faibles en comparaison avec la mise en œuvre d'un algorithme d'estimation de mouvement tel que ceux décrits dans la section 4.2.1.

Si aucun effort particulier n'est réalisé pour extraire certains mouvements de caméra, il faut toutefois garder à l'esprit que les seuls mouvements détectés sont de type translation, ou localement approximés par une translation : il s'agit donc uniquement des travellings, tiltings et pannings. D'autre part, aucune classification plus fine n'est réalisée par notre technique : l'ensemble des mouvements de caméra de type translation détectés sont regroupés dans une seule et même classe.

D'autres effets de caméra tels que les zooms ne sont pas détectés. Ceci s'explique par le fait que les zooms n'apparaissent nullement sur les courbes d'évolution du critère sous forme de dôme. L'extraction par érosions successives et chapeau haut-de-forme, spécialement conçue pour ces formes caractéristiques ne produit donc aucun résultat.

Les zooms avant se caractérisent expérimentalement par une zone de croissance linéaire des différences d'ordre 1, comme l'illustre la courbe d'évolution présentée dans la figure 4.5. Quant aux zooms arrière, ils correspondent trivialement à des zones de décroissance linéaire (pour un exemple se reporter au début de la courbe d'évolution de la figure 3.38).

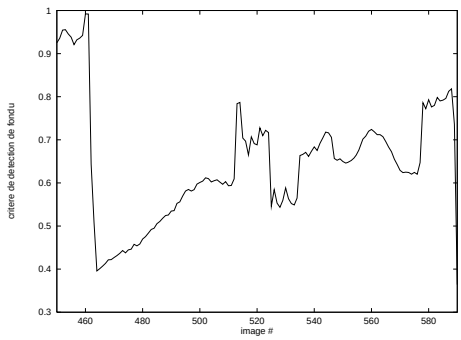


FIG. 4.5: Caractérisation d'un zoom sur la courbe d'évolution du critère de dissemblance utilisé pour la détection des fondus. Deux zooms sont présents aux intervalles $[462; 505]$ et $[550; 560]$.

Des filtrages morphologiques par des éléments structurants adaptés pourraient ainsi très certainement être mis en place, de façon à extraire également ces zones caractéristiques des zooms dans les courbes d'évolution du critère.

Prenons par exemple le cas d'un zoom avant. La première étape d'érosion, par le même élément structurant que pour l'extraction des travellings, résulte en la même zone de croissance linéaire mais décalée d'une image dans le temps. Le résidu de cette opération, obtenu par différence entre la courbe originale et l'érodé, correspond au gradient morphologique interne [82]. Ce gradient apparaît sous la forme de dôme constant (de hauteur, la pente de la zone de zoom), pendant toute la durée du zoom. Il reste alors à appliquer notre outil de détection de dôme sur le gradient pour extraire les zooms avant. Le processus décrit ici, qui n'a pas été réalisé, est illustré de façon schématique dans la figure 4.6.

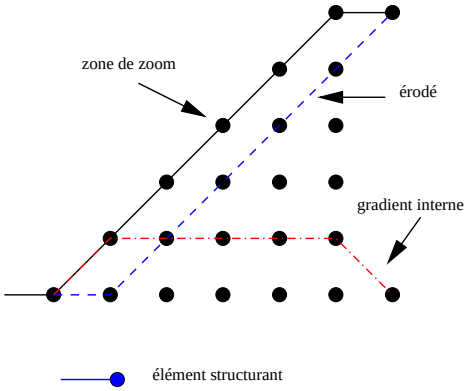


FIG. 4.6: Filtrage morphologique d'une zone à croissance linéaire.

Si la classification obtenue dans le cas des translations n'est pas totale (aucune information sur la direction de translation n'est disponible), la distinction zoom avant-zoom arrière est cette fois-ci envisageable. Les deux courbes étant totalement symétriques l'une de l'autre, il suffit d'éroder, non pas par l'élément structurant utilisé pour les travellings, mais par son symétrique (i.e. l'origine est déplacée de droite à gauche), pour obtenir des dômes correspondant aux zooms arrière, dans le gradient interne.

S'il reste encore des possibilités d'exploration de l'information de type mouvement de caméra à extraire des courbes d'évolution, et même si la caractérisation de certains mouvements ne sera jamais totale, rappelons encore une fois qu'il s'agit ici d'une retombée directe de l'algorithme de détection de fondus, ne représentant aucun travail supplémentaire. Cette extraction doit donc être envisagée comme une première passe à effectuer sur les documents vidéo de façon à n'étudier par la suite que les zones résultant de cette première classification. Des techniques d'estimation de mouvement, mais cette fois-ci uniquement localisée sur une portion de prise de vue, et simplifiée du fait de la première classification (on sait désormais qu'on recherche une translation par exemple), ou des techniques plus géométriques (poursuite de l'étude des masques de transitions) peuvent alors permettre de déterminer à moindre coût la véritable nature du mouvement de caméra rencontré.

Cette classification constitue en outre un premier point de départ idéal pour la réalisation de résumés de prises de vue que sont les **mosaïques**, dont la technique de construction et des exemples sont introduits dans la section 4.3.5.

Nous possédons donc, à ce stade, une structure de document vidéo en diverses prises de vue (au sens de la structure informatique que nous avons définie en 2.4) qu'il est possible de classifier en prise de vue classique ou normale, coupure, fondu ou mouvement de caméra, notre étiquetage s'étant enrichi d'un nouveau type. Bien sûr, dans la structure de document vidéo, non plus telle qu'elle est implantée informatiquement, mais telle qu'elle s'organise, avec ses macro- et micro-découpages, l'étiquette mouvement de caméra appartient à un niveau hiérarchique inférieur, par rapport aux coupures, fondus et prises de vue classiques. Il s'agit en effet du premier type de morceau de prise de vue, que l'on est en mesure d'extraire.

La section suivante propose l'extraction d'une seconde classe de morceaux de prises de vue particuliers que sont les flashes, événements fréquents dans les journaux télévisés, catégorie particulière que nous avons choisie comme application.

4.2.5 Utilisation des flashes

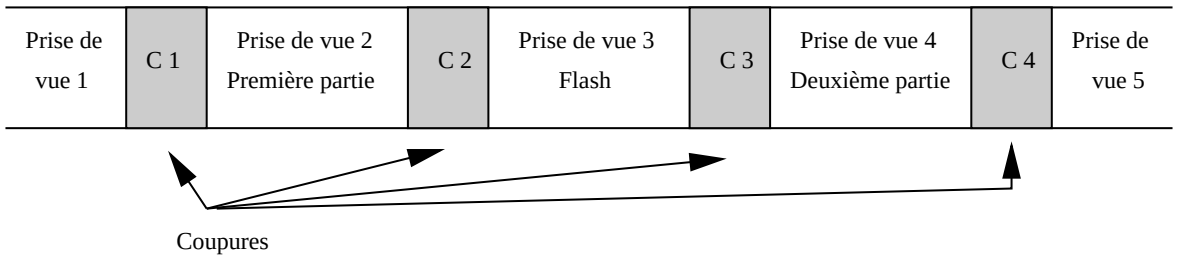
Lors des reportages issus de journaux télévisés, il est bien rare de ne pas rencontrer ces événements courants que sont les flashes d'appareils photo des reporters de la presse écrite. Ces événements parasites sont bien souvent responsables de nombreuses fausses détections lors de l'extraction des transitions d'un document vidéo. Certaines techniques prônent la normalisation de la luminance (ou de la couleur) avant toute détection de transitions, et/ou le choix d'un critère invariant aux changements d'illumination, de façon à s'affranchir de ces fausses détections.

Loin de choisir cette voie, notre alternative est d'accepter les fausses détections dues à des flashes et d'y remédier par la suite par une étude adaptée à ces prises de vue particulières, de façon à les classifier non plus comme prise de vue à part entière, génératrice d'une ou deux fausses détections, mais comme morceau d'une prise de vue plus importante, ce qui ajoute, à notre liste d'étiquettes, un type supplémentaire : le **flash**.

La nouvelle classification de ces morceaux de prises de vue contribue ainsi à la microsegmentation temporelle d'une prise de vue en un ou plusieurs morceaux. D'autre part, l'information, détectée incidemment lors de l'extraction de transitions, de présence d'un flash dans une prise de vue donnée est conservée.

Suite à la segmentation en plans d'un document vidéo, un flash donne lieu à une, ou à deux fausses alarmes, suivant que la prise de vue initiale est découpée en deux ou trois morceaux. Dans le premier cas, la coupure parasite intervient soit avant, soit après le flash qui se retrouve donc entièrement inclus dans l'un ou l'autre des deux morceaux de prises de vue résultants. Dans le second cas, le flash est extrait comme prise de vue à part entière, entouré de deux coupures parasites et la prise de vue initiale se poursuit de part et d'autre.

Dans le cas d'une fréquence de comparaison de 5Hz entre images du fichier vidéo, les flashes détectés correspondent à ce deuxième cas et sont en outre le plus souvent restreints à une seule image. On observe donc, après la détection des transitions les situations erronées suivantes :



L'algorithme mis en place pour détecter et corriger ce genre d'événement repose sur trois propriétés :

- La prise de vue **3** possède une valeur de luminance moyenne bien supérieure à celles des prises de vue **2** et **4**.
- Les prises de vue **2** et **4** sont deux morceaux d'une seule et même prise de vue. Elles ont donc des contenus similaires.
- La prise de vue **3** ne contient qu'une seule image à la fréquence de comparaison de 5Hz.

Une fois la détection de transitions réalisée, la structure obtenue est parcourue et seules les prises de vue ne contenant qu'une seule image (troisième critère) sont retenues : pour ces prises de vue uniquement, on teste la similarité de contenu des deux prises de vue voisines. Ceci est réalisé grâce à l'outil de détection de relation entre prises de vue que nous avons déjà

évoqué et qui sera détaillé dans le chapitre 7. Pour l'instant, retenons que seules les images clés de chacune des prises de vue sont testées et qu'en fonction de leur similarité, on déduit ou non la présence d'une relation, i.e. d'une similarité de contenu entre les deux prises de vue.

Lorsque les deux prises de vue, de part et d'autre d'une prise de vue candidate pour être étiquetée flash, sont déterminées comme étant en relation, on procède alors au dernier test de calcul de la différence de luminance moyenne entre la prise de vue avant et le flash d'une part, et entre la prise de vue après et le flash d'autre part. Dans le cas où ces deux différences sont positives, i.e. il y a bien eu croissance, puis décroissance de la luminance, la valeur minimale est conservée comme donnant une mesure de la probabilité de présence d'un flash. Cette valeur est stockée comme un attribut particulier du morceau de prise de vue intermédiaire, qui est alors étiqueté **flash**. Une amélioration possible de l'algorithme consisterait à utiliser ici une différence de luminance normalisée et à vérifier, au contraire, la similarité de contenu entre les prises de vue 2, 3 et 4, par l'établissement de relations. Un tel critère est en effet plus robuste que le test d'une simple croissance de la luminance moyenne, qui, si elle était utilisée toute seule, n'est pas uniquement caractéristique de la présence de flashes.

Les deux sélections successives (prise de vue d'une seule image et dont les prises de vue voisines sont en relation) permettent une performance exceptionnelle en terme de temps de calcul de notre méthode. Très peu de prises de vue sont en effet candidates après le test des deux premiers critères, qui sont eux-mêmes de durée négligeable : dans le premier cas il s'agit de comparer l'attribut *Nombre_Images* à la valeur 1 et dans le second cas, si les relations ont déjà été établies auparavant (cf. chapitre 7), il s'agit de vérifier que la prise de vue $(i+1)$ est présente dans la liste de relations de la prise de vue $(i-1)$.

L'algorithme 3 résume le processus complet, sous une forme plus proche de la réalisation concrète informatique.

Algorithme 3 Algorithme de détection de flash.

pour tout P_i, prise de vue **faire**
  si nombre d'images $(P_i) = 1$ **alors**
    si P_{i-1} et P_{i+1} sont en relation **alors**
      Calcul de la luminance moyenne I_i de P_i
      Calcul de la luminance moyenne I_{i-1} de P_{i-1}
      Calcul de la luminance moyenne I_{i+1} de P_{i+1}
      si $I_i - I_{i-1} > 0$ **et** $I_i - I_{i+1} > 0$ **alors**
        Présence_Flash $(P_i) \leftarrow \min(I_i - I_{i-1}, I_i - I_{i+1})$
        Etiquetage de P_i en tant que **flash**
        Suppression des transitions de type coupures de part et d'autre de P_i
      fin si
    fin si
  fin si
fin pour

Dans la base de données disponible, après détection des transitions, trois flashes ressortent

comme prises de vue à part entière. Ces trois exemples sont en outre correctement modélisés par les trois hypothèses ci-dessus. Ils sont parfaitement détectés et la structure du document vidéo est aisément modifiée en :

Prise de vue 1	C 1	Morceau 1 Prise de vue 2	Morceau 2, Flash Prise de vue 2	Morceau 3 Prise de vue 2	C 2	Prise de vue 3

Nous proposons dans la figure 4.7 une illustration concrète d'une telle structure avant et après modification sur un exemple réel.

En résumé, la méthode détaillée ici possède deux applications, dépendantes l'une de l'autre : la détection de flash permet en effet à la fois de supprimer des fausses alarmes de la détection de transitions, et de franchir une étape supplémentaire dans l'élaboration de la micro-segmentation des prises de vue.

Bien sûr la micro-segmentation temporelle n'est pas terminée et toute information supplémentaire telle que le mouvement d'objets d'intérêt, ou la présence d'autres informations de nature plus sémantique permet de contribuer à son élaboration. Dans le cadre de ce mémoire, pour satisfaire à notre objectif de simplicité des techniques mises en œuvre afin de maximiser le rapport qualité de l'analyse/vitesse, nous arrêtons là l'extraction de morceaux de prises de vue et nous continuons à présent notre descente dans la hiérarchie des entités structurantes du document vidéo : pour chaque prise de vue ou morceau de prise de vue, il s'agit à présent de ne conserver qu'une représentation la plus succincte possible de leur contenu.

4.3 Représentation des structures prises de vue ou morceaux de prises de vue

4.3.1 Introduction

La structuration temporelle du document vidéo se poursuit ici par une sélection d'images représentatives du contenu sémantique, ou bien par la construction de mosaïques d'images. Dans les cas, ces deux processus impliquent la mise en œuvre d'une forme de structuration légèrement différente.

Il ne s'agit plus véritablement de segmentation au sens partitionnement, i.e. l'union de tous les morceaux de la partition forme l'objet de départ, comme c'était le cas lors du découpage en séquences, scènes, prises de vue et morceaux de prises de vue. La notion de partition s'accompagne en effet d'une idée de segmentation sans perte : des informations sur la structure et l'organisation du document vidéo sont ajoutées, mais ce dernier n'est en rien modifié, ni altéré, ni réduit.

L'extraction d'images représentatives, ou *images clés*, sous-entend au contraire que seules certaines images sont sélectionnées, au détriment d'autres jugées ne pas apporter d'information sémantique supplémentaire. Il s'agit d'un processus d'abstraction, pour lequel la notion de partionnement ne s'applique plus, puisqu'il est impossible à partir des seules images clés de reconstruire à l'identique le fichier vidéo original. Par contre, le contenu sémantique du document vidéo doit, quant à lui, rester intact. Ici encore les notions de fichier vidéo et document vidéo s'opposent.

On aboutit ainsi à la définition d'une image clé, représentative du contenu sémantique d'une entité. Remarquons qu'une image clé n'est reconnue en tant que telle qu'ajoutée à

(a) (b)

FIG. 4.7: Modification de la structure de document vidéo dans le cas d'un flash : (a) Avant correction, deux fausses alarmes sont présentes dues à la présence d'un flash détecté comme prise de vue à part entière. (b) Après correction, le flash est intégré dans la prise de vue qui se poursuit de part et d'autre.

l'ensemble des autres images clés extraites pour une même entité.

Définition 16. Image clé (*keyframe*) *Une image clé est une image représentative d'une partie du contenu informationnel d'une entité (prise de vue ou morceau de prise de vue) du document vidéo. L'ensemble des images clés extraites pour une entité doit permettre de reconstituer tout le contenu sémantique de cette entité, sans redondance, ni perte d'information. Il s'agit d'un ensemble minimal, en ce sens que la suppression d'une image clé conduit automatiquement à la perte d'informations sémantiques.*

Cette définition pose de façon évidente le problème de la caractérisation de ce qui est de l'information et ce qui n'en est pas. En théorie les images clés devraient correspondre à des primitives sémantiques telles que les objets d'intérêt, des actions ou des événements nouveaux. En pratique, et dans le cadre d'une extraction automatique, une analyse d'un si haut niveau sémantique n'est pas encore réalisable. Aussi la majorité des techniques existantes, la nôtre y compris, se basent-elles sur des critères bas niveau calculés sur toutes les images d'une prise de vue pour effectuer leur sélection. Nous en proposons dans la section suivante un rapide état de l'art.

4.3.2 Etat de l'art

Dans le problème de l'extraction des images clés d'une entité donnée, deux points de vue s'opposent :

- soit l'information sémantique la plus intéressante est à rechercher dans les zones stables présentes dans l'entité et les images clés sont alors à extraire à l'intérieur des ces zones et donc en dehors des périodes de transition, forcément plus mouvementées ;
- soit au contraire, ces périodes de transition, que sont par exemple l'apparition d'une incrustation de texte, signifient un changement et donc l'apparition de nouvelles informations sémantiques, et c'est justement à ces moments-là, ou juste après, qu'il convient de sélectionner les images clés.

Les premières techniques proposées dans la littérature effectuaient un choix arbitraire des images clés, peu représentatif du contenu informatif de chaque prise de vue, puisque les première et dernière images de l'entité étaient sélectionnées, auxquelles étaient ajoutées soit l'image du milieu, soit un certain nombre d'images prises à un taux d'échantillonnage fixé. Face à l'avantage évident de ne nécessiter aucune information sémantique, de telles méthodes possèdent également l'inconvénient majeur d'aboutir à une sélection aveugle, quelle que soit l'entité, donc bien souvent redondante et non représentative de l'intégralité du contenu sémantique.

Or il est à présent évident que le choix d'images clés dépend fortement du type de contenu du fichier vidéo traité : une seule image clé suffit ainsi à représenter une prise de vue de présentateur de journal télévisé, alors que plusieurs seront nécessaires dans le cas d'une prise de vue contenant des objets d'intérêt en mouvement.

Plusieurs auteurs ont donc cherché à améliorer le choix des images clés en se basant sur des critères divers de mouvement, de couleur, de présence de visage, etc.

Zhang *et al.* [105] étudient ainsi l'évolution de critères tels que la couleur dominante, les moments statistiques de couleur, les histogrammmes couleur et la luminance moyenne, couplés aux mouvements de caméra et d'objets importants en taille, pour effectuer leur sélection d'images clés, à laquelle ils ajoutent les première et dernière images de l'entité. Ils poursuivent et étendent leurs travaux dans [107], afin de travailler directement sur un flot comprimé de type

MPEG et de rajouter une option supplémentaire de choix manuel de la densité des images clés pour chaque entité. Dans ce cas, leur algorithme atteint le temps réel et aboutit à l'extraction de deux à trois images clés par prise de vue, ce qui, d'après eux, reste trop généreux par rapport aux tests effectués par des opérateurs humains.

Dans [99], Wactlar *et al.* procèdent successivement à une détection de mouvement, de régions de couleur peau et de texte sur toutes les images de l'entité, pour sélectionner les images clés répondant positivement à l'un au moins de ces trois critères (la réponse positive à l'un d'entre eux sur une image donnée provoquant l'arrêt du processus pour cette image). Par défaut, si aucune image n'a pu être extraite, l'image milieu est conservée. L'inconvénient majeur est ici bien sûr le temps d'exécution et la complexité des trois détections successives menées sur chaque image.

Smith *et al.* [89] proposent une technique similaire puisqu'ils sélectionnent leurs images clés en fonction du niveau de priorité suivant : tout d'abord, les images contenant des visages ou du texte, puis les images statiques succédant à un mouvement de caméra, puis les images contenant à la fois un mouvement de caméra et des visages ou du texte, et enfin, par défaut, les images de début de prise de vue. Mais encore une fois toutes les images d'une prise de vue donnée sont étudiées, pour au final l'obtention d'une redondance importante de l'information dans leur sélection.

D'autres auteurs [26] utilisent l'algorithme de détection des coupures de Zabih *et al.* [104], pour extraire les images-clés. Rappelons que cette technique est basée sur une étude des bords entrants et sortants dans les images. L'hypothèse réalisée est alors la suivante : les périodes de changements sémantiques que l'on désire extraire comme images-clés font partie des images contenant une forte proportion de bords entrants ou sortants. Dans le cas d'une séquence contenant beaucoup de mouvements, conduisant à l'extraction d'un trop grand nombre d'images clés, les auteurs retournent alors à une sélection régulière des images représentatives.

Enfin nous terminons cet état de l'art par les travaux de Yeung *et al.* [102], qui exposent une technique d'un niveau sémantique et d'une complexité moins élevées, et en ce sens proche de ce que nous proposons dans la section suivante. Ils commencent par sélectionner la première image de la prise de vue, puis ils comparent successivement toutes les images suivantes jusqu'à trouver une image suffisamment différente de la première image clé, suivant un critère de dissemblance donné et un seuil défini expérimentalement. Cette nouvelle image est ajoutée à l'ensemble des images clés et devient le nouveau point de départ du processus pour toutes les images restantes. La comparaison n'est pas effectuée directement sur les images originales mais sur des images échantillonnées spatialement (typiquement les images DC et DC+2AC extraites des flots MPEG et M-JPEG) pour se ramener à des tailles de 40×30 pixels. Ils aboutissent ainsi à une sélection de 2% à 10% d'images clés par prise de vue.

4.3.3 Etude de la deuxième hiérarchie de pics

Pour continuer dans le même esprit que ce que nous avons présenté jusqu'à présent, notre méthode d'extraction d'images clés se doit de ne pas utiliser d'informations d'un niveau sémantique trop élevé. Aussi, tout comme Yeung *et al.* [102], les images sont-elles sélectionnées lorsqu'elles produisent une mesure de dissemblance trop élevée par rapport à un seuil fixé à l'avance, que nous appelerons *Seuil_ImageClé*.

Cependant le processus de sélection de Yeung n'est pas exactement reproduit ici puisqu'on ne compare pas la dernière image clé sélectionnée successivement avec toutes ses suivantes, jusqu'à en trouver une qui soit suffisamment dissemblable. Notre algorithme est plus simple en

ce sens qu'une image donnée n'est comparée qu'avec celle lui succédant immédiatement. Dans le cas où la valeur de critère de dissemblance est supérieure au seuil *Seuil_ ImageClé*, les deux images sont conservées comme images clés. Un tel processus nous a été dicté par l'étude des courbes d'évolution obtenues lors de la détection de coupures (cf. section 3.2). Ces courbes, une fois filtrées par chapeau haut-de-forme inf, mènent en effet à deux hiérarchies de pics de hauteurs différentes :

- la première, regroupant les pics les plus élevés, constitue l'ensemble des coupures extraites pour chaque fichier vidéo ;
- la seconde contient, quant à elle, toute une série de pics plus bas (figure 4.8) qui correspondent aussi à des changements, mais moindres, entre images successives.

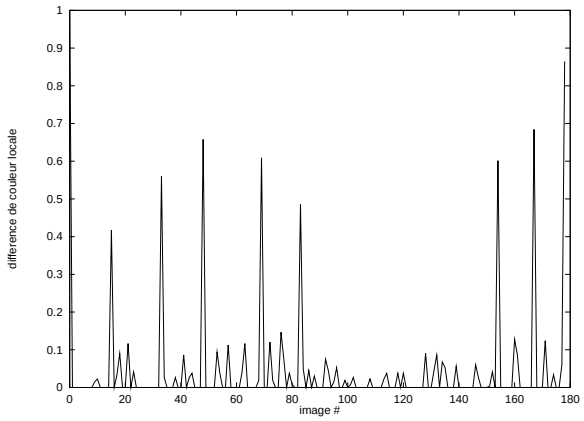


FIG. 4.8: Illustration de la première et de la deuxième hiérarchies de pics extraits des courbes d'évolution du critère de changements de scènes.

La sélection d'images caractéristiques proposée correspond exactement à l'extraction de cette deuxième hiérarchie de pics, signes de changements internes aux prises de vue.

Le principe mis en œuvre présente l'énorme avantage de ne pas nécessiter d'autres calculs que ceux déjà réalisés lors du critère de détection de transition. Une simple comparaison supplémentaire entre les valeurs de critères et le nouveau seuil de détection d'images clés est ajoutée. En pratique, ce seuil est automatiquement fixé à la moitié du seuil global, utilisé pour la détection des coupures, i.e. de la première hiérarchie de pics.

A cette sélection automatique d'images correspondant aux changements internes de la prise de vue, sont ajoutées les première et dernière images de chaque prise de vue. Cet ajout fournit une sélection amplement suffisante de la partie "stable" de la scène représentée, de cette façon les deux aspects de l'extraction des images clés rappelés en tout début de la section précédente sont présents dans notre approche.

Avant de détailler les résultats obtenus par notre méthode, nous désirons effectuer rapidement une remarque sur une autre approche testée et qui s'est finalement révélée inexploitable [36]. Dans le cadre d'une extraction des images clés dans les parties stables d'une prise de vue, et si on cherche toujours à exploiter le critère de dissemblance utilisé pour la détection de transitions, les instants les plus stables doivent correspondre aux valeurs les plus faibles du critère, donc aux pics inversés sur les courbes d'évolution.

L'opérateur dual du chapeau haut-de-forme blanc, le *chapeau haut-de-forme noir* dont une définition est disponible dans l'annexe B, permet d'extraire de telles formes locales sur les

courbes. Cependant après quelques tests destinés à déterminer à quels événements correspondaient ces pics inverses, il s'est avéré qu'aucune information sémantique exploitable ne leur était directement liée.

4.3.4 Illustration des résultats - Conclusion

L'extraction de la deuxième hiérarchie de pics mène à la sélection d'en moyenne 2.1% d'images clés par fichier vidéo sur l'ensemble de la base de données dont nous disposons, ce qui représente entre 1 et 3 images par prise de vue. Cette valeur est donc tout à fait comparable avec les résultats cités par Yeung *et al.* [102].

Toutefois, une étude plus approfondie et qualitative de la sélection effectuée conduit à la conclusion que pour moitié les images extraites ne contiennent pas d'information nouvelle. L'autre moitié cependant correspond bien à un contenu sémantique neuf, tel que l'apparition d'une incrustation de texte, un mouvement d'objets, un flash n'ayant pas provoqué de fausse alarme lors de la détection de transition, une transition perdue, etc.

Plusieurs exemples d'images extraites sont fournis dans le cas où un gain d'information est réalisé et dans le cas contraire où l'information est redondante, dans la figure 4.9. Il est à noter que chaque prise de vue est au moins représentée par deux images clés toujours présentes, quel que soit le résultat de la détection de la deuxième hiérarchie de pics, qui sont les première et dernière images.

A ce stade, rappelons les définitions de deux critères de mesure de performance très utilisés dans le domaine de la recherche d'images dans une base de données, mais qui s'applique également très bien à un outil tel que l'extraction des images clés.

Définition 17. Précision *[75] La précision P est définie comme la proportion d'images pertinentes trouvées par rapport au nombre d'images trouvées :*

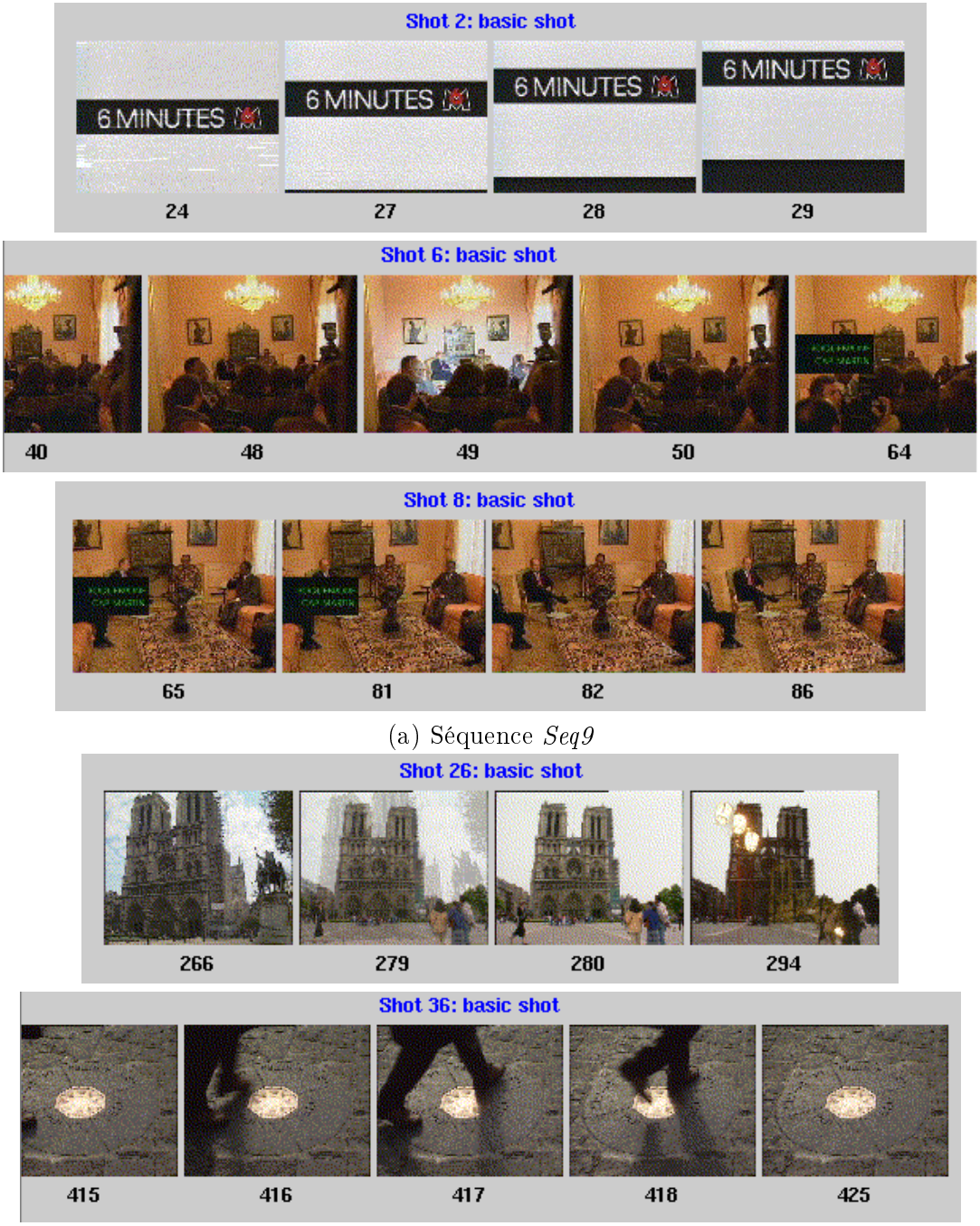
$$P = \frac{\textit{nombre d'images pertinentes trouvées}}{\textit{nombre d'images trouvées}} \tag{4.4}$$

Définition 18. Rappel *[75] Le rappel R est défini comme la proportion d'images pertinentes trouvées par rapport au nombre d'images pertinentes :*

$$R = \frac{\textit{nombre d'images pertinentes trouvées}}{\textit{nombre d'images pertinentes}} \tag{4.5}$$

L'algorithme détaillé ici atteint donc des valeurs de précision peu attrayantes, de l'ordre de 65%. Par contre, au vu de tests uniquement qualitatifs, et lorsqu'on ne tient compte que des changements brutaux dans les images, les valeurs de rappel obtenues sont nettement plus élevées ($> 90\%$). Lors de changements plus progressifs comme par exemple un mouvement de caméra, aucune image clé n'est extraite : le critère de détection choisi n'est en effet pas adapté à des transitions ou changements plus progressifs, puisque l'élaboration d'un deuxième critère spécialement dédié à ce type de modifications a dû être élaboré, dans la section 3.3.

Face à ces deux inconvénients, la technique présente l'énorme avantage de proposer un coût pratiquement nul en terme de calcul supplémentaire, une fois la détection de transitions réalisée. Le choix du seuil *Seuil_ImageClé* n'est en outre pas directement laissé au choix de l'utilisateur, puisqu'il est automatiquement indexé sur la valeur du seuil local, ce qui permet de considérer la méthode comme sans paramètres. Au vu de ces avantages, nous avons donc cherché à remédier à la redondance d'informations et au problème des événements plus



(a) Séquence *Seq9*

(b) Séquence *Paris*

FIG. 4.9: Exemples d'images clés supplémentaires issues de la deuxième hiérarchie de pics. Dans le cas de la prise de vue 8, (a)), la détection d'incrustation de texte fournit une information redondante, du fait de l'ajout des images de début et de fin. Par contre on détecte des effets spéciaux (prise de vue 2, (a)), des flashes (prise de vue 6, (a)), des fondus oubliés (prise de vue 26, (b)) et des mouvements d'objets en gros plan (prise de vue 36, (b)).

progressifs par la suite, plutôt que de construire un autre algorithme, plus coûteux et plus sophistiqué.

Nous exposons ainsi dans la section 4.4 un outil de détection de changement au sein d'une même prise de vue permettant de revenir sur le choix des images clés de façon à en supprimer la redondance. Cette nouvelle information de changement peut s'avérer utile pour déterminer si une seule image caractéristique est suffisante à la description du contenu sémantique, ou au contraire si d'autres outils doivent être initiés, dans le but d'obtenir des informations supplémentaires.

Parmi ces informations nouvelles alors extraites, l'information de mouvement notamment, et particulièrement de mouvement de caméra, non retranscrite dans le premier choix d'images clés, peut ainsi conduire soit à la sélection d'images supplémentaires, soit à la création de mosaïques d'images, sans doute alors la meilleure façon de représenter ce type de changements. Cette information est en outre en partie déjà disponible, ainsi que nous l'avons détaillé dans la section 4.2.2.

La section suivante est l'occasion de rappeler ce que l'on entend par mosaïque d'images et de faire quelques remarques sur l'apport des outils déjà mis en place dans ce mémoire comme aide à la construction de telles représentations.

4.3.5 Mosaïques d'images

La mosaïque d'images est une représentation très élégante du contenu d'une scène, lorsque celle-ci a été filmée par une caméra. Il s'agit d'une image virtuelle, obtenue par fusion des contenus des images successives d'une prise de vue. Plus grande que les images individuelles, elle permet une représentation globale de la scène filmée. Par exemple, lors d'une prise de vue acquise par une caméra en translation, la mosaïque consiste en une bande horizontale plus large, correspondant à l'ensemble de la scène visible.

Concrètement le processus de création d'une mosaïque d'images s'effectue en trois étapes principales [30]. On commence par estimer le mouvement dominant entre les images successives de la prise de vue. Cette extraction du mouvement dominant consiste en sa modélisation par une homographie et en l'estimation des paramètres de cette homographie.

Une fois le mouvement dominant calculé, les images de la prise de vue sont recalées par rapport à un référentiel commun, par exemple celui de la première image de la séquence.

Chaque pixel de la mosaïque prend alors pour valeur la médiane de ses valeurs dans chacune des images superposées. Ce filtrage constitue la troisième et dernière étape de la construction d'une mosaïque, dont la figure 4.10 fournit une illustration.

Bien sûr, plusieurs variantes de construction existent, mais toutes sont basées sur le même mode opératoire ; les différences se situent alors dans le choix de la technique d'estimation du mouvement dominant (réalisée, par exemple, soit par flot optique, soit par corrélation, soit par recalage de primitives), le choix du référentiel de recalage (certains travaux prônent le recalage dans le référentiel-même de la mosaïque, de façon à éviter l'accumulation d'erreurs au fur et à mesure des recalages successifs) et l'opérateur final de filtrage.

Si la construction pratique de la mosaïque d'une prise de vue est toujours réalisable, la représentation obtenue n'est cependant correcte que dans les deux cas suivants :

- la scène est plane et, dans ce cas, le mouvement de caméra peut être quelconque. Cependant lorsque la scène filmée est assez loin de la caméra, l'approximation qui consiste à la considérer plane est possible ;

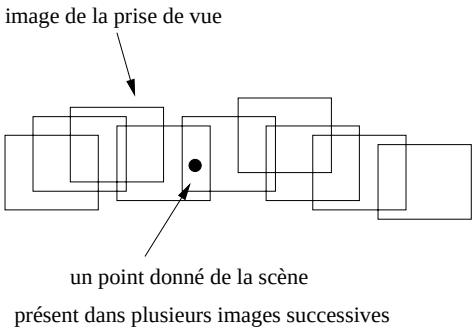


FIG. 4.10: Illustration du principe de construction d'une mosaïque d'images à partir des images d'une prise de vue.

– la scène est quelconque, le mouvement de caméra doit alors être de rotation uniquement.

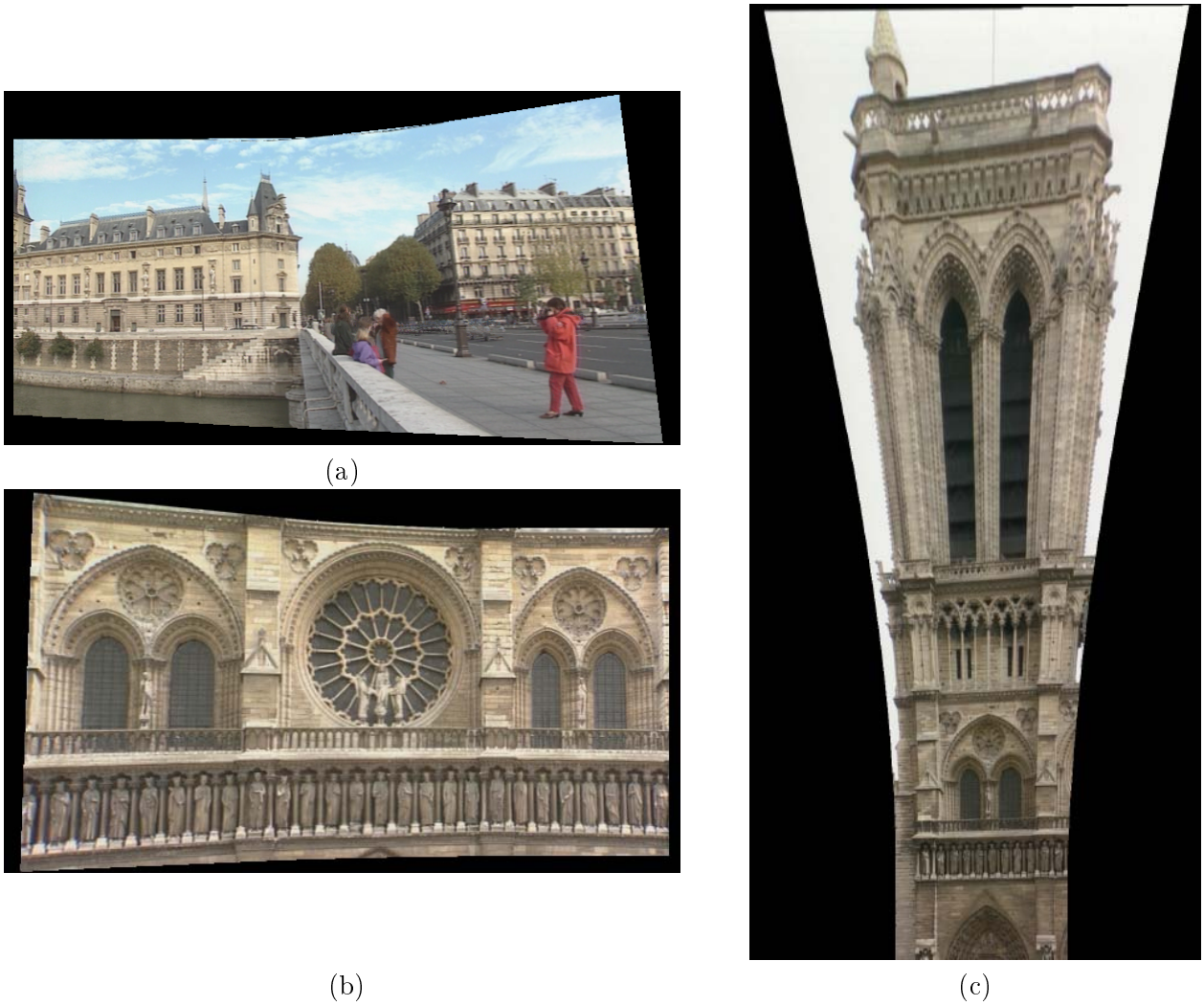
Avec l'une ou l'autre de ces hypothèses, la mosaïque fournit donc une représentation plus globale de la scène [96], par rapport à une collection d'images clés, qui ne constituent somme toute que des points de vue très locaux. Ceci est évidemment particulièrement crucial dans le cas d'un mouvement de caméra, et moins nécessaire lorsque la caméra est fixe. A ce premier atout de l'utilisation d'une mosaïque, s'ajoutent en outre la possibilité de supprimer les objets en mouvement de la scène filmée et/ou de représenter leurs trajectoires globales sur la mosaïque, et donc dans la scène. Ces deux points sont la conséquence directe du filtrage médian final. Ces deux informations supplémentaires sont appréciables dans le cadre d'un processus d'indexation. La représentation de trajectoires est utile notamment dans le cas de requêtes concernant des actions ou des événements.

Ces diverses caractéristiques et propriétés d'une mosaïque d'images étant énoncées, quelques problèmes subsistent. Tout d'abord, il est nécessaire, dans le cadre de l'indexation, de se poser la question : quand initier la construction d'une mosaïque ? Comme nous l'avons déjà évoqué ci-dessus, une telle représentation n'apporte aucune information supplémentaire par exemple dans le cas d'une caméra fixe ; au contraire son intérêt dans le cas d'une caméra en mouvement est primordial. Ensuite, même dans le cas où la mosaïque est susceptible d'apporter une nouvelle information, certaines prises de vue ne permettent pas d'aboutir à une représentation correcte de la scène, soit lorsque les hypothèses de représentation correcte ne sont pas vérifiées, soit lorsque le mouvement dominant extrait de la prise de vue n'est pas le mouvement de la caméra, mais le mouvement d'un objet en gros plan dans la scène, par exemple. Enfin la technique de construction de mosaïque sous-entend que la caméra est idéale, ce qui malheureusement n'est pas toujours le cas. On assiste alors à l'apparition de distorsions non linéaires dans la mosaïque (cf. mosaïque (c), figure 4.11).

Dans le cadre de ce mémoire, aucun travail particulier d'amélioration du processus lui-même de construction de la mosaïque n'a été entrepris. Nous désirons simplement mettre en relief ici une utilisation possible de notre première classification des morceaux de prises de vue en tant que mouvements de caméra, exposée au paragraphe 4.2.2. Cette classification représente en effet un élément de réponse au premier problème de décision de la construction d'une mosaïque. La classification que nous proposons permet de sélectionner les prises de vue, et même plus précisement les morceaux de prises de vue, pour lesquels il sera intéressant de construire par la suite une mosaïque d'images.

Trois exemples de mosaïques réalisées sur les morceaux de prises de vue étiquetées mouve-

ment de caméra, par notre algorithme, et réalisées par un algorithme classique , sont proposés à titre d'illustration dans la figure 4.11.



(a) (b) (c)

FIG. 4.11: Exemples de mosaïques d'images obtenues pour quelques prises de vue ou morceaux de prises de vue étiquetés mouvement de caméra, extraits de la séquence *Paris* : (a) prise de vue 2 ; (b) Morceau de prise de vue 16 ; (c) Prise de vue 19. Pour comparaison, les images clés extraites correspondant à la mosaïque (b) ont été fournies dans la figure 4.4.

Ajoutons pour terminer que la mosaïque vient fournir, en contre point des images clés extraites, correspondant, dans le cadre de nos travaux, aux instants de changements dans la prise de vue, une représentation non seulement globale mais aussi, et surtout, stable de la scène (cf. section 4.3.2).

Nous proposons à présent de revenir sur l'ensemble des images clés extraites, et surtout sur un outil de caractérisation du changement interne d'une prise de vue donnée, dont l'utilisation principale, détaillée dans le paragraphe suivant, est la suppression de la redondance contenue dans l'ensemble des images clés.

4.4 Etude du changement interne

Notre sélection d'images clés dans une prise de vue mène à la conclusion qu'il y a bien souvent redondance d'informations, ce qui est contradictoire avec la notion d'ensemble minimal (cf. définition 16), qui doit être la qualité principale de l'ensemble des images représentatives pour une entité donnée. L'objectif de cette section est alors le suivant : en se plaçant en aval du choix des images représentatives, il s'agit de détecter celles qui sont redondantes dans le but de les supprimer.

L'outil mis en place prend le nom de détecteur de changements internes aux prises de vue dans la mesure où la suppression d'images clés tend à confirmer l'hypothèse que la prise de vue concernée se résume en peu d'images, et donc contient peu de changement, le cas extrême étant bien sûr la conservation d'une unique image représentative pour toute une prise de vue, conduisant à la conclusion que cette dernière ne contient pas du tout de changement. Le nombre d'images clés restant après suppression de la redondance donne donc une mesure du changement interne à la prise de vue.

No de prise de vue	0	1	2	3	4	5	6	7	8	9
Changement	non	non	non	non	non	non	non	non	non	non
No de prise de vue	10	11	12	13	14	15	16	17	18	19
Changement	non	non	non	non	oui	non	non	non	non	non
No de prise de vue	20	21	22	23	24	25	26	27	28	29
Changement	non	non	non	non	oui	non	oui	non	non	non

FIG. 4.12: Tableau des changements internes aux prises de vue détectés pour la séquence *travaux*.

Concrètement la réalisation de l'outil de détection de changement est encore une fois très simple, et le même critère de dissemblance appliqué lors de la détection de coupures est réutilisé.

Supposons que l'outil de détection d'images clés ait donné lieu à l'extraction de k images pour une prise de vue donnée. On note ces k images $K_1, K_2, \ldots, K_k$. L'opérateur proposé étudie tous les triplets successifs (K_i, K_{i+1}, K_{i+2}) à l'aide du critère de dissemblance, comparé cette fois-ci à un nouveau seuil, le **seuil de changement**.

On décide de la suppression de l'image clé K_{i+1} lorsque le triplet (K_i, K_{i+1}, K_{i+2}) est sans changement, i.e. lorsqu'aucune des deux paires (K_i, K_{i+1}) et (K_{i+1}, K_{i+2}) n'a de valeur de critère supérieure au seuil de changement. Au contraire, un triplet (K_i, K_{i+1}, K_{i+2}) est considéré comme contenant du changement, lorsqu'au moins une des deux paires précédentes a une valeur de critère supérieure au seuil. Dans ce cas, on décide de conserver l'image K_{i+1} et on passe à l'étude du triplet suivant $(K_{i+1}, K_{i+2}, K_{i+3})$. Dans l'hypothèse d'une suppression de l'image K_{i+1}, on étudie ensuite le triplet (K_i, K_{i+2}, K_{i+3}).

Dans le cas où la prise de vue ne contient que deux images clés K_i et K_{i+1}, dès le départ ou bien après suppression d'images redondantes, le critère de dissemblance est calculé entre ces deux images uniquement. Pour une valeur supérieure au seuil de changement, K_{i+1} est supprimée.

Cette procédure est itérée jusqu'à avoir traité chronologiquement tous les triplets d'images clés. Une telle façon de procéder privilégie dans une paire donnée l'image de plus faible indice, i.e. apparue la première. On conserve notamment toujours ainsi la première des images clés.

Pour ce qui est de notre objectif de rapidité et simplicité de l'outil mis en œuvre, $2(k-2)$ calculs de critère de dissemblance sont nécessaires, dans le cas extrême où toutes les images clés sauf la première doivent être supprimées. L'algorithme est donc au pire en $O((k-2)n \times N_p)$, avec n le nombre de pixels dans une image et N_p le nombre de prises de vue de la séquence. Notons toutefois que la majorité des prises de vue ne contient que deux images clés, la première et la dernière, aucun pic secondaire n'ayant été détecté. Dans ce cas, un seul calcul de critère supplémentaire est nécessaire.

Il reste à choisir le seuil de changement à une valeur légèrement plus haute que le seuil global. De la valeur du seuil choisie dépendent les taux de précision et de rappel obtenus pour chaque prise de vue. Plus la valeur du seuil est haute, plus on supprime d'images clés, avec pour conséquence l'augmentation de la précision, au risque de voir le rappel diminuer, lorsqu'on en supprime trop.

En pratique, pour l'ensemble des séquences de la base de données, la même valeur de seuil de changement, égale à deux fois le seuil global, a été conservée. Pour cette valeur, on atteint une valeur moyenne de 1.4% d'images clés par séquence, pour une précision moyenne de 91.7%. Ces valeurs moyennes sont en outre bien représentatives de l'ensemble des données puisque les écarts-type du taux d'images clés et de la précision sont respectivement de 0.5% et 10%.

Cependant pour cette valeur de seuil, on note déjà une baisse légère du rappel, lorsqu'on étudie de façon qualitative les images clés sélectionnées avant et après étude du changement.

Nous donnons à titre d'illustration le résultat de la détection de changement et de la suppression des images clés pour la séquence *travaux* dans le tableau 4.12 et les figures 4.13 (avant changement) et 4.14 (après changement). Cette séquence est un exemple idéal où toutes les images clés supprimées correspondent à une information redondante et où aucune perte d'images pertinentes n'a lieu.

Dans le cadre d'une poursuite de ces travaux d'indexation, une étude plus approfondie du choix de la valeur de seuil de changement serait la bienvenue. Nous nous sommes restreints ici à une valeur fixe, donnant la meilleure précision possible, sans pour autant provoquer une baisse significative du rappel.

Une amélioration possible, consistant à prendre un seuil, non pas fixe quelle que soit la paire d'images clés comparées, mais proportionnel à la durée séparant ces deux images, serait par exemple envisageable. En effet, plus deux images clés sont loin l'une de l'autre dans la prise de vue, plus le changement autorisé entre ces deux prises de vue peut être important.

Après l'exposé de cette piste possible d'amélioration de notre outil, visant à la suppression de son unique paramètre, nous terminons cette étude du changement interne par l'évocation d'une autre application : la validation de la structure linéaire déjà établie.

Nous sommes à présent en mesure de donner le qualificatif de "sans changement" (une seule image clé) ou au contraire "avec changement", à chaque prise de vue d'un document vidéo. Cette information doit alors coïncider avec l'étiquetage effectué pour chaque prise de vue. L'étude de configurations telles que "prise de vue de type classique sans changement" ou "prise de vue de type fondu avec changement" permet en effet de valider et invalider la structure linéaire établie. Nous ne nous attardons pas plus dans ce paragraphe sur le processus de validation de la structure de document vidéo, puiqu'il fait l'objet de la deuxième application du chapitre 8.

Ceci clôt l'exposé des outils développés dans le cadre de nos travaux pour la réalisation du micro-découpage temporel d'un document vidéo.

FIG. 4.13: Images clés extraites pour la séquence *travaux*, avant suppression des redondances.

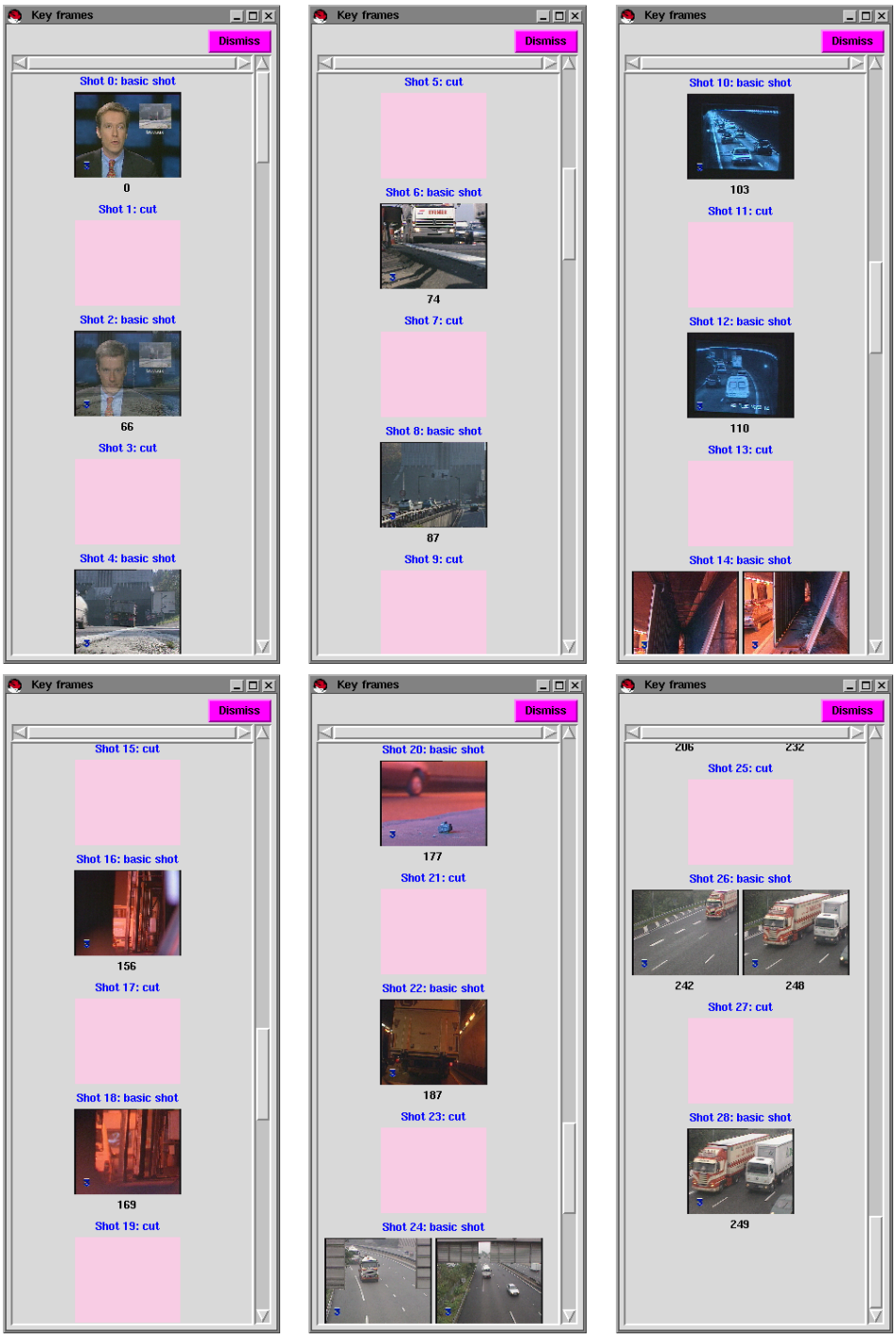


FIG. 4.14: Images clés extraites pour la séquence *travaux*, après suppression des redondances.

4.5 Conclusion

Au cours de ce chapitre consacré à la partie micro-découpage de la structuration linéaire temporelle d'un document vidéo, deux niveaux hiérarchiques supplémentaires ont été atteints : le morceau de prise de vue et l'image clé.

Parmi les découpages possibles d'une prise de vue, nous nous sommes appliqués à détecter les morceaux de type mouvement de caméra et flash. Dans le premier cas, l'outil de détection mis en œuvre n'est autre que l'opérateur de détection de fondu du chapitre 3, puisque nous avons établi que le fondu comme les mouvements de caméra de type rotation ou translation possèdent des caractéristiques similaires, pour le critère de dissemblance choisi. Si la classification obtenue n'est, et ne sera jamais totale (aucune information sur la direction de translation ou sur la distinction rotation-translation n'est disponible), elle a cependant de nombreux atouts parmi lesquels son coût nul, sa rapidité et sa simplicité en comparaison de techniques mettant en œuvre une estimation du mouvement dominant dans la scène. Elle est en outre un excellent point de départ pour le lancement, localement sur les zones étiquetées mouvement de caméra, d'algorithmes plus sophistiqués et qui se trouvent du coup simplifiés, dans la mesure où on connaît déjà une partie de la réponse concernant la nature du mouvement recherché. Il reste à étendre cette classification à d'autres effets tels que les zooms. Là encore, nous avons montré que cette extension était rapidement réalisable, même si les tests n'ont pas été effectués en pratique.

La deuxième étiquette que nous sommes à présent également en mesure d'attribuer à un morceau de prise de vue est le flash. Cet étiquetage résulte là encore de l'application d'un algorithme rapide, basé sur l'établissement de relations entre prises de vue et sur une augmentation locale de la luminance.

Suite à l'extraction de morceaux de prises de vue, le second niveau du micro-découpage, formé par les images clés, a été réalisé grâce à l'extraction, toujours par filtrage morphologique, de la deuxième hiérarchie de pics dans la courbe d'évolution du critère de détection de coupures cette fois-ci. Cette deuxième hiérarchie fournit, encore une fois à un coût minimal, un ensemble d'images représentatives du contenu d'une prise de vue, constituant un résumé drastique du contenu sémantique, puisque seules 2.1% des images du document sont sélectionnées, avec peu de perte d'informations pertinentes : on atteint des taux de rappel de l'ordre de 90%. Cependant la précision est faible (66%) : trop d'images sélectionnées apportent une information redondante.

Dans le but d'éliminer cette redondance, un outil de détection de changements internes à une prise de vue donnée a été mis en place. Cet outil, reposant sur une comparaison de triplets d'images clés consécutives, a prouvé son efficacité par la baisse du taux d'images clés sélectionnées (1.4%) et une augmentation importante de la précision qui atteint, après détection de changement, 91.7%, au prix de l'observation empirique d'une baisse légère du rappel.

Avec l'extraction des images clés, une nouvelle étape est franchie dans l'organisation structurelle d'un document vidéo, puisque le découpage cesse sur le plan temporel, pour continuer cette fois-ci spatialement dans chaque image, avec un partitionnement en régions en objets d'intérêt.

Ce dernier niveau hiérarchique de la structure d'un document constitue le noyau de la partie suivante intitulée : *Structuration linéaire spatiale*.

Deuxième partie

Structuration linéaire spatiale

Chapitre 5

Segmentation : de l'image clé aux régions

5.1 Introduction

Après la réalisation du découpage temporel du fichier vidéo, aboutissant à l'extraction d'images clés ou de mosaïques, il convient de poursuivre par un découpage cette fois-ci spatial de ces images en différents objets d'intérêt. Ce chapitre constitue donc la première partie de ce que nous avons appelé *Structuration linéaire* ***spatiale*** d'un document vidéo.

5.1.1 Avant-propos

Les images clés et mosaïques, extraites ou construites en fin de découpage temporel, ont tout d'abord pour but direct de fournir un résumé représentatif et visuel d'une entité à un utilisateur, dans le cadre d'une recherche dans une base de séquences d'images. Dans cette optique, une fois la tâche difficile consistant à sélectionner ces images clés réalisée, l'objectif final est atteint.

Une deuxième utilisation se dégage toutefois, pour laquelle les images clés ou mosaïques ne sont pas l'étape ultime du processus de structuration, mais une étape intermédiaire dans l'extraction d'informations. La structuration spatiale de ces images permet en effet à la fois de confirmer ou d'infirmer ce que l'on sait déjà sur la structuration temporelle du document vidéo, et de pousser encore plus loin le processus de structuration, par l'extraction et l'étude d'objets particuliers.

Tout comme nous l'avions noté lors de la structuration temporelle, la structuration spatiale prend deux statuts différents suivant que son résultat est un véritable partitionnement de chaque image (i.e. l'union des éléments de la partition reconstitue l'image initiale et deux éléments distincts n'ont aucune intersection), ou suivant qu'on réalise uniquement l'extraction de quelques objets particuliers (texte, incrustation, visages, etc.).

Cette deuxième possibilité sera exposée dans le chapitre 6. Nous détaillons ici la structuration spatiale mise en place en terme de partitionnement, qui n'est autre qu'une segmentation spatiale classique d'images en régions. Une telle segmentation contient l'information supplémentaire non négligeable, d'organisation spatiale des diverses régions entre elles.

Les objets particuliers extraits au chapitre suivant, tout comme les régions de notre segmentation, sont autant d'entités supplémentaires, qui, une fois identifiées et classifiées, serviront

de base à l'établissement de relations de tous ordres, syntaxiques et sémantiques.

Bien sûr, il est à présent couramment admis qu'il n'existe pas une seule segmentation spatiale d'une image a priori, mais que plusieurs sont possibles suivant entre autres l'utilisation que l'on désire en faire. Cependant la classification des régions extraites repose sur l'analyse de primitives de couleur, texture, forme, etc. Elle nécessite donc une segmentation spatiale en régions homogènes, suivant les critères de ces primitives.

Nous décrivons plus en détail dans le paragraphe suivant les qualités que devra posséder la segmentation ou plutôt le niveau de segmentation auquel nous désirons aboutir. Mais notons dès à présent que les régions extraites au final devront correspondre le plus possible à la notion d'objet visuel, telle qu'elle est définie dans la norme MPEG4. Ces objets sont en effet décrits comme des "unités atomiques du contenu d'une image ou d'une vidéo". En ce sens une personne est un objet visuel, tout comme la région correspondant au ciel dans une image. De par la notion d'atomicité contenue dans un objet visuel, chaque région doit donc former un tout et ne pas regrouper deux parties d'objets visuels distincts.

Les motivations dirigeant l'obtention d'une telle segmentation spatiale d'une image clé ou mosaïque étant posées, nous détaillons dans la section suivante les atouts et nouveautés de la technique de segmentation mise en œuvre dans ce mémoire.

5.1.2 Atouts de la segmentation proposée

Les qualités nouvelles que possède la segmentation spatiale exposée ici sont au nombre de trois :

- Tout d'abord, on vise par cet outil, l'obtention de grandes régions homogènes (en couleur), sans se préoccuper des petits détails fins qu'on ne cherche nullement à conserver dans la segmentation finale. Dans un premier temps, ces détails ne nous semblent pas, en effet, être essentiels à une première indexation.
- La qualité suivante de la segmentation mise en œuvre est sa généralité. Une des difficultés du problème de l'indexation d'une base de données quelconques provient en effet du fait que les outils élaborés doivent s'appliquer à des sources a priori très diverses, sans aucune connaissance sur leur contenu. Il est donc absolument nécessaire que la segmentation soit performante quelle que soit l'image de départ. L'utilisation de marqueurs des objets d'intérêt, comme aide à la segmentation, est donc à proscrire.
- Enfin, la troisième nouveauté consiste en la séparation, dès le départ, des pixels de l'image en deux catégories : les pixels possédant une couleur propre et les pixels possédant plutôt un niveau de gris. Pour cela nous proposons une transformation étudiée dans la section 5.3 qui réalise une telle classification. Une fois cette étape achevée, les traitements suivants, proches du processus classique de segmentation en morphologie mathématique, possèdent la caractéristique innovante d'être appliqués séparemment sur les deux classes de pixels, comme cela sera exposé dans la section 5.4.

Les trois atouts principaux de l'algorithme de segmentation construit dans le cadre de ce travail étant introduits, nous proposons de continuer par une présentation des réalisations existantes en segmentation couleur, sous la forme d'un rapide état de l'art, et des outils morphologiques de segmentation, tels que la ligne de partage des eaux (LPE) [12, 17], ou la segmentation hiérarchique [12], qui nous seront utiles par la suite.

5.2 Segmentation couleur et outils morphologiques

5.2.1 Etat de l'art en segmentation couleur

L'état de l'art réalisé ici sera bref dans la mesure où, si la segmentation d'images à niveaux de gris apparaît à l'heure actuelle comme relativement bien maîtrisée, grâce aux outils de morphologie mathématique notamment [17, 73], il en est tout autrement pour les images couleur.

Travailler en vectoriel (i.e. sur les trois canaux couleur) pose en effet de multiples problèmes parmi lesquels la taille des données, le choix de l'espace de couleur et l'apparition de couleurs parasites lors des traitements, i.e. de couleurs nouvelles, non présentes dans l'image originale, et surtout très différentes des couleurs des pixels voisins.

Ces fausses couleurs apparaissent lorsqu'un traitement est appliqué distinctement sur chacun des trois canaux couleur, l'image couleur finale étant alors obtenue par recombinaison des trois composantes traitées. Un tel effet parasite est illustré dans la figure 5.1, dans le cas simple d'une dilatation par un élément structurant carré de taille 1, appliquée successivement sur les trois canaux R, V, et B. La dilatation étant à la base d'un grand nombre d'opérateurs de morphologie mathématique, le phénomène d'apparition de couleurs est donc fréquent pour l'ensemble de ces outils, et tout particulièrement pour les filtres. Pour limiter l'apparition de ces fausses couleur, la solution consiste bien sûr à travailler véritablement en vectoriel et à définir des traitements de l'espace de couleur vectoriel dans lui-même.

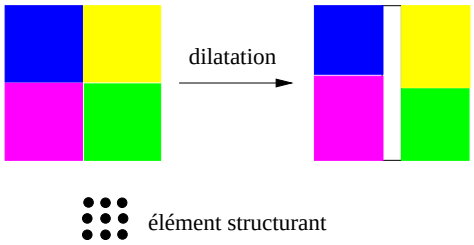


FIG. 5.1: Illustration de l'apparition de fausses couleurs dans le cas d'une dilatation appliquée séparément à chacun des trois canaux couleurs RVB, suivie d'une recombinaison des trois images résultantes pour former une seule image finale.

Se pose alors le problème d'absence d'ordre total dans un tel espace. Outre cet inconvénient majeur, le choix de l'espace de couleur de représentation et de traitement des images reste délicat. Un nombre conséquent d'espaces couleur ont en effet été définis et testés depuis plusieurs dizaines d'années [66, 88, 64], sans qu'aucun n'émerge comme le meilleur espace de représentation.

On cite ainsi, parmi les plus connus, les espaces RVB, XYZ, HLS, HSV, YUV, Lab, Luv, etc. Trois grandes caractéristiques permettent de classifier ces différents espaces :

- ils sont obtenus par transformation linéaire (RVB, XYZ, YUV), ou au contraire non linéaire (HLS, HSV, Lab, Luv) de l'espace RVB ;
- ils correspondent à la perception psycho-physique des couleurs, qu'ils décrivent en fonction de trois paramètres : teinte, saturation et luminance (HLS, HSV) ;
- ils sont perceptuellement uniformes en chrominance et/ou en luminance, i.e. les distances numériques que l'on peut calculer entre deux couleurs correspondent aux différences

psycho-visuelles perçues (YUV, Lab, Luv).

Tous espaces confondus, les techniques de segmentation couleur se classent en quatre grandes catégories.

Le premier groupe de techniques, et également le plus ancien, base la segmentation sur des seuillages d'histogrammes, avec l'inconvénient évident de la difficulté de déterminer les valeurs de seuil.

Puis viennent les méthodes reposant sur une classification (*clustering*) des pixels en fonction d'un critère de similarité de couleur. De tels algorithmes sont en général gourmands en ressources machine. Les solutions visant à remédier à cet inconvénient sont doubles : soit développer des méthodes efficaces de stockage et d'accès aux données tri-dimensionnelles, i.e. les couleurs, soit projeter ces données dans un espace de dimension plus petite. Un exemple d'un tel algorithme est proposé par Celenk [22, 23], qui choisit de travailler dans l'espace Lab exprimé en coordonnées cylindriques, notées $L^*H^\circ C^*$, de façon à se rapprocher au maximum de la perception humaine.

Face à ces techniques de classification qui ne prennent pas en compte les relations spatiales des pixels, mais uniquement des critères de similarité de couleur, plusieurs méthodes de croissance et fusion de régions ont été développées, dont celle de Tremeau [97]. L'innovation de cette dernière technique repose sur l'utilisation conjointe de critères de similarité locaux, régionaux et globaux. Les régions ainsi obtenues peuvent être spatialement déconnectées, dans la mesure où elles restent homogènes en couleur. L'espace choisi est RVB mais l'algorithme peut être adapté sans difficulté à d'autres espaces tels que Lab ou Luv.

A cheval sur ces deux catégories de techniques se trouvent des méthodes utilisant à la fois, ou plutôt successivement, une classification et une fusion de régions (cf. Schettini [81]).

Enfin le dernier groupe d'algorithmes de segmentation couleur repose sur la détection de contours dans les images, et parmi ceux-ci les algorithmes [79] basés sur une segmentation morphologique de type LPE. De par le processus mis en œuvre, ces algorithmes se positionnent en outre parfois plus dans le groupe de méthodes basées sur des fusions de régions, que dans ce dernier groupe. Une telle segmentation reposant presque toujours sur le calcul du gradient de l'image originale, quatre façons de procéder sont en effet à distinguer :

- Transformation de l'image couleur en une image à niveaux de gris (par maximum des composantes par exemple), puis calcul du gradient classique sur cette nouvelle image et ligne de partage des eaux.
- Calcul des trois gradients classiques sur chacune des composantes de l'image couleur, puis recombinaison de ces trois gradients en un gradient résultant (norme par exemple) et ligne de partage des eaux [34]. Une variante consiste à calculer directement un gradient vectoriel couleur sur lequel on construit la LPE. Sharafenko *et al.* [87] proposent ainsi la définition d'un gradient Luv.
- Calcul des trois gradients sur chacune des composantes, suivi de trois LPE et recombinaison de ces trois LPE pour former une LPE résultante [73].
- Calcul de la LPE directement sur l'image couleur sans passer par l'établissement d'un gradient [70]. Il ne s'agit plus alors véritablement d'une ligne de partage des eaux, mais plutôt d'un algorithme de fusion de régions à partir de marqueurs.

La description de ce dernier groupe d'algorithmes utilisant des outils morphologiques conclut le tour d'horizon des techniques de segmentation couleur existantes. Les segmentations par LPE étant naturellement dans la ligne morphologique de nos travaux, nous les avons

sélectionnées par la suite. Elles ont par ailleurs prouvé, à de nombreuses reprises, la validité de leurs résultats, de par la qualité des régions extraites et le positionnement des contours fermés obtenus.

Nous continuons donc à présent par un exposé plus poussé des outils de segmentation morphologique proprement dits.

5.2.2 Segmentation morphologique

Cette section s'organise en quatre points principaux. Nous commençons par donner une définition de l'opérateur de base qu'est la ligne de partage des eaux ; puis nous explicitons la façon dont cet outil s'insère dans un processus complet de segmentation. Cette étude débouche naturellement sur l'inconvénient majeur de l'opérateur de LPE : la sursegmentation, à laquelle on remédie généralement grâce à l'utilisation de marqueurs. Le choix de tels marqueurs n'étant pas toujours possible, nous terminons par l'exposé de l'outil de segmentation hiérarchique, qui s'en affranchit.

5.2.2.1 Ligne de partage des eaux

La ligne de partage des eaux (LPE) est l'opérateur morphologique sur lequel repose tout le processus de segmentation [12, 17].

Si on considère une image numérique à niveaux de gris comme un relief topographique (cf. figure 5.2, (a)), cette transformation consiste à simuler l'inondation de ce relief à partir des minima régionaux de l'image. Une description imagée du processus est la suivante. Si on imagine que les minima de l'image sont troués, l'opération revient à plonger le relief dans un lac. Au fur et à mesure que le relief s'enfonce, l'eau rentre d'abord par le minimum le plus bas, puis successivement par les minima d'altitudes plus élevées, et remplit progressivement les bassins versants correspondant à chaque minimum.

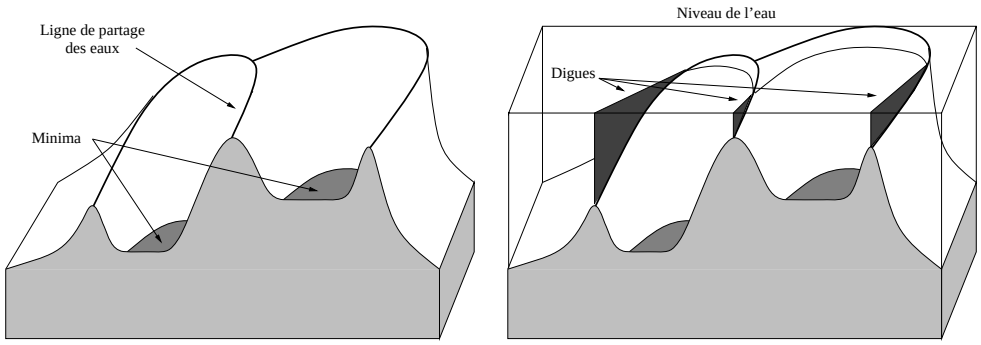


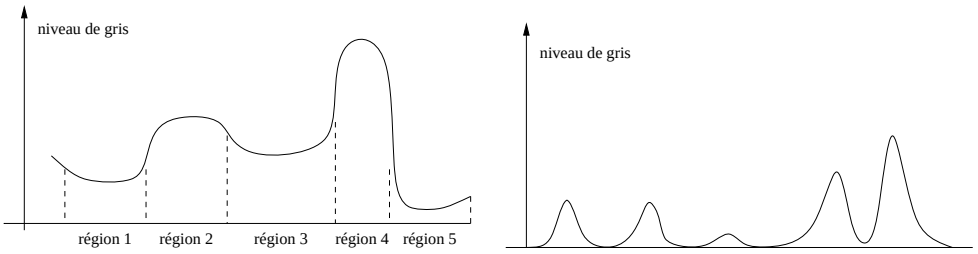
FIG. 5.2: Construction de la ligne de partage des eaux d'une image par un processus d'inondation.

A tout instant, le niveau d'eau est le même dans tous les bassins versants. Lorsqu'un col entre deux bassins versants est atteint, des eaux provenant de deux minima différents se rencontrent. On érige alors des digues pour empêcher ces "mélanges" (cf. figure 5.2, (b)). L'ensemble des ces digues, placées sur les lignes de crête du relief, constitue la ligne de partage des eaux de l'image. Elles délimitent autant de bassins versants qu'il y avait de minima présents dans l'image de départ.

5.2.2.2 Application à la segmentation

Si les lignes de crête de l'image à segmenter correspondaient aux contours des régions que l'on désire extraire, l'application directe de la LPE sur l'image originale fournirait le but recherché, i.e. les bassins versants obtenus correspondraient aux objets dans l'image.

Dans le cas de régions homogènes en niveaux de gris, les lignes de crête ne correspondent malheureusement pas à leurs frontières, comme l'illustre l'exemple monodimensionnel de la figure 5.3, (a).



(a) image originale (b) gradient

FIG. 5.3: Correspondance existant entre les lignes de crête du gradient et les régions de l'image originale.

Par contre l'image gradient construite à partir de cette image originale possède cette propriété. Le gradient de luminance est en effet maximal sur les contours et présente au contraire un niveau relativement uniforme à l'intérieur des objets (cf. figure 5.3, (b)). Le processus de segmentation morphologique consiste donc à appliquer la LPE sur l'image gradient, dérivée de l'image originale. L'ensemble des contours obtenus présente la propriété d'être fermés, délimitant ainsi sans ambiguité les régions, correspondant aux objets de l'image de départ.

Un tel procédé conduit en pratique à des images très sursegmentées, du fait du bruit de construction du gradient, responsable de la présence d'un grand nombre de minima dans l'image, et donc d'un grand nombre de bassins versants. Une solution consistant en l'utilisation de marqueurs, choisis de façon à correspondre aux régions à segmenter, est détaillée dans le paragraphe suivant.

5.2.2.3 Sursegmentation et marqueurs

Ainsi que nous venons de l'exprimer, la forte sursegmentation des images provient de la présence de multiples minima, dus au bruit. Pour que chaque région soit extraite comme un seul bassin versant, il est nécessaire de supprimer tous ces minima parasites et de les remplacer par un seul marqueur. L'inondation débute alors non par les minima de l'image, mais par l'ensemble des marqueurs choisis, un par région à segmenter. Ce nouvel algorithme porte le qualificatif de **LPE contrôlée par marqueurs**.

Une fois le choix de marqueurs effectué, il reste également à choisir sur quelle fonction de l'image originale appliquer la LPE. Jusqu'à présent, cette fonction correspondait au gradient de l'image à segmenter. Là encore, en pratique, du fait du caractère bruité du gradient, les lignes de crête obtenues ne correspondent pas parfaitement aux frontières des objets. Le choix d'autres transformations de l'image de départ, le plus souvent à base de filtrage et de calcul de gradient, peut donc conduire à une segmentation de meilleure qualité.

En résumé le processus de segmentation se décompose en deux étapes : une première étape "intelligente" de choix des marqueurs et de la transformation de l'image originale sur laquelle appliquer la LPE, et une deuxième étape de construction mécanique de cette LPE.

En fonction des images à segmenter, le choix des marqueurs n'est pas toujours possible. Il est alors nécessaire d'utiliser un autre outil morphologique, la segmentation hiérarchique, détaillée dans le paragraphe suivant.

5.2.2.4 Segmentation hiérarchique

Une première raison de la difficulté d'extraction de marqueurs des objets à segmenter est tout simplement l'absence d'informations sur le contenu de l'image, comme c'est notamment bien souvent le cas dans le processus le plus général d'indexation d'une base de données quelconque.

Il est également fréquent de rencontrer des images pour lesquelles les objets à segmenter sont complexes. Souvent formés de plusieurs morceaux de niveaux de gris ou couleurs différents et de géométries elles-aussi complexes, il n'est pas toujours aisé de leur associer des caractéristiques simples et uniques.

A la difficulté du choix des marqueurs, s'ajoute celle du choix de la transformation de l'image originale sur laquelle appliquer la LPE. Dans les deux cas, la raison de ces difficultés provient de la trop grande complexité de l'image à segmenter. Partant de cette constatation, la déduction immédiate est qu'une simplification de l'image originale s'impose, sans pour autant qu'il y ait perte de l'information pertinente pour la segmentation.

Cette procédure conduit alors à une approche hiérarchique de la segmentation.

La première LPE construite à partir de l'image originale fournit en effet une première hiérarchie de segmentation constituée d'une multitude de petites régions. Suite à une simplification de l'image originale et à la construction d'une nouvelle LPE, on bâtit une deuxième hiérarchie de segmentation, constituée de régions plus importantes, englobant les petites régions détectées par la première LPE.

Sur l'image originale avant toute simplification, un observateur humain est, malgré le niveau de bruit et la complexité, à même de distinguer des plages de gris ou de couleurs plus ou moins uniformes. Ceci indique que les gradients de luminance ou de couleur sont de valeurs plus élevées sur ces contours que sur tous les contours surnuméraires, introduits par la sursegmentation à l'intérieur des régions. Une illustration de cet état de fait est proposée dans la figure 5.4.

Dans le processus de segmentation hiérarchique tel que nous le présentons ici et que nous utiliserons comme étape finale de notre algorithme de segmentation, la simplification intervient justement au niveau de ces contours surnuméraires correspondant à des valeurs plus faibles de gradient, entourés de morceaux de contours de gradients plus élevés (cf. figure 5.4). Cette simplification repose donc sur l'hypothèse que les contours les plus forts correspondent aux objets à segmenter. Une telle hypothèse, si elle est vérifiée dans la majorité des cas, peut toutefois parfois être mise en défaut, et mener à la reconsidération du critère de faible gradient comme choix de suppression de certains contours, comme nous le verrons en conclusion de notre technique de segmentation.

Concrètement à chaque étape de cette segmentation hiérarchique, on cherche donc à supprimer les contours les plus faibles entourés de contours de valeurs plus élevées. Cette élimination est réalisée par le passage de l'image des régions correspondant à la première LPE sursegmentée, à un graphe dans lequel chaque nœud est un bassin versant et chaque arête prend la valeur

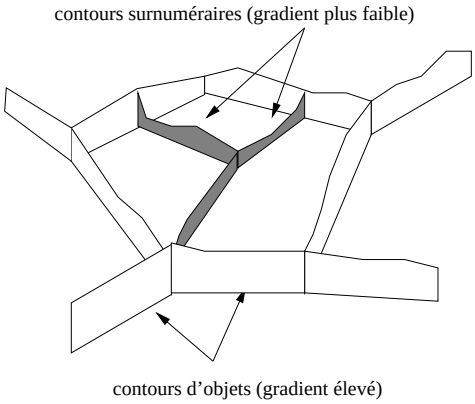


FIG. 5.4: Différence de hauteurs de gradients.

minimale du contour entre deux bassins versants voisins. On affecte ensuite à chaque nœud la valeur minimale de toutes les arêtes qui en sont issues. La détection et la suppression de "contours minimaux" sont alors réalisées automatiquement par la construction d'une nouvelle LPE sur ce graphe modifié, à partir des nouveaux minima locaux créés, correspondant aux zones de faibles contours.

Un tel processus peut bien sûr être itéré de façon à supprimer, dans une nouvelle passe, l'ensemble des contours devenus à leur tour les plus faibles. L'itération d'une telle segmentation hiérarchique se terminant trivialement par l'obtention d'une unique région correspondant à l'image toute entière, il est primordial de bâtir des critères d'arrêt du processus, répondant à l'apparition d'objets dont les caractéristiques de forme, texture, couleur, etc. correspondent à celles des objets recherchés.

Ces caractéristiques n'étant de façon évidente pas toujours connues a priori, d'autres critères tels que le nombre final de régions dans la segmentation, dont on laisse au lecteur le soin d'apprécier la faible adéquation avec le contenu d'une image quelconque, peuvent également être utilisés. Le choix du critère d'arrêt reste cependant un problème posé. Nous nous proposons incidemment d'y répondre ou du moins d'y apporter une solution grâce à l'ensemble de notre méthodologie de segmentation.

5.2.3 Conclusion

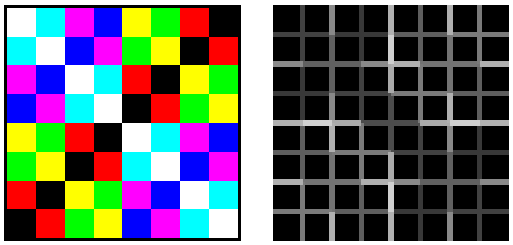
Un des objectifs que nous nous sommes fixés au début de ce mémoire est d'élaborer des outils utiles à la structuration et à l'indexation de documents vidéo de tous types. Cette contrainte, ajoutée à la façon dont les images clés sont extraites au chapitre 4, font que le contenu de ces dernières est bien évidemment lui-aussi quelconque, et par là inconnu. L'outil de segmentation hiérarchique, basé sur des LPE successives d'une image de plus en plus simplifiée, que nous venons de présenter, s'impose donc.

Mais avant de l'appliquer, il convient de déterminer la transformation de l'image originale qu'il faut réaliser de façon à construire les LPE successives. Cette transformation est en fait constituée d'une succession d'étapes, dont la première - et la plus innovante - est ce que nous appelons la **transformation HSV améliorée**. Cette nouvelle transformation fait l'objet du paragraphe suivant.

5.3 Transformation HSV améliorée

5.3.1 Motivation

L'élaboration de la transformation HSV améliorée est motivée par deux constatations. La première vient de l'étude du comportement des outils de segmentation couleur pour les régions d'une image, ayant peu, voire pas du tout, de couleurs. Pour ces régions plutôt "grises", le calcul d'un gradient couleur (par exemple la distance euclidienne dans l'espace *Lab*) ne fournit pas de valeurs aussi élevées que celles obtenues pour les contours des régions plus colorées (cf. figures 5.5, (a) et (b)). Cette impression de différence d'échelle dans les valeurs du gradient pour les zones grises et les zones colorées est par ailleurs confortée par le simple calcul de la valeur du gradient dans *Lab* (distance euclidienne dans l'espace *Lab*) entre deux points de couleur, l'un vert et l'autre magenta d'une part, et entre deux points blanc et noir d'autre part. On aboutit respectivement à des valeurs de gradient de 272.7 et 100.4, ce qui ne correspond pas, à notre avis, à la perception visuelle de différence entre ces deux paires de points. Face à cette différence de résultat sur les zones grises, une solution consiste à les extraire et à les traiter séparemment (par des opérateurs niveaux de gris) des zones de couleur.



(a) image originale (b) gradient Lab

FIG. 5.5: Différence d'échelle des valeurs de gradient Lab (b), obtenues pour les zones couleur et les zones niveaux de gris d'une même image originale (a).

De plus, il est nécessaire, avant l'étape finale de segmentation par LPE, de simplifier le plus possible l'image originale. Une façon d'améliorer cette image de départ, afin d'en extraire plus aisément les grandes plages homogènes en couleur, est de "renforcer" l'impression de couleur. Malheureusement, ce terme de renforcer, qui se comprend intuitivement dans le cas de régions colorées par "augmenter l'impression de couleur", perd toute signification pour les parties grises ou presque grises de l'image. Plutôt que de chercher à renforcer alors une impression colorée, qui n'existe peut-être pas dans ce cas, il est plus sûr, au contraire, de chercher à l'éliminer et à ne conserver qu'une information de niveau de gris.

Une fois ces deux motivations exprimées, il est nécessaire de choisir un espace de couleur dans lequel les deux notions de teinte et de luminance sont séparées, au niveau même des composantes. Dans ce choix nous nous sommes également laissés guider par la qualité de certains espaces à être proches de la perception psycho-visuelle. Pour ces deux raisons, nous avons sélectionné l'espace *HSV* sur lequel nous revenons dans le paragraphe suivant. L'étude de cet espace fera alors apparaître une motivation supplémentaire à la construction de notre transformation HSV améliorée.

5.3.2 Espace HSV

5.3.2.1 Définition

L'espace HSV choisi fait partie des espaces possédant une interprétation très physique de ses composantes. Le triplet de variables le plus proche de la perception consciente humaine, i.e. la longueur d'onde, la pureté et la luminance est en effet très similaire au triplet formé par les trois composantes H, la teinte (longueur d'onde), S, la saturation (pureté), et V, la luminance.

Les deux premières coordonnées H et S contiennent toute l'information couleur d'un pixel. Parmi ces deux composantes, seule la teinte donne véritablement une indication du domaine de couleur (un pixel est-il plutôt rouge ou bleu ?). Ainsi, deux pixels de même teinte, mais de saturations différentes resteront quand même dans le même domaine de couleur. La saturation représente en effet l'opposé de la quantité de blanc contenue dans une couleur, un pixel de saturation minimale contenant une grande quantité de blanc (couleur pastel), et au contraire un pixel de saturation maximale n'en contenant pas du tout (couleur vive). Quant la troisième composante V, elle rend compte de la luminance.

Une représentation de cet espace est proposée sous la forme du schéma 5.6. Dans cet espace tridimensionnel, l'axe vertical correspond à la luminance, l'angle dans le plan horizontal correspond à la teinte et le rayon à la saturation de la couleur.

L'espace HSV se définit par un système non linéaire d'équations de passage à partir de l'espace RVB. Pour chaque pixel de composantes (R, V, B), comprises entre 0 et 255, on pose :

$$\begin{aligned} min &= \min(R, V, B) \\ max &= \max(R, V, B) \end{aligned} \tag{5.1}$$

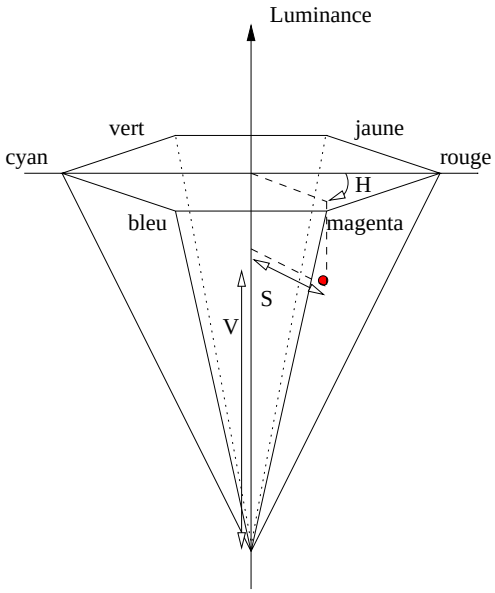


FIG. 5.6: Représentation de l'espace HSV.

Dans le cas particulier où $max = min$ (R, V, B égaux), le triplet (R, V, B) se transforme en :

$$\begin{aligned} H &= 0 \\ S &= 0 \\ V &= max \end{aligned} \tag{5.2}$$

(5.3)

Si, de plus, $V = 0$, on fixe S à zéro également.

La teinte H est en fait indéfinie, dans les deux cas, mais par convention, elle est mise à 0. Dans tous les autres cas, on transforme le triplet (R, V, B) en :

$$V = max \tag{5.4}$$

$$S = \frac{max - min}{max} \tag{5.5}$$

$$H = \begin{cases} \frac{V-B}{max-min} & \text{si} \quad R = max \\ 2 + \frac{B-R}{max-min} & \text{si} \quad V = max \\ 4 + \frac{R-V}{max-min} & \text{si} \quad B = max \end{cases}$$

Il est alors nécessaire de transformer H de la façon suivante :

$$H \longrightarrow H \bmod 6 \tag{5.6}$$

On obtient donc des valeurs de (H, S, V) respectivement dans $[0, 6]$, $[0, 1]$ et $[0, 255]$, avec un espace HSV en forme de cône.

Ces définitions étant établies, le paragraphe suivant est l'occasion de noter un inconvénient de cet espace de représentation, inconvénient qui fournit une motivation supplémentaire à la construction de la transformation HSV améliorée.

5.3.2.2 Fausses couleurs

Une des motivations citées au paragraphe 5.3.1 provenait de la nécessité de renforcer l'impression de couleur des régions colorées d'une image. En se basant sur la définition de la saturation comme opposé de la quantité de blanc contenue dans une couleur, un moyen intuitif d'arriver à ce but est de chercher à augmenter la saturation d'une couleur donnée. Sous l'effet d'une telle opération, une couleur pastel se retrouve être une couleur vive : l'impression de couleur est donc bien renforcée.

La transformation qui consiste à augmenter artificiellement la saturation de chaque pixel d'une image tout en conservant sa teinte peut s'écrire :

$$\begin{aligned} \text{Transformation HSV : HSV} &\longrightarrow \text{HSV} \\ (h, s, v) &\longrightarrow (h', s', v') = (h, 1, v') \end{aligned} \tag{5.7}$$

Nous y ferons référence par la suite comme à la **transformation HSV**. Du fait de la représentation de l'espace HSV, cette transformation, dont on fournit un exemple dans la figure 5.7, présente l'inconvénient de donner des teintes plus ou moins arbitraires pour les pixels situés près de l'axe de luminance. Leur valeur de saturation étant faible, de petites variations de l'angle se traduisent par des variations critiques de teinte.

(a) image originale (b) tranformée HSV

FIG. 5.7: Exemple de transformation HSV appliquée (b) à une image (a).

Ceci vient conforter l'idée déjà évoquée à la section 5.3.1 : un pixel près de l'axe vertical n'ayant pas véritablement de couleur significative, chercher à la renforcer ne peut mener à une transformation satisfaisante.

C'est pour cette raison que, dans la transformation HSV de la figure 5.7, la route dans la petite image incrustée, ou le texte toujours sous cette incrustation, apparaissent avec des teintes multicolores et manifestement erronées.

Ce comportement de la transformation HSV pour des pixels ayant un niveau de gris plutôt qu'une couleur significative constitue un troisième argument en faveur d'une transformation capable tout d'abord de classifier les pixels d'une image entre ceux possédant un niveau de gris et ceux possédant une couleur réelle, puis de transformer les valeurs de ces pixels de façon à, pour la première classe, éliminer toute information de couleur au profit d'un niveau de gris, et, pour la deuxième classe, renforcer l'impression de couleur.

5.3.3 Description de la transformation HSV améliorée

La **transformation HSV améliorée** - puisque constituant véritablement une amélioration de la transformation HSV précédente (équation 5.7) - s'effectue en deux étapes, l'une de classification des pixels, et l'autre de modification de leur couleur en fonction de leur classe d'appartenance. Ce paragraphe se subdivise donc en deux sous-sections dédiées à la description de ces deux étapes successives.

5.3.3.1 Séparation en deux classes

Une première classification entre les pixels de l'image ayant une couleur significative et ceux n'ayant qu'un niveau de gris, mais bâtie dans l'espace HLS, a déjà été proposée dans nos travaux précédents [34], aboutissant à la transformation **HLS** améliorée. Cette classification est obtenue par le choix manuel d'un seuil s_0 sur la saturation, en deçà duquel un pixel est considéré comme n'ayant pas de couleur. Pour des valeurs de seuil croissantes, on conserve donc de moins en moins de pixels dans la classe couleur, et au contraire, les régions à niveaux de gris sont de plus en plus importantes (cf. figure 5.8).

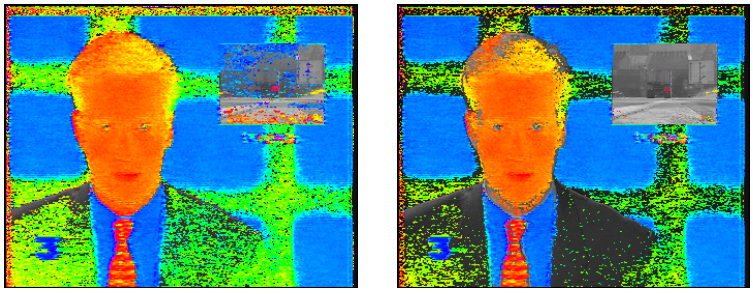


Image transformée ($s_0 = 20$) Image transformée ($s_0 = 40$)

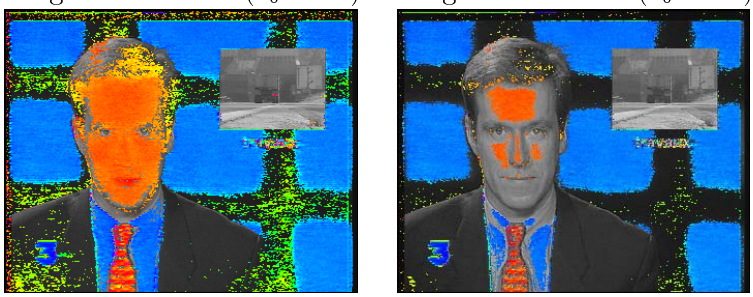


Image transformée ($s_0 = 50$) Image transformée ($s_0 = 80$)

FIG. 5.8: Exemples de transformation HLS améliorée pour des valeurs de s_0 différentes. L'image originale est présentée dans la figure 5.7.

L'espace HLS étant représenté par la superposition d'un deuxième cône inversé, sur un premier cône positionné de façon similaire à celui de l'espace HSV, le seuillage de la saturation S conduit alors à la sélection d'un cylindre autour de l'axe de luminance, dans lequel les pixels sont considérés comme "gris".

La classification que nous proposons ici est plus fine (elle n'aboutit pas à un simple cylindre, mais à une forme plus complexe) que celle construite dans [34], dans la mesure où elle ne repose pas uniquement sur une étude des valeurs de la saturation S, mais sur l'évolution du produit SV.

Des équations 5.4 et 5.5, on déduit une expression intéressante du produit SV :

$$SV = max - min \tag{5.8}$$

Cette expression possède la propriété d'être nulle pour des niveaux de gris ($max = min$), et d'être maximale dans le cas extrême d'une couleur pure, par exemple $R = 255$, $V = 0$ et $B = 0$.

L'équation $SV = cte$ se traduit par un hyperboloïde qui sépare l'espace HSV en deux régions correspondant à la classification recherchée : au-dessus de cette surface, les pixels présentent une couleur significative et en-dessous, ils sont plus proches d'un niveau de gris. Pour une illustration de l'hyperboloïde et de la classification induite, le lecteur se reportera à la figure 5.9.

Bien sûr des valeurs du produit SV différentes produiront des hyperboloïdes plus ou moins proches de l'axe vertical de luminance. Il reste à déterminer quel hyperboloïde choisir, i.e. quelle valeur du produit SV sélectionner pour classifier de la meilleure façon possible les pixels gris et les pixels couleur.

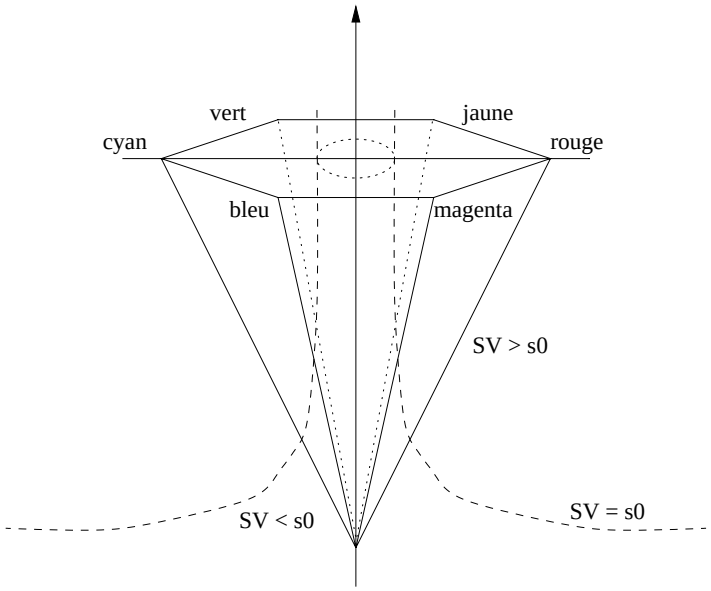
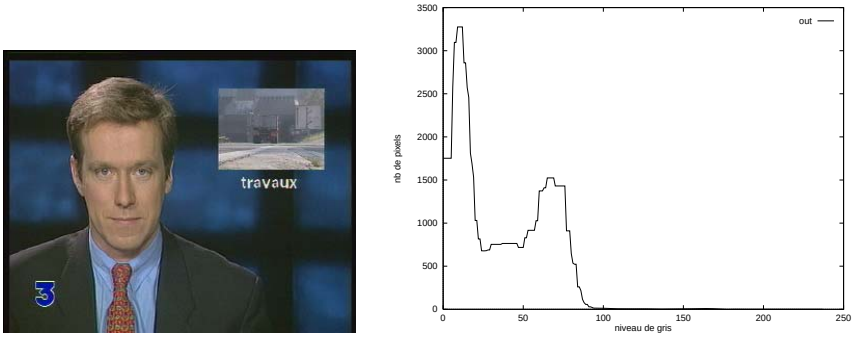


FIG. 5.9: Hyperboloïde $SV = s_0$.

L'étude de l'histogramme de l'image SV est alors très riche en informations. Cet histogramme présente plusieurs modes dont un ou deux principaux dans les petites valeurs de niveaux de gris. Un exemple d'un tel histogramme, ainsi que l'image originale à partir duquel il a été construit sont disponibles figure 5.10. Appelons s_0 la valeur intermédiaire entre ces deux modes principaux.



a) b)

FIG. 5.10: a) Image originale. b) Histogramme des niveaux de gris de l'image SV, construite à partir de l'image originale (seuil $s_0 = 38$).

L'équation $SV = s_0$ correspond à l'hyperboloïde délimitant de façon idéale les pixels ayant une couleur significative ($SV > s_0$) de ceux ayant uniquement un niveau de gris ($SV < s_0$).

En pratique la sélection de la valeur s_0 est faite automatiquement par une technique de classification par K-means sur l'histogramme, avec trois classes recherchées. Les deux premières classes représentent ainsi les deux modes principaux de l'histogramme et la troisième classe, l'ensemble des valeurs résiduelles pour des niveaux de gris élevés. En d'autres termes, la première classe correspond aux pixels de l'image originale ayant seulement un niveau de

gris ; la deuxième classe correspond aux pixels ayant une information colorée ; enfin la dernière classe est constituée des pixels restants (qui n'ont pu être classifiés dans l'une ou l'autre des deux premières classes). Expérimentalement ces pixels semblent correspondre à des couleurs plutôt pastel. Le choix automatique de s_0, comme la limite entre les deux premières classes, est généralement correct pour la plupart des images. Ceci constitue la deuxième amélioration apportée par rapport à la technique présentée dans [34], pour laquelle le choix du seuil s'effectuait manuellement.

5.3.3.2 Renforcement de l'impression de couleur

La séparation en deux classes des pixels de l'image originale étant réalisée, il reste à définir quelle transformation appliquer à chacune d'entre elles. Rappelons que le but recherché est d'augmenter l'impression de couleur pour les pixels possédant déjà une couleur significative, et au contraire d'éliminer toute couleur des pixels plutôt gris.

Lors de la présentation de la transformée HSV, un moyen simple d'augmenter l'impression de couleur a été évoqué : il s'agissait de fixer la saturation à la valeur maximale de 1. Une telle transformation ne respecte cependant pas la géométrie de l'espace HSV en forme de cône à base hexagonale.

Aussi préférons-nous la transformation suivante, dans le cas d'un pixel coloré :

$$\text{Si } sv > s_0, (h, s, v) \longrightarrow (h, S_{max}, v) \tag{5.9}$$

où S_{max} correspond à la saturation du projeté horizontal du point (h, s, v) sur la surface du cône (cf. figure 5.11).

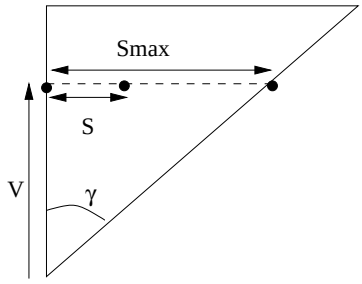


FIG. 5.11: Représentation des deux projetés horizontaux d'un point : l'un sur la surface du cône HSV et l'autre sur l'axe de luminance.

Pour des raisons de simplification des calculs, notons qu'à partir de cet instant nous renormalisons l'intensité v, originellement comprise entre 0 et 255, à des valeurs comprises entre 0 et 1. Dans ce cas, le cône HSV a sa hauteur égale à son rayon maximal, l'angle γ représenté sur la figure 5.11 étant égal à 45°. On arrive alors rapidement à partir de l'expression de la tangente de l'angle γ à l'égalité : $S_{max} = v$.

L'utilisation de la projection sur la surface du cône a en outre l'avantage de se symétriser au cas d'un pixel tel que : $sv < s_0$. La transformation adoptée revient alors en effet à remplacer le point (h, s, v) par son projeté horizontal, cette fois-ci sur l'axe de luminance :

$$\text{Si } sv < s_0, (h, s, v) \longrightarrow (0, 0, v) \tag{5.10}$$

En résumé, on obtient la transformation suivante :

$$\begin{array}{rcl} \text{Transformation HSV Améliorée : HSV} & \longrightarrow & \text{HSV} \\ (h,s,v) & \longrightarrow & (h',s',v') = \left\{ \begin{array}{cc} (h, S_{max}, v) & \text{si } sv \geq s_0 \\ (0,0,v) & \text{si } sv < s_0 \end{array} \right. \end{array} \quad (5.11)$$

où $S_{max} = v$. Notons au passage que cette transformation est un filtre fort commutatif. Pour une démonstration de cette propriété, le lecteur se reportera à l'annexe C.

Des exemples de résultats obtenus par une telle transformation appliquée à des images réelles sont regroupés dans le paragraphe suivant.

5.3.4 Résultats en images

Pour chaque image originale, on fournit la valeur s_0 déterminée automatiquement, l'histogramme de l'image SV et le résultat de la transformation HSV. Les exemples de la figure 5.12 correspondent à des choix automatiques de seuil corrects ; un exemple de mauvaise sélection de s_0 est également proposé dans la figure 5.13. Une analyse approfondie de ce dernier exemple permet d'aboutir à la conclusion que le seuil sélectionné par l'algorithme correspond bien à une séparation entre les deux modes présents dans l'histogramme. Cependant pour cette valeur de seuil, des régions telles que les bâtiments du fond de l'image ou le trottoir sont considérés comme colorés. Si les bâtiments ont effectivement une teinte très légèrement jaune pâle, il n'en est pas de même pour le trottoir qui visuellement (mais est-ce objectif, ou influencé par notre connaissance a priori ?) est gris. Une valeur de seuil plus élevée, ne correspondant pas à la séparation des deux modes, permet cependant une classification globale du trottoir comme région à niveaux de gris. Notons toutefois qu'aucune sélection manuelle du seuil ne permet une séparation parfaite du trottoir et des régions véritablement colorées dans l'image. En outre le choix de l'algorithme de considérer les bâtiments comme des zones colorées, choix qu'un observateur humain peut contester, reste cependant objectivement correct. Il convient donc de moduler l'échec tout relatif de ce dernier exemple.

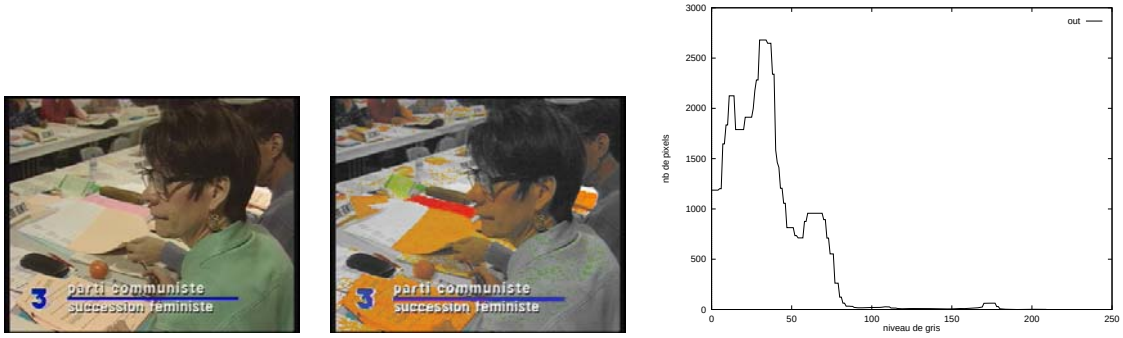
5.3.5 Conclusion

La nouvelle transformation HSV améliorée que nous venons de définir effectue une classification automatique des pixels colorés et des pixels à niveaux de gris d'une image couleur. Une fois cette classification effectuée, elle réalise de plus une modification différente des ces pixels, suivant leur classe d'appartenance : l'impression de couleur est renforcée par le calcul d'une valeur de saturation plus élevée, dans le cas d'un pixel coloré ; au contraire, pour un pixel gris, toute information couleur est éliminée par projection sur l'axe de luminance.

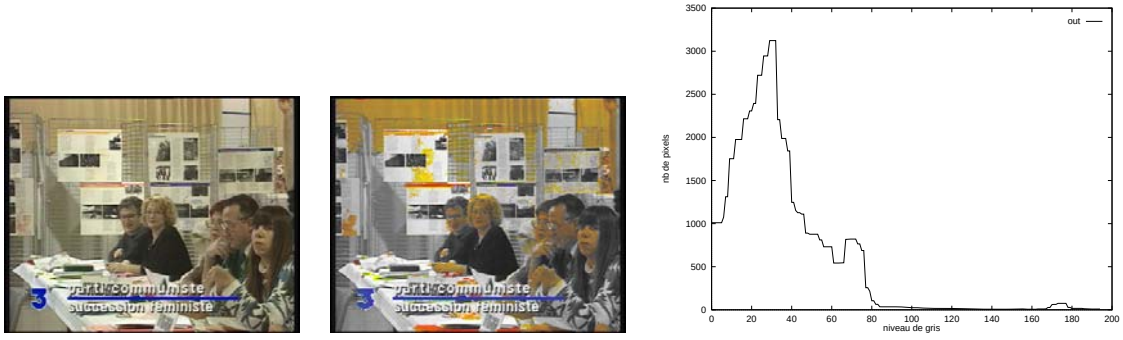
Cette conservation de l'information la plus pertinente (la luminance ou la teinte, suivant la classe) permet une bien meilleure préparation des images couleur, en vue de leur segmentation. Ce prétraitement conserve en effet suffisamment d'information pour espérer obtenir les grandes régions homogènes de l'image, tout en la simplifiant.

Cette simplification intervient en effet dans le passage d'un espace tri-dimensionnel à un espace mono ou bi-dimensionnel suivant les classes. Un pixel donné est en effet représenté uniquement par sa luminance (projection sur l'axe vertical) ou bien par trois composantes (h,s,v) dont deux seulement sont désormais indépendantes (projection sur la surface du cône). Outre la simplification des images, cette réduction du nombre de dimensions constitue de plus

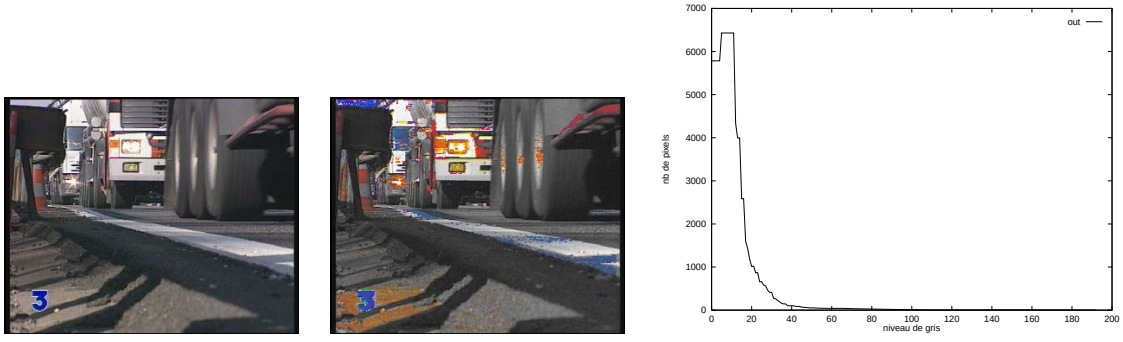
(a) seuil $s_0 = 38$



(b) seuil $s_0 = 40$

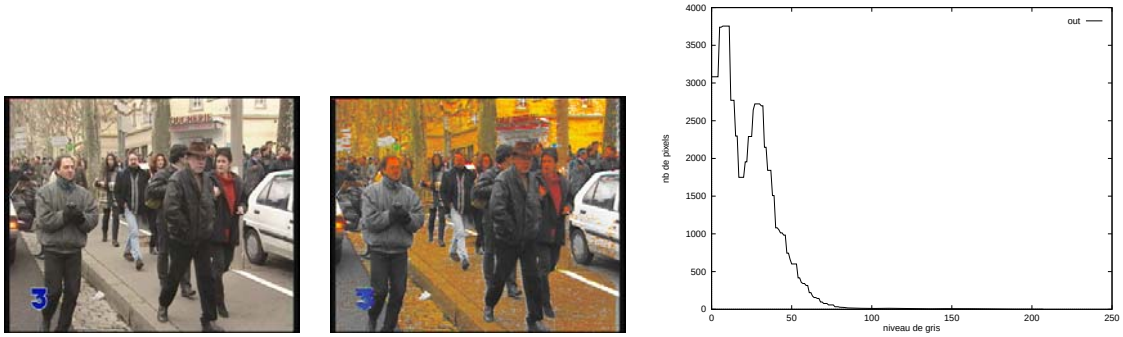


(c) seuil $s_0 = 40$



(d) seuil $s_0 = 17$

FIG. 5.12: Exemples de résultats de la transformation HSV améliorée, pour lesquels le choix automatique du seuil s_0 est correct. On présente à chaque fois l'image originale (gauche), le résultat de la transformée HSV améliorée (milieu) et l'histogramme des niveaux de gris de l'image SV (droite).



(e) seuil $s_0 = 23$

FIG. 5.13: Exemple de résultat de la transformation HSV améliorée, pour lequel le choix automatique du seuil s_0 peut être amélioré (une valeur proche de 35 étant idéale). On présente l'image originale (gauche), le résultat de la transformée HSV améliorée (milieu) et l'histogramme des niveaux de gris de l'image SV (droite).

un élément de réponse, qu'il serait intéressant de pousser plus loin, au problème de la taille des données, lorsqu'on travaille sur des images ou séquences couleur.

Dans l'attente d'une exploitation de cette simplification, les images obtenues par la transformation HSV améliorée constituent la première étape de la segmentation couleur mise en œuvre dans ce mémoire, la classification introduite permettant par la suite l'application de traitements séparés pour chacune des deux classes. L'exposé de la suite du processus de segmentation fait l'objet de la section suivante.

5.4 Segmentation couleur proposée

5.4.1 Description des différentes étapes

Comme nous l'avons déjà souligné au paragraphe 5.1.2, une des nouveautés de l'algorithme de segmentation bâti ici, outre l'élaboration de la transformation HSV améliorée, consiste en la succession des prétraitements employés, avant le calcul final de la LPE. Chacun de ces prétraitements est par ailleurs appliqué indépendamment à chacune des deux classes de pixels obtenues, suite à la transformation HSV améliorée et que nous illustrons sous la forme d'une image binaire (en blanc, les pixels possédant une couleur significative, en noir les pixels ne possédant qu'un niveau de gris), dans la figure 5.14, pour l'image originale présentée dans la figure 5.10, (a). L'image binaire présentée n'est en fait pas le résultat brut de la transformation HSV améliorée, mais le résultat d'un léger filtrage morphologique de type ouverture/fermeture de taille 1, de façon à éliminer les quelques points erronés, dans chacune des classes. C'est sous ce masque filtré que seront effectués les traitements suivants pour chacune des classes.

Le processus complet de segmentation s'organise alors ainsi :

- transformation HSV améliorée de l'image originale conduisant à deux classes de pixels ;
- réduction (quantification) séparée pour chacune des classes du nombre de niveaux de gris ou du nombre de couleurs dans l'image (section 5.4.2) ;
- filtrage médian appliqué indépendamment sur les deux classes (section 5.4.3) ;

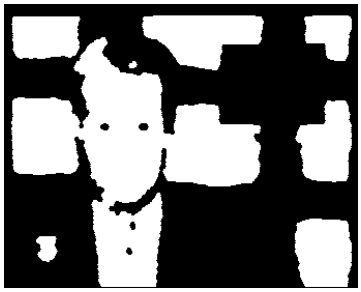


FIG. 5.14: Illustration de la classification pixels de couleur / pixels gris résultant de la transformation HSV améliorée de l'image originale présentée dans la figure 5.10, (a).

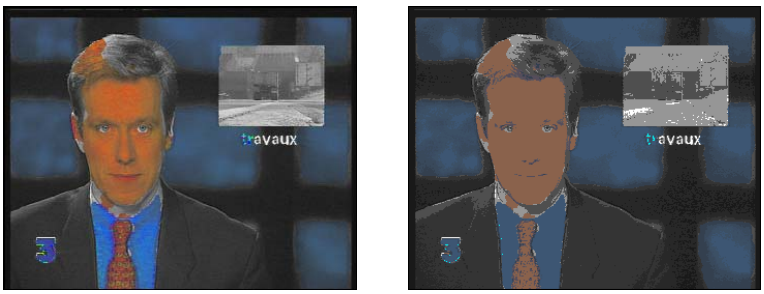
- calcul d'un gradient différent, suivant que l'on se positionne dans une région à niveaux de gris, dans une région couleur, ou sur une frontière entre les deux classes (section 5.4.4) ;
- construction d'une première LPE, suivie d'une étape supplémentaire de segmentation hiérarchique aboutissant à une seconde et dernière LPE (section 5.4.5), pour chacune des classes ;
- fusion des deux dernières LPE obtenues pour chacune des classes, pour construire la segmentation finale.

Nous proposons à présent une présentation plus détaillée de chacune de ces étapes successives et des raisons ayant conduit à leur conception.

5.4.2 Réduction

La transformation HSV améliorée de l'image originale a donné lieu à un renforcement de l'impression de couleur ou au contraire à sa suppression suivant l'une ou l'autre des deux classes de pixels. Une telle transformation contribue en outre, de par les deux projections, soit sur la surface du cône HSV, soit sur l'axe de luminance, à une diminution du nombre de dimensions de l'espace couleur utilisé. Par là, on agit dans le sens de la simplification de l'image originale.

Cette simplification n'est cependant pas suffisante, au regard du but que nous nous sommes fixés de ne conserver au final que les grandes régions de l'image, homogènes en couleur.



(a) transformée HSV améliorée (b) réduction

FIG. 5.15: Etape de réduction des couleurs et niveaux de gris (b) à partir du résultat de la transformation HSV améliorée (a).

Pour cette raison, le deuxième prétraitement appliqué consiste en une réduction drastique du nombre de couleurs (pour les régions colorées) et du nombre de niveaux de gris (pour les régions à niveaux de gris). On passe ainsi d'une image contenant en général environ 10000 couleur (codée sur 24 bits), à une image ne possédant que 10 couleurs et 6 niveaux de gris.

Ces deux chiffres sont choisis expérimentalement pour deux raisons. Il s'agit, d'une part, de respecter le pouvoir discrimant de l'œil qui distingue un nombre de couleurs supérieur à celui des niveaux de gris. D'autre part, on arrive ainsi à un total de 16 niveaux de gris et couleurs, valeur classique pour des données numériques en traitement d'images notamment (les données sont codées sur 4 bits). Cette valeur de 16 a été sélectionnée par opposition aux 16 millions de couleurs de l'image originale (24 bits) et aux 256 couleurs obtenues par un passage à 8 bits. L'ordre de grandeur de la diminution obtenue correspond alors véritablement à une réduction conséquente du nombre de couleurs.

L'algorithme utilisé pour chaque classe de pixels est basé sur les travaux de Heckbert [58], sans diffusion. Il repose classiquement sur une analyse statistique de l'occurence des couleurs et la réduction par remplacement des couleurs les moins représentées.

On effectue donc deux quantifications séparées, suivant le masque des régions blanches et des régions noires de la figure 5.14, menant au résultat présenté dans la figure 5.15.

La réduction sévère appliquée a pour effet d'homogénéïser les plages de couleurs ou de niveaux de gris. Cependant, suite à cette étape, les images restent encore suffisamment bruitées (présence de multiples points isolés au milieu de plages homogènes), pour qu'une étape supplémentaire de filtrage s'impose.

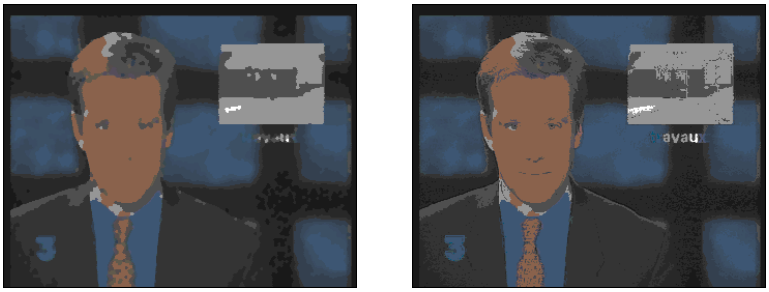
5.4.3 Filtrage

Effectuer un filtrage des images à segmenter est une étape classique et nécessaire à la limitation de la sursegmentation et à l'obtention d'une LPE correcte. En morphologie mathématique, les filtres utilisés sont majoritairement des filtres morphologiques alternés séquentiels, soit de type ouverture-fermeture classiques, soit par reconstruction, ou des nivellements, opérateurs introduits tout récemment par Meyer [71]. Si ces filtres ont prouvé leur très grande efficacité dans le cas d'un traitement à niveaux de gris, leur application à des images couleur pose le problème déjà évoqué du travail dans un espace vectoriel sans relation d'ordre intuitive. Les définitions de ces divers filtres, hormis les nivellements, sont proposés dans l'annexe B.

Dans un but de comparaison, nous fournissons les résultats de l'application d'un filtre alterné séquentiel par ouverture et fermeture de taille 1 (pour des tailles supérieures, trop d'informations sont supprimées de l'image originale) et d'un filtre alterné séquentiel par reconstruction dans la figure 5.16.

Cette figure illustre de façon très explicite les avantages et les inconvénients de tels filtres. Le filtre par ouverture-fermeture classique est beaucoup plus grossier que le filtre par reconstruction. Par contre, pour une taille 1, ce dernier ne permet pas de s'affranchir suffisamment du pointillisme, c'est-à-dire des petits amas de points isolés, alors que le premier l'élimine en partie. Cependant un désavantage de poids pour chacun de ces filtres réside dans le fait que, en agissant séparemment sur chaque composante, ils introduisent des couleurs parasites. Remarquons par exemple l'apparition de pixels de couleur verte dans la cravate, dans la figure 5.16, (a).

Pour ces raisons, les filtres morphologiques n'ont finalement pas été sélectionnés pour cette étape et nous nous sommes orientés vers un opérateur véritablement vectoriel : le filtre médian vectoriel.



Filtre par ouverture - fermeture Filtre par reconstruction

FIG. 5.16: Effets des filtres alternés séquentiels par ouverture-fermeture classiques et par reconstruction sur l'image réduite à 16 couleurs.

Courant lorsqu'il s'agit d'images à niveaux de gris, le filtrage médian repose sur le fait qu'il est possible de définir un ordre sur les pixels en fonction de leur niveau de gris. Dans le cas d'images couleurs, aucun ordre total ne peut être défini, la définition de couleur médiane n'est alors pas une notion évidente. Nous donnons donc à présent une description succincte de la façon dont ce filtrage agit sur les couleurs. Pour plus d'information, le lecteur pourra se référer à [9].

Chaque pixel de l'image couleur est en fait remplacé par le pixel, parmi ses voisins et lui-même, qui est à la distance minimale, en terme de distance dans l'espace de couleur, de tous les autres points. Un avantage évident de cette façon de procéder est qu'aucune nouvelle couleur n'est créée, le filtre ne sélectionnant que des couleurs de l'image de départ.

Un tel filtrage est en outre appliqué aux trois composantes couleur en même temps, leur conférant ainsi un même poids dans la transformation. Seul point faible de ce type de filtre, il reste à choisir dans quel espace de couleur procéder de façon à calculer la distance entre les couleurs de deux pixels voisins dans l'image. Ce dernier point soulève donc à nouveau le problème de l'uniformité des différents espaces de couleurs.

Pour notre part, nous avons choisi d'appliquer respectivement deux filtrages médians classiques sur un voisinage 3×3 et successifs sur les régions de pixels à niveaux de gris, et deux filtrages médians vectoriels, toujours successifs et de même voisinage, dans l'espace RVB aux régions de pixels couleur. Le résultat de ce double filtrage est présenté dans la figure 5.17.

FIG. 5.17: Effet du filtre médian classique sur les zones à niveaux de gris et du filtre médian vectoriel sur les zones couleur.

L'image résultante ne présente aucune fausse couleur, et le pointillisme a été considérablement réduit, pour une restitution correcte des contours des objets.

Ces différents prétraitements étant réalisés, l'image est à présent suffisamment simplifiée pour permettre l'élaboration de son gradient. Cette construction, dépendante des deux classes de pixels, est discutée dans le paragraphe suivant.

5.4.4 Gradient

Une des raisons ayant conduit à la conception et l'utilisation de la transformation HSV est la constatation de la mauvaise gestion des transitions dans les niveaux de gris, par le gradient couleur *Lab*. Du fait de notre séparation des pixels de l'image en deux classes distinctes, nous proposons donc ici de construire un gradient différent suivant que l'on se trouve dans l'une ou l'autre de ces deux classes, ou bien sur leur frontière commune, cette dernière zone devant permettre de faire la fusion entre les deux informations gradient couleur et gradient à niveaux de gris.

Suite à l'étape de transformation HSV améliorée, les deux classes de pixels couleur et niveaux de gris se trouvent respectivement sur la surface du cône de l'espace HSV ou sur l'axe de luminance. Tous les autres prétraitements (réduction et filtrage) ont de même conservé cette répartition. Pour cette raison, nous avons fait le choix d'exprimer le gradient entre deux pixels dans cet espace privilégié.

Quelle que soit la classe d'appartenance des pixels, nous définissons le module du gradient en un point comme le maximum des modules des gradients entre ce point et chacun de ses voisins. L'expression de ce module entre deux points donnés A et B est, quant à elle, dépendante des classes d'appartenance de A et de B, et s'exprime simplement comme une distance entre les valeurs de A et de B.

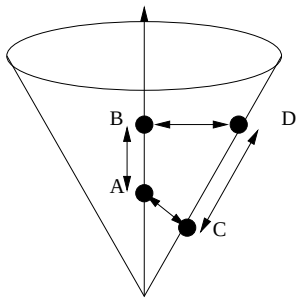


FIG. 5.18: Représentation des divers gradients couleur/couleur (distance CD), gris/gris (distance AB) et couleur/gris (distances BD ou AC), dans l'espace HSV.

Gradient entre deux pixels à niveaux de gris. Il s'agit ici du cas le plus simple, puisque nous n'avons conservé pour chacun de ces deux pixels qu'une information de luminance. L'expression du module du gradient est alors classique :

$$\text{Soit } A \text{ et } B \text{ deux pixels gris,} \qquad Grad(A,B) = |V_A - V_B| \tag{5.12}$$

Une telle distance se représente bien sûr simplement sur l'axe de luminance (cf. figure 5.18).

Gradient entre deux pixels couleur. Ces pixels appartiennent tous deux à la surface du cône HSV et leurs trois coordonnées ne sont plus indépendantes, mais s'expriment sous la forme du triplet (H, S_{max}, V) où $S_{max} = V$.

Faire le choix du calcul d'une distance euclidienne entre ces deux pixels implique le tracé d'un chemin extérieur à la surface du cône. La topologie de cette surface n'est alors plus respectée. Pour cette raison nous avons choisi de calculer une distance géodésique entre deux pixels de cette surface, c'est-à-dire la longueur du plus petit chemin inclus dans le cône et reliant ces pixels (cf. annexe B, section B.4).

Pour ce faire, et afin de représenter correctement ces géodésiques, nous proposons de déplier le cône de façon à le représenter à l'aide d'une surface plane cette fois-ci (cf. figure 5.19). Avec cette nouvelle représentation, la distance géodésique recherchée entre les points C et D correspond à une distance euclidienne classique.

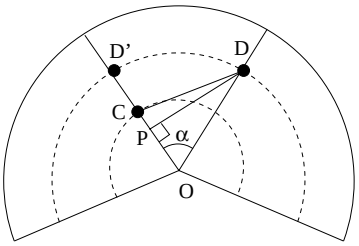
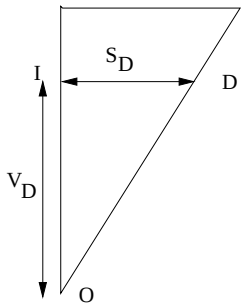


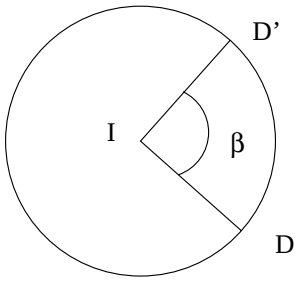
FIG. 5.19: Représentation dépliée du cône HSV.

Afin de calculer le module du gradient entre C et D, quelques notations sont essentielles ; nous les avons représentées sur la figure 5.19. On note ainsi (H_C, S_C, V_C) et (H_D, S_D, V_D) les coordonnées respectives de C et D. D' est alors le point de coordonnées (H_C, S_D, V_D), et le triangle ODD' est isocèle. On note α l'angle $\widehat{DOD'}$, qui correspond aussi à l'angle $\widehat{DOC}$, et P le projeté orthogonal de D sur la demi-droite $[OC)$. Toujours sur cette figure, on a trivialement :

$$\overset{\frown}{DD'} = \alpha OD \tag{5.13}$$



(a) Coupe verticale



(b) Coupe horizontale à hauteur des points D et D'

FIG. 5.20: Coupes du cône HSV.

Sur une coupe verticale du cône HSV, présentée dans la figure 5.20 (a), il apparaît que la distance OD a pour expression :

$$OD = \sqrt{V_D^2 + S_D^2} \tag{5.14}$$

Une coupe cette fois-ci horizontale du cône à la hauteur des points D et D' fournit la figure 5.20, (b), à partir de laquelle on obtient une autre expression de $\overset{\frown}{DD'}$:

$$\overset{\frown}{DD'} = \beta ID$$

avec $\beta = |H_{D'} - H_D| = |H_C - H_D|$ et $ID = S_D$, I étant le projeté orthogonal de D et D' sur l'axe de luminance. Notons par ailleurs que l'expression de β étant en radian, il est nécessaire d'exprimer également $H_{D'}$, H_D et H_C dans cette même unité.
Soit :

$$\overset{\frown}{DD'} = |H_C - H_D| S_D \tag{5.15}$$

La combinaison des équations 5.13, 5.14 et 5.15 fournit l'expression de l'angle α :

$$\alpha = \frac{|H_C - H_D| S_D}{\sqrt{V_D^2 + S_D^2}} \qquad V_D \neq 0 \tag{5.16}$$

Revenons à présent sur le calcul de la distance CD dans la figure 5.19 :

$$CD^2 = CP^2 + PD^2 \tag{5.17}$$

avec : $CP = OC - OP$, soit $CP = \sqrt{V_C^2 + S_C^2} - OP$ d'après l'équation 5.14. L'expression du sinus et cosinus de l'angle α permettent en outre d'obtenir :

$$\begin{aligned} \sin\alpha = \frac{PD}{OD} \quad &\text{soit :} \quad PD = \sin\alpha \sqrt{V_D^2 + S_D^2} \\ \cos\alpha = \frac{OP}{OD} \quad &\text{soit :} \quad OP = \cos\alpha \sqrt{V_D^2 + S_D^2} \end{aligned} \tag{5.18}$$

En remplaçant ces deux équations dans l'équation 5.17, on obtient :

$$CD^2 = V_C^2 + S_C^2 - 2\cos\alpha \sqrt{V_C^2 + S_C^2} \sqrt{V_D^2 + S_D^2})^2 + V_D^2 + S_D^2 \tag{5.19}$$

D'où la formulation finale du module du gradient :

$$Grad(C, D) = \sqrt{V_C^2 + S_C^2 - 2\cos\alpha \sqrt{V_C^2 + S_C^2} \sqrt{V_D^2 + S_D^2})^2 + V_D^2 + S_D^2} \tag{5.20}$$

Rappelons que pour un pixel couleur, on a la relation supplémentaire :

$$\begin{aligned} V &= S \\ \text{Soit :} \quad S^2 + V^2 &= 2 \times S^2 \end{aligned} \tag{5.21}$$

L'équation 5.20 devient donc :

$$Grad(C, D) = \sqrt{2} \sqrt{S_C^2 - 2\cos\alpha S_C S_D + S_D^2} \tag{5.22}$$

avec $\alpha = \frac{|H_C - H_D|}{\sqrt{2}}$

Gradient entre un pixel couleur et un pixel à niveaux de gris. Dans ce dernier cas, l'expression de ce gradient mixte doit idéalement être continue tant avec l'expression du gradient entre deux points à niveaux de gris, qu'avec celle du gradient entre deux points couleur.

Prenons la définition suivante d'un point limite :

Définition 19. *Un point $A(h, s, v)$ est un point limite pour la transformation HSV améliorée lorsque ses coordonnées vérifient : $sv = s_0$, où s_0 est le seuil délimitant les deux classes de pixels à niveaux de gris ou colorés.*

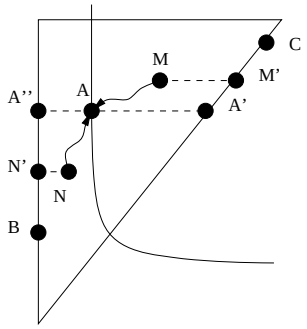


FIG. 5.21: Etude de la continuité du gradient.

Ces points limites appartiennent à l'hyperboloïde séparant l'espace HSV en deux.

Le problème de la continuité entre les deux formulations de gradient gris et couleur se pose donc entre un point de l'axe de luminance B et un point limite A d'une part, et entre un point de la surface du cône C et le même point A d'autre part. La figure 5.21 fournit une illustration de la position de ces différents points.

Considérons à présent la suite de points M contenus dans le demi-espace $\{(h, s, v),\ sv > s_0\}$ et tendant vers le point A. On dira que ces points tendent vers A, tout en restant à droite de A. La fonction vectorielle $M(h, s, v) \longrightarrow M'(h, s_{max}, v)$ est de façon évidente continue (il s'agit d'une projection).

De par sa définition le gradient couleur $Grad(M', C)$ tend donc continûment vers $Grad(A', C)$ où A' est le point de coordonnées $(h(C), s_{max}(A), v(A))$. On obtient donc en réalité une continuité à droite du gradient :

Lorsque M tend vers A, à droite de A, $Grad(M, C)$ tend vers $Grad(A', C)$.

De même, si on considère la suite de points N tendant vers A à gauche de A, i.e. dans le demi-espace $\{(h, s, v),\ sv < s_0\}$, on obtient aisément la continuité à gauche du gradient, la luminance étant continue :

$$N \longrightarrow_{\text{à gauche}} A,\ Grad(N, B) \longrightarrow Grad(A'', B)$$

Il reste donc à définir la valeur du gradient mixte en A arbitrairement

- soit comme la valeur $Grad(A', C)$, dans ce cas on conserve la définition du gradient couleur à droite ;
- soit comme la moyenne $\frac{Grad(A',C)+Grad(A'',B)}{2}$;

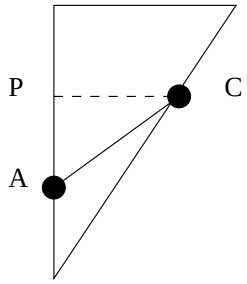


FIG. 5.22: Représentation du module du gradient mixte, entre un pixel gris A et un pixel couleur C.

– soit comme la valeur de la distance euclidienne entre B et C par exemple.

Quel que soit le choix final, on n'obtient pas de continuité totale du gradient, mais simplement deux continuités de part et d'autre de l'hyperboloïde. Pour notre part nous avons choisi la troisième formulation du gradient mixte, illustrée sur la figure 5.22. Cette dernière a pourtant tendance à renforcer l'effet de gradient entre deux pixels qui, avant transformée HSV améliorée, pouvaient être relativement proches mais de part et d'autre de l'hyperboloïde. En revanche, nous restituons ainsi plus justement les différences entre deux points, de saturations différentes mais de luminances proches. Les deux autres formulations que nous avons évoquées ont elles l'effet inverse. Dans la mesure où nous exprimons le gradient en un point comme le maximum des gradients entre ce point et l'ensemble de ces voisins, nous avons donc préféré obtenir des valeurs de gradient mixte sur-estimées dans certains cas.

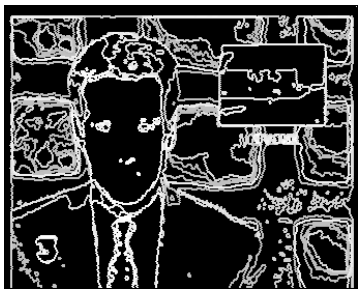


FIG. 5.23: Gradient dans l'espace HSV. On présente des valeurs de gradient renormalisées afin d'en augmenter le contraste.

Les trois formulations du gradient pour chacune des classes couleur, gris et frontière couleur/gris étant à présent établies, nous proposons sans plus tarder, dans la figure 5.23, le résultat obtenu sur l'image issue du filtrage médian. Notons que les valeurs obtenues pour chacune des classes sont cette fois-ci du même ordre de grandeur.

Cette construction d'un gradient cohérent pour chaque classe constitue la dernière étape avant la réalisation de la segmentation hiérarchique, que nous présentons à présent.

5.4.5 Segmentation hiérarchique

Suite à la construction du gradient réalisée à l'étape précédente, deux voies de poursuite de l'algorithme s'offrent alors, pour l'établissement de la segmentation hiérarchique. Le gradient ayant été bâti de façon à être continu, il est possible de réaliser la segmentation sur l'image entière, sans plus tenir compte des deux classes couleurs et niveaux de gris, avec le risque qu'une telle démarche mène à la fusion de régions de classes différentes. Si cette fusion peut, dans certains cas, permettre de revenir sur une classification erronée de certains pixels, elle constitue cependant dans la majorité des cas un retour en arrière, dans la mesure où l'on peut voir le résultat de la classification HSV comme une première segmentation de l'image de départ. Pour cette raison, une deuxième alternative consiste à procéder à la segmentation hiérarchique sur chacune des classes indépendamment, comme s'il s'agissait de deux images distinctes. On réalise alors simplement l'union des deux partitions résultantes pour obtenir la segmentation complète de l'image originale.

Afin de tirer parti au maximum de la classification couleur / gris, nous avons opté pour cette deuxième solution, quitte à revenir par la suite, lors d'un post-traitement à définir, sur les cas de classification erronée. Toutefois le processus de segmentation hiérarchique mis en jeu est identique quelle que soit l'alternative choisie. Nous le détaillons donc à présent.

5.4.5.1 Construction

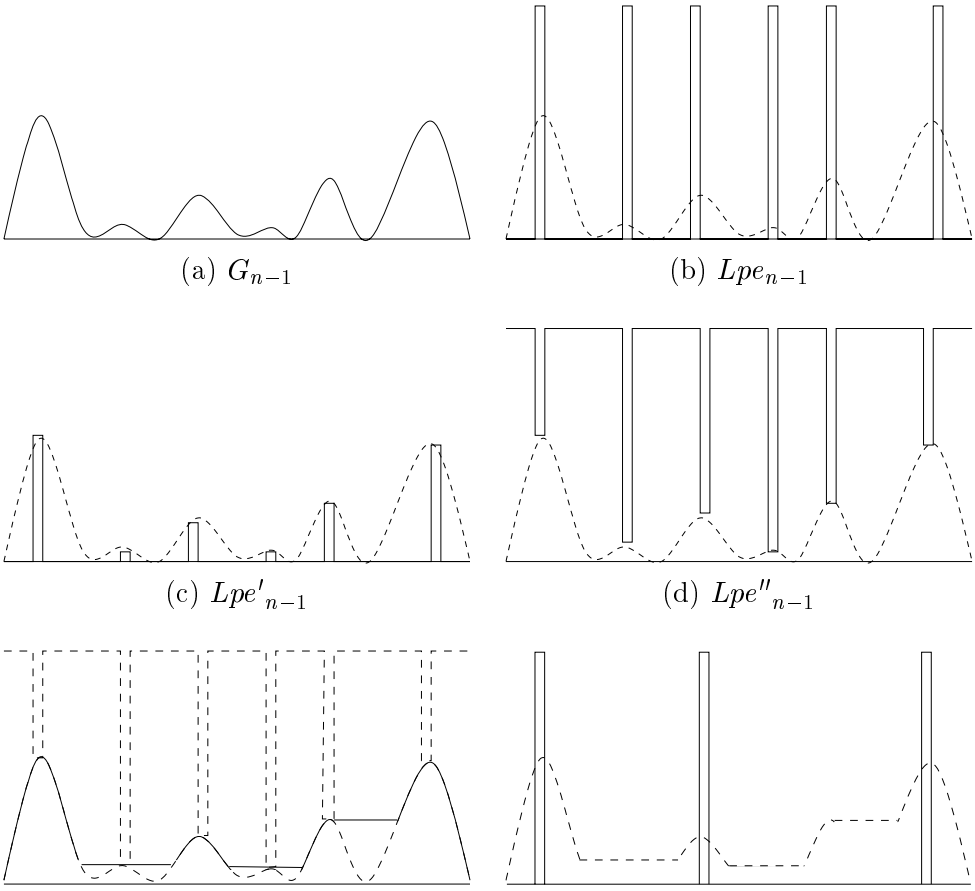
Ainsi que nous l'avons déjà évoqué au paragraphe 5.2.2.4, la segmentation hiérarchique, comme son nom l'indique, consiste à obtenir des segmentations de plus en plus grossières, à partir d'une première segmentation de l'image originale [14]. Ces nouvelles segmentations sont construites grâce à la suppression des contours les plus faibles, i.e. ayant de faibles valeurs de gradient, dans la segmentation originale.

A chaque niveau de la hiérarchie, cette suppression est réalisée grâce à une modification du gradient de l'image originale, processus que nous illustrons dans les étapes successives de la figure 5.24. Supposons que la segmentation soit arrivée au niveau n de la hiérarchie. A cette étape on dispose du gradient G_{n-1} (figure 5.24, (a)) et de la LPE Lpe_{n-1} de l'étape précédente (figure 5.24, (b)). La LPE apparaît comme une image binaire de contours blancs entourant des bassins versants de couleur noire.

La première étape consiste en la modification de la LPE Lpe_{n-1}, de façon à mettre les valeurs du gradient G_{n-2} à l'endroit des contours. On obtient ainsi une image Lpe'_{n-1}, toujours noire dans les bassins versants et avec des contours de différents niveaux de gris, correspondant aux valeurs du gradient (figure 5.24, (c)). Lpe'_{n-1} est ensuite complémentée comme l'illustre le schéma (d) (les bassins versants deviennent blancs et les contours prennent pour valeur, la différence entre la valeur maximale de l'image et leur ancienne valeur). Cette nouvelle image Lpe''_{n-1} étant obtenue, il reste à reconstruire (cf. processus de reconstruction morphologique, annexe B) le gradient G_{n-1} sous la LPE Lpe''_{n-1}. On obtient ainsi le gradient modifié de l'étape n, G_n, illustré par le schéma (e). La nouvelle LPE Lpe_n peut maintenant être construite à partir de ce gradient.

Théoriquement (cf. schéma (f)), cette LPE reprend tous les anciens contours de la LPE précédente, sauf ceux éliminés par modification du gradient.

En pratique, malheureusement, la construction de la nouvelle LPE ne restitue pas exactement les contours de l'étape $n-1$ et on assiste souvent à un décalage de un, voire plusieurs pixels. Trois raisons peuvent expliquer ce décalage. Nous les décrivons successivement par la



(a) G_{n-1} (b) Lpe_{n-1}

(c) Lpe'_{n-1} (d) Lpe''_{n-1}

(e) $G_n = D^{\infty}_{Lpe''_{n-1}}(G_{n-1})$ (f) Lpe_n

FIG. 5.24: Etapes de modification du gradient et d'obtention de la nouvelle LPE pour le niveau n de la segmentation hiérarchique.

suite, ainsi que les solutions que nous avons mis en œuvre pour y remédier.

5.4.5.2 LPE double

La première cause du décalage des contours observé lors de la construction de la LPE provient directement de la façon dont est réalisée l'inondation. En théorie, tous les pixels atteints par l'eau à un instant t doivent être traités simultanément. En pratique, un ordre implicite est instauré dans la mesure où les pixels d'un niveau donné sont, par exemple, stockés dans une file d'attente [17], puis traités séquentiellement au fur et à mesure qu'ils sont extraits de la file. Cet ordre est responsable du décalage éventuel d'un pixel de la LPE au niveau des contours. Nous illustrons cet état de fait dans le schéma de la figure 5.25.

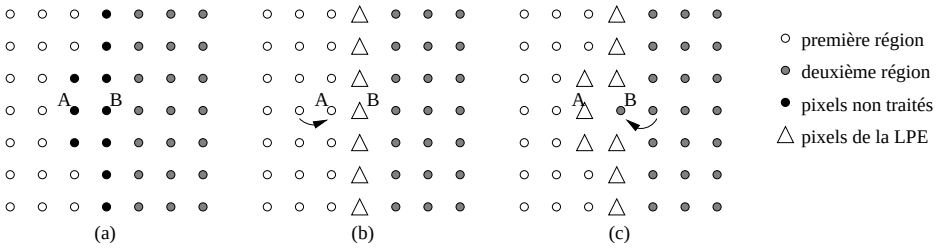


FIG. 5.25: Placements différents de la ligne de partage des eaux sur les contours, suivant l'ordre de traitements des pixels : (a) A est traité avant B, (b) B est traité avant A.

Faisons l'hypothèse que l'inondation s'est propagée dans chacune des deux régions représentées, en blanc et en gris (respectivement à droite et à gauche de l'image). Au niveau courant, seuls les points intermédiaires (en noir) restent à étiqueter. Regardons plus précisément les points A et B, et imaginons que le point A soit extrait le premier de la file d'attente. Ce point ne possédant dans son voisinage que des points étiquetés "blanc" ou "noir", il est lui-aussi rajouté à cette région.

Au moment où B est extrait, B possède par contre des voisins blancs (A notamment) et des voisins gris, il devient donc un point LPE (marqués par un triangle) (cf. figure 5.25, (b)). Si l'ordre dans la file d'attente avait été inverse, B au contraire aurait été étiqueté "gris" et A appartiendrait à la LPE, comme l'illustre la situation de la figure 5.25, (c). Dans le processus de segmentation hiérarchique plusieurs LPE successives sont effectuées. De l'une à l'autre, on ne peut être sûr que l'ordre de traitement des pixels est le même. Des décalages de un pixel ont donc lieu, lorsque l'ordre est modifié.

Une façon simple de règler ce problème consiste à étiqueter les deux points A et B comme des points LPE. On construit ainsi des lignes de contours doubles, aux endroits où les pixels se retrouvent dans la même situation que celle de la figure 5.25.

Un tel algorithme de construction d'une LPE double est simple à mettre en œuvre. Par rapport à un algorithme classique de LPE par file d'attente hiérarchique [17], deux points diffèrent. Tout d'abord, à un niveau de priorité p donné, on effectue une première passe sur l'ensemble des pixels de ce niveau. Dans le cadre de notre exemple, faisons l'hypothèse que A est traité avant B. Dans ce cas, comme dans l'algorithme classique de LPE, A est étiqueté comme la région de gauche. De plus on garde en mémoire son statut d'"étiqueté temporaire". Lorsque vient le tour de B d'être traité, B a parmi ses voisins des pixels de la région de droite et A comme pixel de la région de gauche, étiqueté temporairement. Dans ce cas on interdit à A de propager son étiquette, B prend donc temporairement celle de la région droite. Cependant,

encore une fois on garde en mémoire la présence de A parmi les voisins de B. On examine ainsi tous les voisins de B lors d'une première passe.

Lorsque ce premier tour des voisins de B est terminé et si B n'a pas en plus été étiqueté directement LPE, B appartient à la région de droite et contient parmi ses voisins un point étiqueté temporairement et avec une étiquette différente. Ce cas de figure justifie un deuxième examem des voisins de B. Tous les voisins étiquetés temporairement et avec une étiquette différente, et le point B deviennent alors des points de la LPE.

Une fois tous les points du niveau de priorité p examinés, on effectue un second tour de ces points, afin d'étiqueter cette fois-ci définitivement les points ayant encore une étiquette temporaire, i.e. qui ne sont pas devenus points de la LPE.

L'algorithme de production d'une LPE double, qui ne constitue par ailleurs pas vraiment une nouveauté dans la mesure où ses prémices ont déjà été introduits par Beucher [12], permet donc de s'affranchir de l'ordre de traitement des pixels, et donc des décalages des contours dus à cet ordre. Cependant une fois cette nouvelle LPE construite, on s'aperçoit de la persistance de certains décalages qui cette fois-ci peuvent être importants. Après analyse de ces situations, deux nouveaux cas se présentent que nous détaillons dans le paragraphe suivant.

5.4.5.3 Artéfact des plateaux et "boutonnières ouvertes"

Malgré l'utilisation d'une LPE double, on assiste quand même, lors de la construction de LPE successives, au décalage de certains contours. Et cette fois-ci ces décalages sont souvent de plus d'un pixel.

La première explication de l'apparition de ces décalages est la création de plateaux lors de la modification du gradient. Une telle situation, illustrée dans la figure 5.26, a lieu lorsqu'une crête minimale du gradient est entourée de deux crêtes plus élevées et exactement de même hauteur. Dans ce cas, la modification du gradient comble entièrement la cuvette entre les deux crêtes élevées, produisant ainsi un nouveau plateau. Lors de la construction de la LPE, un contour apparaît alors exactement au milieu du plateau créé. La gestion des plateaux lors de la construction de la LPE est un problème connu, et pour lequel les solutions de placement de la LPE [12, 65] proposent soit de faire appartenir le plateau entier à la LPE, soit positionnent le contour au centre de ce plateau. Quelle que soit la méthodologie employée, nous assistons, pour notre problème à la création de contours parasites, qui n'étaient pas présents dans les LPE précédentes.

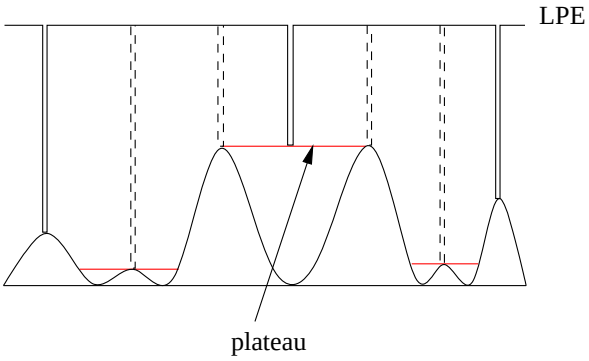


FIG. 5.26: Création de plateau par modification du gradient et LPE résultante.

L'autre cause de décalage ou même d'apparition de contours se rencontre dans certaines situations particulières, similaires aux "boutonnières", décrites dans la thèse de Beucher [12], et que nous appelerons par la suite "boutonnières ouvertes". Un exemple d'une telle situation est proposée dans la figure 5.27. Avant toute modification du gradient, la LPE vient se positionner telle que cela est représenté sur le schéma de gauche, en forme de croix séparant quatre bassins versants de minima tous égaux à 1. La modification du gradient supprime les contours les plus faibles, qui correspondent ici aux deux contours entre les minima A et B, et entre les minima C et D. Par reconstruction les quatre minima sont comblés et mis à la valeur 3, valeur minimale des contours à supprimer. Une fois cette reconstruction achevée, la LPE ne se positionne cependant pas correctement du fait du couloir vertical entre les minima C et D, à la valeur 3 entouré de deux murs à la valeur 4. On assiste ainsi à un décalage de la LPE au détriment des minima C et D.

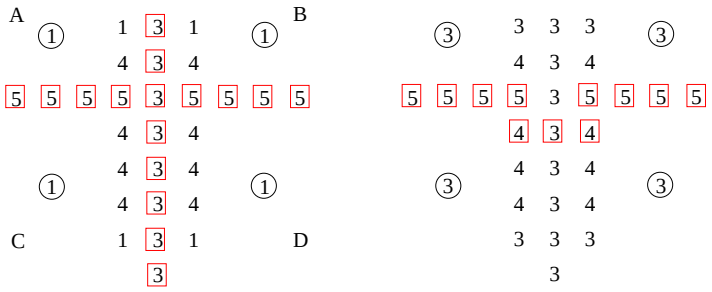


FIG. 5.27: Cas particulier de déplacement des contours.

Une telle situation ainsi que celle donnant naissance à de nouveaux plateaux sont malheureusement des artéfacts qu'il est impossible d'éviter, dès qu'on reconstruit une nouvelle LPE, double ou non, lors d'un processus de segmentation hiérarchique. Suite à cette constatation, la seule alternative restante est de construire une fois pour toute une LPE, avant toute hiérarchisation, puis de procéder à une fusion de certains bassins versants, sans modification des contours non concernés par la fusion. C'est ce que nous présentons dans la section suivante.

5.4.5.4 Segmentation hiérarchique sur graphe

Il s'agit ici d'exposer dans le détail l'algorithme que nous avons mis en place afin d'obtenir les mêmes niveaux de segmentation que par la segmentation hiérarchique classique, mais sans reconstruire de nouvelle LPE à chaque étape. Pour ce faire, nous avons pris la décision de ne plus travailler au niveau des images directement, mais à partir de graphes représentant les relations de voisinages entre régions. L'ensemble des différentes étapes du processus est illustré sur un exemple concret dans la figure 5.28. Au fur et à mesure de la description de l'algorithme, nous ferons donc référence à cet exemple, pour une illustration concrète.

On construit ainsi un premier graphe à partir de la LPE originale (cf. figure 5.28, (a) et (b)) : chaque nœud correspond à un bassin versant et chaque arête à un contour entre deux bassins versants voisins. On commence par donner à chaque arête la valeur minimale du contour correspondant, puis on étiquète chaque nœud à la valeur minimale de toutes les arêtes partant de ce nœud. Cette première étape réalise la modification du gradient (figure 5.28, (b)).

La fusion des bassins versants s'effectue alors en 3 étapes :

– On commence par fusionner les bassins versants voisins qui, suite à la modification du

gradient, se trouvent avoir la même valeur (figure 5.28, (c)). Au niveau du graphe, la fusion de deux bassins versants se fait par suppression d'un des deux nœuds et par union des deux ensembles d'arêtes provenant de chaque nœud. Lorsque deux arêtes sont redondantes, i.e. les deux nœuds originaux avaient un voisin commum, on ne conserve de façon évidente qu'une seule arête et on lui affecte la valeur minimale des deux arêtes originales.

- On cherche les minima parmi les nœuds du graphe (figure 5.28, (d)). Tout comme dans une LPE classique, ces nœuds particuliers seront les points de départ de l'inondation.
- On procède à une inondation du graphe classique, mais qui, au lieu de se dérouler au niveau du pixel, se déroule au niveau des régions. Cette inondation au niveau des régions est le premier point clé de cet algorithme : on décide en effet globalement de l'ajout d'une région entière à un bassin versant, par propagation des étiquettes. Les contours entre deux régions sont donc entièrement supprimés lorsque celles-ci fusionnent. La première étape de l'inondation correspond à la figure 5.28, (e). L'étiquette correspondant à la couleur magenta se propage. Pour reprendre l'analogie avec la montée d'eau, lorsqu'une région est atteinte au contraire par de l'eau provenant de deux minima différents, cette région est classée comme appartenant à la ligne de partage des eaux, et ne peut plus propager l'inondation. L'étiquetage de certaines régions comme appartenant à la LPE est cette fois-ci proposé dans la figure 5.28, (f).

L'inondation étant terminée, le graphe est découpé en deux types d'entités :

- les bassins versants ayant chacun une étiquette différente et constitués d'une ou plusieurs régions ayant entièrement fusionné ;
- les régions étiquetées LPE, qui malgré leur appartenance à la LPE, n'ont pas fusionné et constituent chacune un nœud élémentaire du graphe.

Cet étiquetage de certaines régions entières comme appartenant à la LPE, ainsi que leur conservation en tant que régions indépendantes constituent le deuxième point clé de cet algorithme. Au contraire d'une inondation au niveau des pixels, l'ensemble des régions LPE ne sont donc pas fusionnées dans le cas général.

Comme dans l'algorithme classique de construction de la LPE au niveau pixel, certaines régions arrivent à la fin du processus d'inondation sans pour autant avoir été étiquetées. Une telle situation intervient lorsqu'une ou plusieurs régions sont entourées exclusivement d'une ou plusieurs régions de type LPE. Ces dernières ne pouvant propager l'inondation, les régions intérieures restent non-étiquetées.

Dans ce cas, si ces régions constituent une feuille du graphe (i.e. elles n'ont qu'une seule région voisine et de type LPE, ou encore, elles sont complètement insérées dans une seule et même région), alors, et dans ce cas particulier seulement, ces régions sont fusionnées à la région LPE les entourant. Sinon, lorsqu'elles sont non-étiquetées et entourées de plusieurs régions LPE, elles sont elles-mêmes étiquetées LPE mais restent indépendantes les unes des autres (i.e. aucune fusion n'a lieu).

Revenons à présent sur les trois situations de décalage des contours dans les LPE successives, à l'origine de la conception de notre algorithme de segmentation hiérarchique sur graphe.

Le décalage d'un pixel dû à l'ordre de traitement des pixels est supprimé de façon évidente, puisqu'on travaille à présent au niveau région et sur une seule et même LPE initiale. Cependant il est à noter que l'ordre de traitement est à présent instauré au niveau des régions, qui pourront

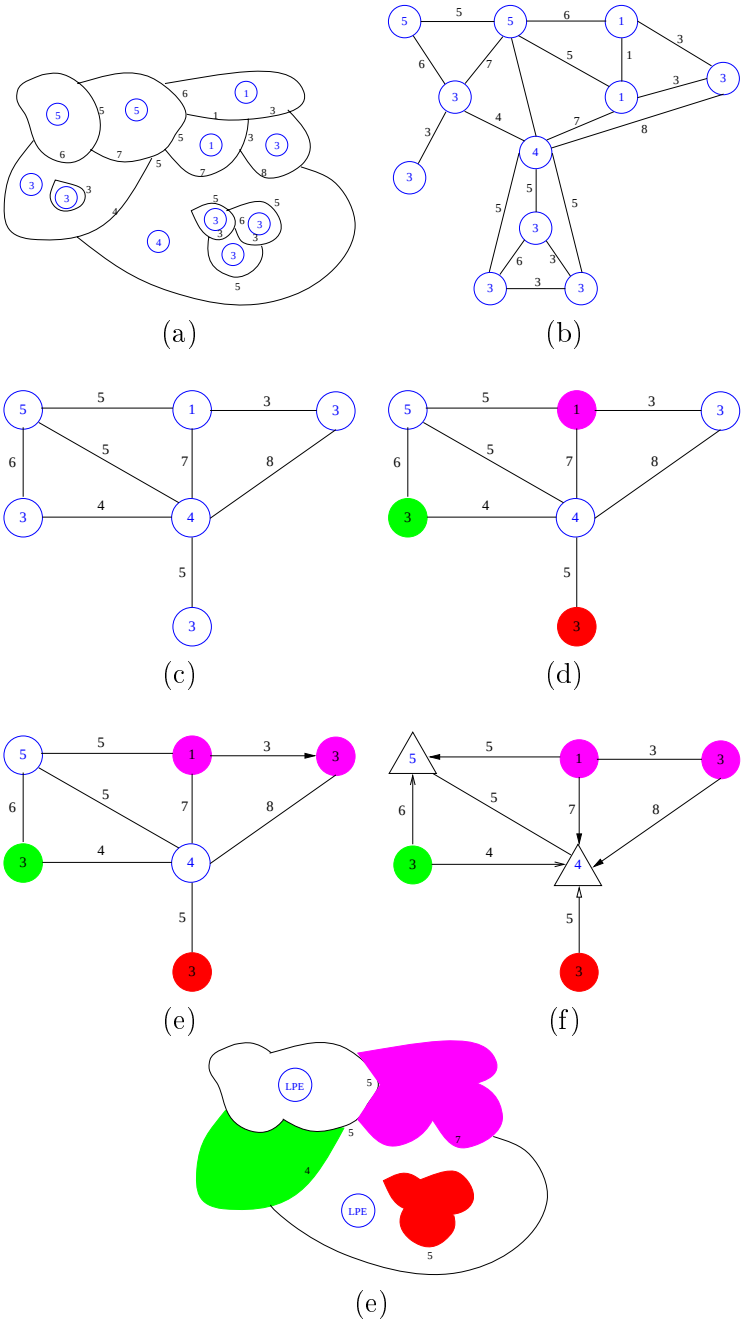


FIG. 5.28: Différentes étapes du processus de segmentation hiérarchique sur graphe.

donc se retrouver alternativement affectées à l'une ou l'autre des étiquettes voisines. Toutefois, une fois les conséquences de l'ordre instauré à une étape n de la segmentation hiérarchique appliquées, aucune des étapes suivantes ne modifie cet ordre, ni la fusion établie.

Le cas de la création de plateau, responsable de la génération de faux contours dans les LPE successives, se retrouve éliminé du fait de la fusion des régions entières. Le contour se place donc sur un bord ou l'autre du plateau, ou bien disparaît complètement comme c'est le cas lors de la fusion de régions voisines de même étiquette, dans l'exemple de la figure 5.28, (c).

Enfin l'apparition de boutonnières ouvertes n'a plus lieu d'être, puisqu'encore une fois les fusions sont effectuées au niveau région, et non pour chaque pixel indépendamment.

L'algorithme de segmentation hiérarchique sur graphe que nous venons de bâtir constitue l'étape finale de notre processus entier de segmentation. Nous venons d'exposer les divers points forts de cet outil, du point de vue de sa réalisation concrète, capable d'apporter une solution aux problèmes de décalage des contours. Il convient de rappeler à présent pourquoi notre choix s'est porté vers cet opérateur, plutôt que vers une segmentation à base de marqueurs, par exemple. C'est ce que nous effectuons au début du paragraphe suivant, consacré par la suite à la présentation de résultats sur des images de contenus variés.

5.4.6 Résultats

Dans le domaine de l'indexation d'images et de séquences, les sources originales, les images, sont inombrables et leurs contenus très divers. L'utilisation de marqueurs pour situer les objets d'intérêt est à proscrire, dans la mesure où ces objets sont a priori inconnus. Pour cette raison, le principe de segmentation hiérarchique permet de s'affranchir du choix des marqueurs. A ce stade notons que les hiérarchies de segmentations font également l'objet de nombreux autres travaux [67, 2, 98, 25, 72]. Les segmentations alors obtenues, tout comme celle que nous utilisons, reposent toutes sur le choix, une fois la hiérarchie de segmentation établie, du niveau adéquat pour l'application envisagée. Sur ce point, i.e. le choix du critère d'arrêt de la hiérarchisation, nos méthodes diffèrent cependant. En effet, si les techniques précédemment citées se basent soit sur une taille moyenne des régions obtenues, soit sur un nombre de régions maximal au final (critères qui sous-entendent une connaissance même minime du contenu de la scène), l'ensemble du processus que nous avons choisi, aboutit dès la première étape de la hiérarchisation à une segmentation suffisamment simplifiée, quel que soit le contenu de l'image originale, pour que toute étape supplémentaire soit surperflue, voire impossible. Rappelons que nous effectuons en réalité deux segmentations hiérarchiques pour chacune des classes couleurs et niveaux de gris. Quelle que soit la classe, poursuivre la segmentation hiérarchique aboutit cependant majoritairement à une segmentation en deux ou trois, et même parfois une seule région (la classe entière dans ce cas). Cette simplification en une seule étape est obtenue bien sûr grâce à la segmentation hiérarchique, mais aussi et surtout grâce à l'ensemble des prétraitements effectués auparavant.

Pour ces deux raisons, aucun choix véritable de l'arrêt du processus de segmentation hiérarchique n'est effectué : aller au delà d'une étape conduit irrémédiablement à la perte de trop d'informations et à la fusion d'un trop grand nombre de régions. Par là, l'ensemble de notre processus s'afranchit donc totalement d'un quelconque critère d'arrêt : une seule étape de segmentation hiérarchique est effectuée, quelle que soit l'image de départ.

Nous fournissons à présent (cf. figures 5.29 et 5.30) divers exemples de segmentations complètes d'images de contenus suffisamment variés pour prouver le caractère général de notre

méthode.

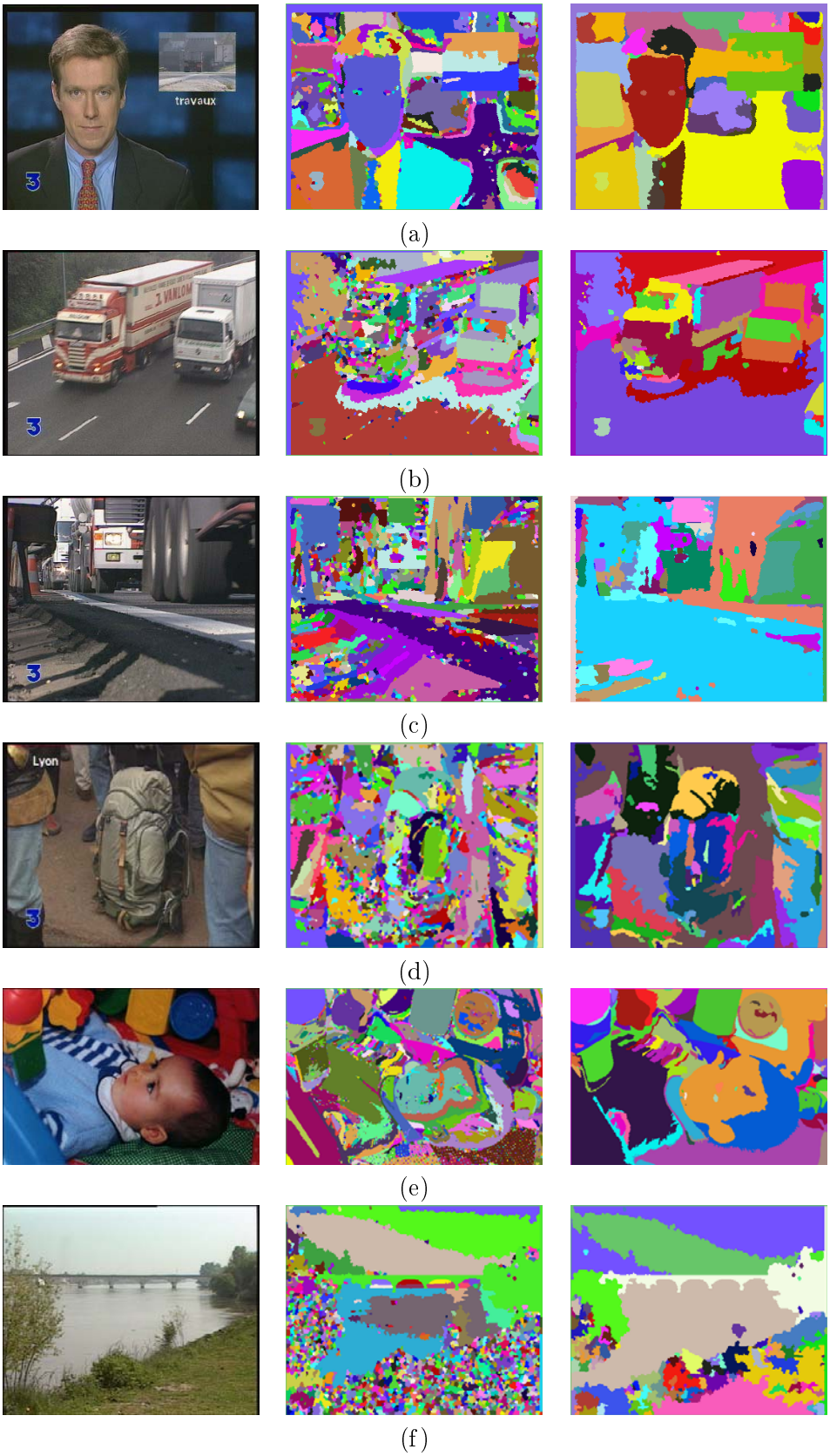
Au vu de ces deux séries d'exemples, la validité de notre méthode de segmentation en grandes régions homogènes est confirmée. Rappelons que le processus est par ailleurs automatique quelle que soit l'image d'entrée. Certains points méritent cependant qu'on s'y attarde. Responsables d'erreurs de segmentation qui apparaissent dans les exemples fournis, ils nécessitent encore quelques améliorations. Notons ainsi :

- l'apparition de fusions erronées, soit de deux régions de niveaux de gris très différents (fond et imagette de l'image (a), figure 5.29, ou fond et visage de l'image (e), figure 5.30), soit de deux régions de couleurs différentes (ciel et arbre de l'image (c), figure 5.30). Deux raisons peuvent expliquer de telles fusions. Tout d'abord dans le processus de segmentation hiérarchique, l'étape de modification du gradient peut conduire à la génération de zones de plateaux, qui ne donnent plus lieu à la création de contours lors de la construction de la nouvelle lpe. C'est par exemple ce qui s'est produit dans l'image (e), figure 5.30. L'autre raison provient du fait même de la construction d'une segmentation hiérarchique basée sur le gradient. Ce critère pourrait sans aucun doute être avantageusement complété par d'autres critères tels que des moments couleur supplémentaires, des critères de texture, ou bien encore la dynamique des contours [50, 51].
- l'absence de fusion de régions pourtant très homogènes, comme c'est le cas de la zone de mer pour l'image (d), figure 5.30. Là encore, l'ajout de critères supplémentaires, de caractérisation de la texture notamment, permettrait d'augmenter la robustesse de l'étape de segmentation hiérarchique.
- des erreurs de placement du seuil de la transformation HSV améliorée combinées à une mauvaise étape de réduction des couleurs (un exemple est fourni par l'image (b), figure 5.30). Une étape supplémentaire de retour sur la classification couleur / gris permettrait sans doute de corriger les erreurs produites. Quant à la réduction de couleurs, ou de niveaux de gris, il est envisageable de remplacer le nombre fixe de couleurs et de niveaux de gris obtenu au final, par des valeurs déterminées automatiquement pour chaque image, en fonction d'une étude de ses histogrammes des teintes et des luminances. L'algorithme de réduction utilisé actuellement module déjà ce nombre final de 10 couleurs et 6 niveaux de gris, puisqu'il s'agit uniquement d'une limite maximale imposée. Cependant pour l'exemple de l'image (f), figure 5.29, autoriser 10 couleurs alors que l'image, après transformation HSV améliorée ne contient au final qu'une teinte globalement verte, mène au partitionnement de la zone d'herbe en un nombre conséquent de petites régions qui n'ont pas lieu d'être. Le même phénomène s'est produit pour le foulard rouge de l'image (e), figure 5.30.

Ceci termine notre analyse des résultats obtenus par notre processus de segmentation. Quelques solutions visant à améliorer les points faibles de notre algorithme ont été suggérées. Nous proposons à présent de conclure ce chapitre de segmentation spatiale des images clés par un bilan de nos apports et par l'évocation de perspectives de poursuite de nos travaux, dans ce domaine.

5.5 Conclusion

La segmentation des images clés issues de la structuration linéaire temporelle pose le problème de la grande variété de contenus, dans un cadre très large d'indexation de documents



(a) (b) (c) (d) (e) (f)

FIG. 5.29: Résultats de l'ensemble du processus de segmentation appliqué à des images de contenus divers (première partie).

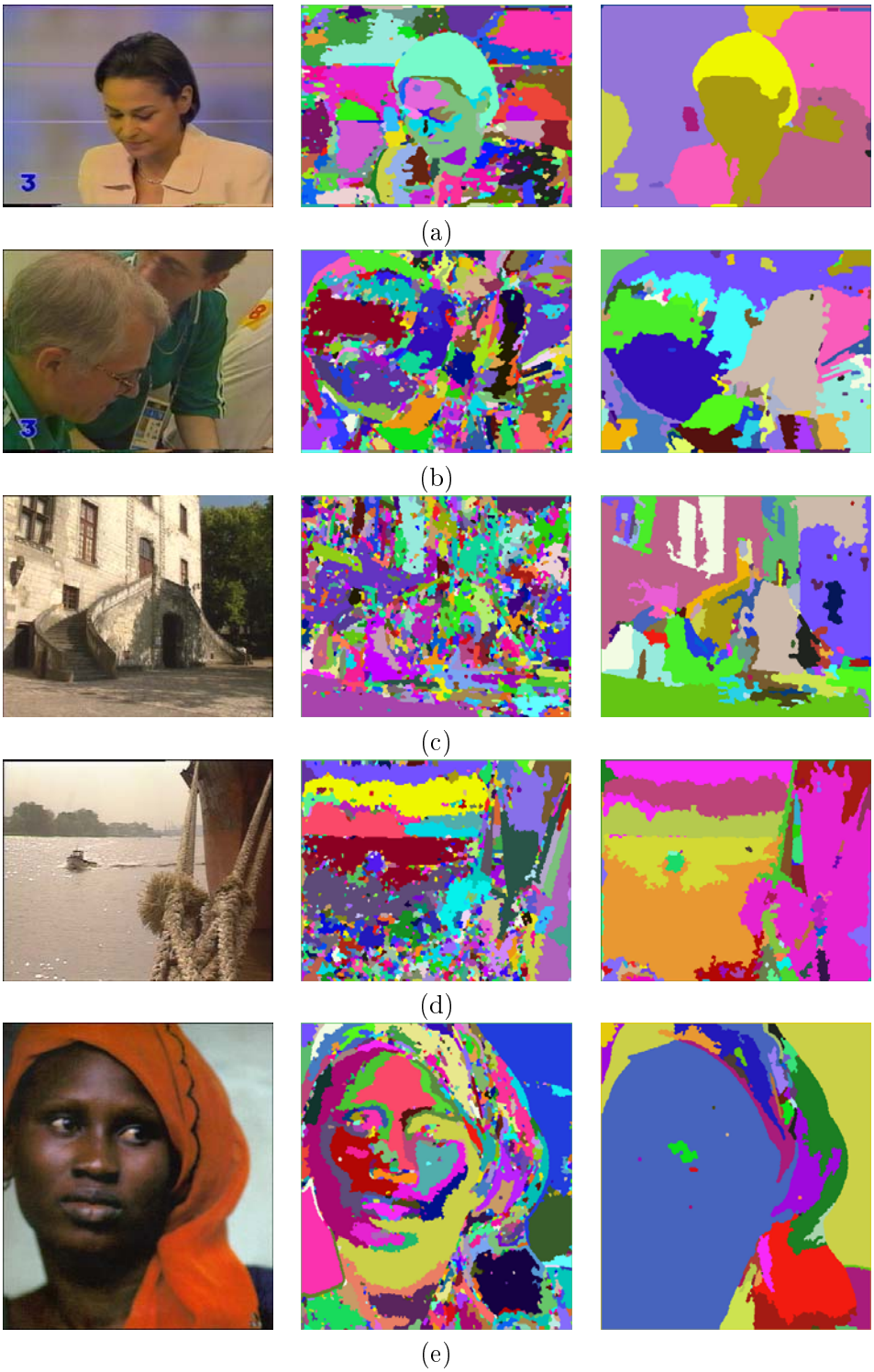


FIG. 5.30: Résultats de l'ensemble du processus de segmentation appliqué à des images de contenus divers (deuxième partie).

vidéo quelconques. Face à cette contrainte, nous sommes en mesure de proposer un processus complet et automatique de segmentation en grandes régions homogènes - les détails fins ne sont pas extraits. Les points forts de ce processus reposent sur :

- l'élaboration d'un opérateur appelé transformée HSV améliorée, permettant d'une part la classification automatique des pixels en deux groupes, ceux possédant une couleur significative, et ceux ne possédant qu'un niveau de gris, et d'autre part, le renforcement de l'impression de couleur et du niveau de gris, suivant la classe d'appartenance des pixels ;
- une succession de prétraitements conduisant à une simplification extrême de l'image à segmenter, et sans précédent, puisque chaque prétraitement est appliqué distinctement à chacune des classes couleur/niveau de gris ;
- la construction de trois gradients pour chacune des classes couleur, niveau de gris et la zone de frontière entre ces deux classes, permettant une continuité de l'information d'une classe à l'autre ;
- l'application d'une seule étape de segmentation hiérarchique sur graphe. Pour ce dernier point, trois contributions particulières méritent d'être soulignées : premièrement grâce à l'ensemble du processus, aucun test d'arrêt dépendant de l'image de départ n'est nécessaire, une seule étape de la hiérarchisation suffit ; deuxièmement, les trois événements responsables, dans le processus de hiérarchisation, du décalage de certains contours ont été catalogués et étudiés. Enfin, une solution finale de segmentation hiérarchique sur graphe a été développée, permettant d'éliminer ces décalages.

De l'ensemble du processus, par ailleurs validé par les segmentations obtenues sur des images variées, se dégagent cependant trois voies de poursuite de nos travaux :

- Du fait des deux segmentations hiérarchiques menées en parallèle sur chacune des classes couleurs et niveaux de gris, aucune fusion n'est possible entre deux régions issues de classes différentes. Si une telle contrainte est logique dans le contexte d'une classification correcte ou sans ambiguïté, elle est remise en cause dans les cas d'erreurs de classification, dues à un mauvais choix du seuil de la transformation HSV améliorée par exemple. De même, il paraît souhaitable, pour des régions situées de part et d'autre de l'hyperboloïde limitant les deux classes, et pour lesquelles la classification est parfois ambiguë, d'autoriser de telles fusions inter-classes. Pour ces raisons une étape supplémentaire de retour sur la classification après segmentation peut s'avérer intéressante.
- Le second axe de recherche et de poursuite de nos travaux repose sur un effort de combinaison de plusieurs critères (couleur, texture, etc.), afin de mener la segmentation hiérarchique. Une telle segmentation conduirait ainsi à des niveaux hiérarchiques différents suivant l'homogénéité des régions. Il est en effet envisageable, en combinant des informations de texture et de couleur, d'autoriser plusieurs étapes de hiérarchisation pour des zones plus texturées (plus inhomogènes), et d'arrêtre le processus au bout d'une seule étape pour des régions au contraire sans texture.
- Enfin, et il s'agit d'une perspective d'amélioration plus simple et plus directe que les précédentes, l'étape de réduction et le choix des nombres de couleurs et de niveaux de gris pourraient être déterminés en fonction d'une étude des histogrammes des teintes et des luminance de l'image d'entrée, plutôt que d'être fixés à des valeurs constantes quelle que soit cette image.

Nous sommes à présent en mesure d'obtenir une segmentation en grandes régions homogènes, qui, rappelons-le encore une fois, est entièrement automatique. Il convient à présent de continuer le processus de structuration par une classification de ces différentes régions, en fonction de leur contenu sémantique. Une telle classification rejoint le processus d'extraction d'objets, deuxième volet de la structuration linéaire spatiale, qui fait l'objet du chapitre suivant.

Chapitre 6

Extraction d'objets particuliers et outils d'indexation

6.1 Introduction

Nous continuons ici la structuration spatiale d'un document vidéo, commencée au chapitre précédent, par un deuxième et dernier volet consistant en l'extraction d'objets particuliers. Au contraire de l'étape de segmentation précédente aboutissant à un partitionnement de chaque image clé, nous ne proposons que d'extraire certaines régions d'intérêt, encore appelées objets visuels [86].

Plus qu'une simple étape supplémentaire du processus de structuration, l'extraction d'objets réalise en outre un pas supplémentaire vers un niveau plus sémantique de la structuration. Il s'agit en effet non seulement d'extraire certains objets, mais également de les classifier en fonction de leur contenu : on cherche ainsi à sélectionner les zones de végétation, de ciel, de texte, les visages, etc. Une fois réalisée, une telle classification apporte tout ou partie de la réponse à des question du type : s'agit-il d'une scène en extérieur ou en intérieur, de jour ou de nuit, comportant des personnes, etc. C'est en ce sens que l'aspect purement syntaxique de la structuration développée jusqu'à présent est dépassé et un niveau plus sémantique atteint.

Dans cette optique, ce chapitre offre donc un ensemble d'outils, permettant l'extraction de divers types d'objets dans une image. Ces outils se révéleront indispensables par la suite, pour l'élaboration de la partie relationnelle de la structuration. En ce sens, et dans la poursuite des travaux présentés jusqu'ici, ils partagent à nouveau la qualité d'être simples à mettre en œuvre et de fournir des résultats satisfaisants, pour leur utilisation dans la structuration relationnelle. Cependant, s'ils nous permettent d'ores et déjà d'établir des relations entre les diverses entités d'un document, comme cela sera détaillé au chapitre suivant, nous verrons qu'ils restent à inscrire au nombre des perspectives d'améliorations possibles de nos travaux.

Avant de décrire plus précisément l'organisation des différentes sections, nous souhaitons mettre en lumière dès à présent le fait non négligeable que les outils de classification mis en œuvre peuvent également être appliqués aux régions issues de la segmentation spatiale obtenue au chapitre précédent. L'extraction d'objets et la classification que nous proposons sont en effet réalisées par la recherche de certaines primitives de couleur, texture et forme qu'il est tout à fait envisageable d'appliquer aux régions de la segmentation. Nous en donnerons par ailleurs quelques exemples au cours du développement de ce chapitre.

L'analyse de telles primitives, n'est en outre pas une idée nouvelle puisque déjà présente

dans de nombreux travaux [41, 76, 93]. L'originalité de notre démarche se situe plutôt dans la nature des outils développés, reposant dans leur majorité sur des filtrages morphologiques simples des images.

Mais revenons à l'organisation concrète de ce chapitre et à la présentation des divers outils. Nous commençons par un premier outil d'extraction d'incrustations ou de bandeaux de forme rectangulaire à base d'opérateurs morphologiques (section 6.2), suivi d'un deuxième outil d'extraction de texte cette fois-ci, reposant toujours sur une succession de filtrages morphologiques des images (section 6.3). Vient alors une technique d'étude spectrale des objets (section 6.4) : on propose ainsi deux exemples d'applications de ce genre d'outil, tout d'abord la classification de diverses régions en tant que végétation, ciel ou eau, débouchant sur un début de distinction entre scènes extérieures et scènes intérieures, puis l'extraction de zones correspondant à de la peau humaine, et ceci quelle que soit sa couleur.

Enfin, avant de procéder à leur description proprement dite, soulignons que la liste que nous venons de fournir n'est pas exhaustive, loin de là. D'autres objets pourraient ainsi être extraits et donner lieu à une complémentation de la structure relationnelle que nous proposerons au chapitre 7. Cependant, soulignons que tous les outils développés ici, s'ils reposent tous sur des traitements d'images simples et de bas niveau, permettent d'ores et déjà, par leur combinaison et l'ajout de règles de décision, d'atteindre un niveau sémantique élevé, dont l'exemple le plus complexe sera fourni au travers de la dernière application du chapitre 8.

Procédons à présent à l'exposé de ces outils.

6.2 Extraction d'incrustations et de bandeaux

Dans le cadre de l'application un peu particulière que nous nous sommes fixés, i.e. les journaux télévisés, l'incrustation fait partie des objets possédant un caractère sémantique que l'on se doit de chercher à extraire des images. Une incrustation est généralement de deux types ; il s'agit soit d'un bandeau de texte, soit d'une image. En règle générale, ces incrustations sont significatives, pour le bandeau de texte, d'informations supplémentaires sur les noms et qualités de la personne filmée, lorsqu'elle existe, ou sur le reportage lui-même. Les incrustations d'images, que nous appeleront par la suite imagettes, ont, elles, bien souvent lieu à l'intérieur d'une prise de vue de présentateur et ont un objectif double : illustrer les propos du présentateur et donner une première introduction au prochain sujet de reportage. Pour ces deux types d'incrustations (texte et image), les raisons motivant leur détection sont donc fondées.

Les caractéristiques géométriques sur lesquelles baser notre détection sont par ailleurs relativement évidentes. En effet, les bandeaux comme les imagettes sont généralement de forme rectangulaire. Cette caractéristique limite donc l'espace de recherche de telles incrustations, qui par contre peuvent être situées un peu partout dans l'image, avec une restriction pour la place centrale, réservée au sujet principal de l'image (les incrustations ne sont en effet que de l'information supplémentaire).

De par leur utilisation d'un élément structurant comme forme comparative, les outils morphologiques sont tout particulièrement bien adaptés à l'extraction de formes spécifiques dans les images. Ainsi l'algorithme d'extraction d'incrustations et de bandeaux est-il composé de plusieurs étapes de filtrage morphologique par deux éléments structurants de forme dédiée : un carré et un segment vertical ou horizontal. Pour une description et un rappel des définitions des outils morphologiques employés, le lecteur se reportera à l'annexe B.

Algorithme 4 Algorithme de détection d'incrustations dans une image clé.

Entrée :

I_i image clé

$dimX$ et $dimY$ les dimensions en X et Y de I_i

Extraction des contours verticaux et horizontaux

Calcul de G_i, gradient morphologique par un élément structurant carré en 4-connexité

Etape de filtrage des contours

Calcul de OH_i, ouverture horizontale de taille $(dimX/16)$ de G_i

Calcul de OV_i, ouverture verticale de taille $(dimY/16)$ de G_i

Supremum O_i des deux ouvertures OH_i et OV_i

Etape de bouchage des rectangles extraits

Reconstruction R_i de O_i à partir d'un point au milieu de l'image

Ouverture par reconstruction R_i' de R_i par un élément structurant carré de taille $\min(dimX/16, dimY/16)$

Recherche des maxima de forme rectangulaire

Extraction des maxima M_i de R_i'

Ouverture de taille $\min(dimX/16, dimY/16)$ de M_i

Reconstruction R_i'' de M_i à partir des résidus de l'ouverture

Obtention du masque des incrustations

Calcul de $Incrust_i$ le masque des incrustations comme le résidu de la reconstruction R_i''

Sortie :

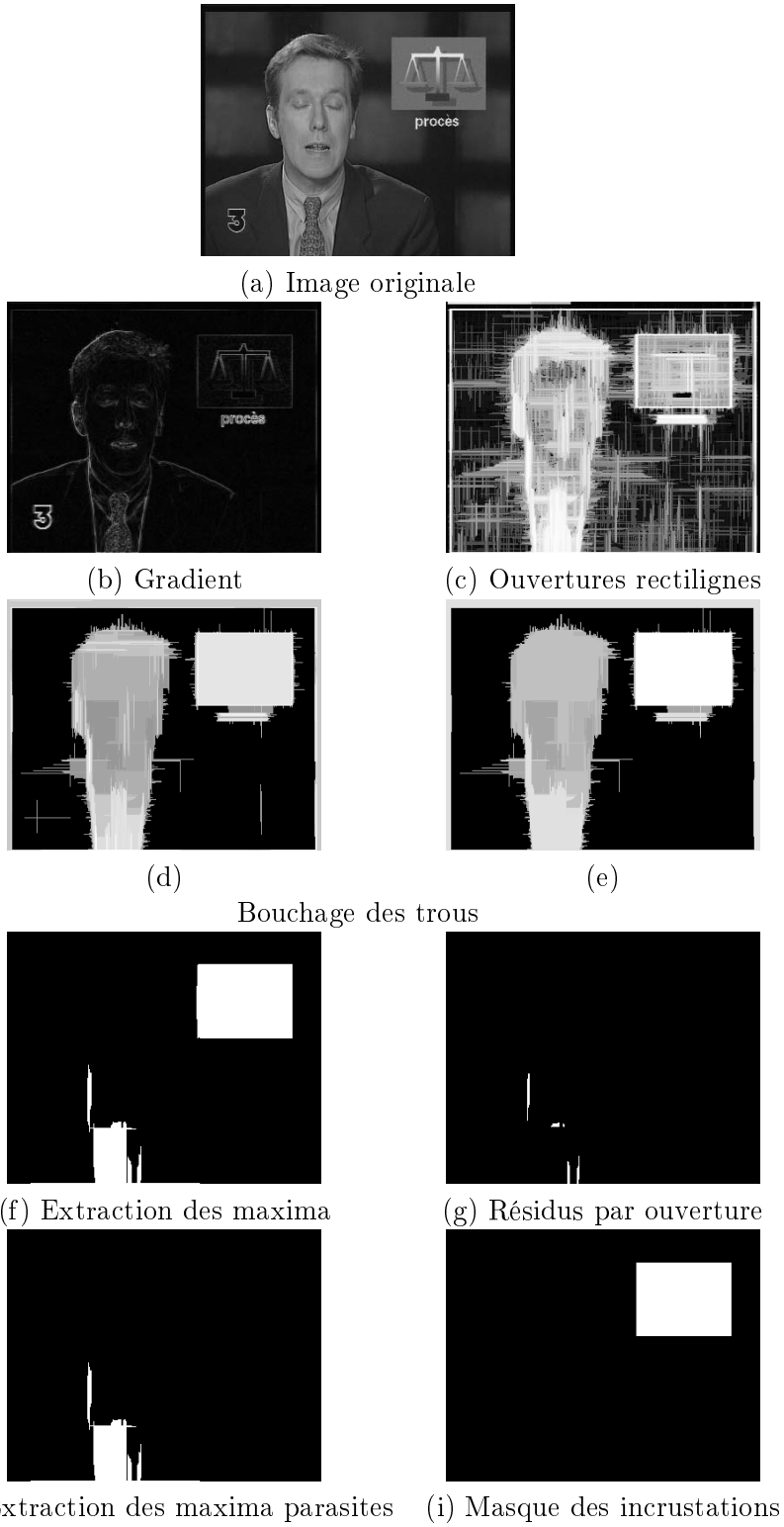
$Incrust_i$ masque des incrustations

Le processus complet d'extraction des incrustations est proposé dans l'algorithme 4. Nous en détaillons à présent les grandes lignes avant de fournir des exemples de résultats obtenus sur des images clés des documents vidéo dont nous disposons. Ainsi que nous l'avons déjà évoqué en début de paragraphe, les incrustations que nous cherchons à extraire se distinguent dans l'image, de par leur forme rectangulaire. La première étape de l'algorithme consiste donc à extraire de l'image tous les contours rectilignes. Ceci est réalisé en deux temps, tout d'abord par le calcul d'un gradient morphologique classique sur la luminance de l'image uniquement, puis par un filtrage de ces contours par deux ouvertures horizontale et verticale de grande taille, de façon à ne conserver que les contours rectilignes. A ce stade, notons que la taille de l'ouverture choisie est un paramètre de l'algorithme dont la valeur est fixée automatiquement pour chaque séquence, en fonction des dimensions de l'image originale. Le choix initial de cette taille d'ouverture (i.e. les tailles de l'image originale divisées par 16) a été fixé par expérience pour correspondre à la taille relative moyenne d'une incrustation dans une image clé. Cette première étape d'extraction des contours rectilignes est illustrée sous la forme des deux images (b) et (c) de la figure 6.1, à partir de l'image originale présentée en (a).

Une fois ces contours extraits, on procède à une étape de bouchage des formes rectilignes apparues. Cette étape est également effectuée en deux temps, par deux reconstructions successives : l'une, à partir du point situé exactement au milieu de l'image (image (d), figure 6.1) ; et l'autre, de l'image résultante par son ouvert (image (e), figure 6.1). Le point situé au milieu de l'image a en effet une forte probabilité de se trouver en dehors de toute incrustation, généralement plutôt excentrée. On obtient donc ainsi un bouchage des incrustations, y compris lorsqu'elles sont situées sur les bords de l'image : elles apparaissent alors également comme des contours fermés. La deuxième reconstruction a pour but d'applanir les niveaux de gris à l'intérieur des incrustations, de façon à créer des plateaux qu'il est alors facile d'extraire, lors de la troisième étape.

Cette troisième et dernière étape consiste tout d'abord en l'extraction des maxima de la reconstruction précédente, ce qui fournit l'image (f) de la figure 6.1. Puis de ces maxima, nous ne cherchons à conserver que ceux correspondant à des formes rectangulaires parfaites. Ceci est obtenu par détection des maxima plus tourmentés : on calcule le résidu de l'ouvert (image (g)), c'est-à-dire la différence original - ouvert, par un élément structurant carré de grande taille, puis on reconstruit l'image des maxima par ce résidu. On obtient ainsi l'image (h). Le masque des incrustations, illustré dans l'image finale (i), est, quant à lui, obtenu par simple différence. Notons que la robustesse de cette étape pourrait être améliorée. Il n'est en effet pas toujours évident que les incrustations apparaissent comme des maxima de formes rectangulaires parfaites : il est possible que de petites barbules subsistent, résultant en la non-détection de ces incrustations par le résidu de l'ouverture. Cependant si des barbules peuvent être présentes, il est clair qu'elles doivent être moins importantes et moins nombreuses que pour un maxima de forme quelconque. Un léger filtrage morphologique supplémentaire avant de calculer le résidu de l'ouverture devrait donc permettre d'augmenter la robustesse de cette étape sans pour autant la complexifier.

L'algorithme que nous venons de décrire possède l'avantage de ne reposer que sur des traitements morphologiques simples, que l'on sait à l'heure actuelle optimiser, ou programmer sur des architectures dédiées. Il ne repose en outre sur aucun paramètre. Trois uniques hypothèses forment sa structure sous-jacente : les incrustations sont de forme rectiligne presque parfaite, jamais au centre de l'image et ont une taille minimale de $1/16^e$ des images originales. Appliquée à l'ensemble des images clés extraites lors de la structuration linéaire, la méthode développée a prouvé son efficacité : les incrustations, y compris les bandeaux intersectant le



(a) Image originale

(b) Gradient

(c) Ouvertures rectilignes

(d)

(e)

Bouchage des trous

(f) Extraction des maxima

(g) Résidus par ouverture

(h) Extraction des maxima parasites

(i) Masque des incrustations

FIG. 6.1: Processus de détection des incrustations dans une image clé, par une succession de traitements morphologiques.

bord de l'image, sont correctement détectées et très peu de fausses alarmes ont lieu, du fait du critère très sélectif d'incrustations parfaitement rectangulaires. Ces fausses alarmes sont en outre pour une grande partie facilement éliminables du fait de la surface de l'incrustation alors détectée, qui correspond à un pourcentage trop important de l'image originale. Nous présentons dans la figure 6.2 des exemples de telles détections.

L'outil que nous venons de présenter reste cependant à inscrire au titre des améliorations futures. Si le taux de fausses alarmes est bas, le taux de détection d'incrustations réelles pourrait quant à lui être bien supérieur et plusieurs situations ne donnent lieu à aucune détection d'incrustation. C'est par exemple le cas lorsqu'un bandeau est situé de telle façon qu'il coupe l'image en deux. L'étape de bouchage de trous mène dans ce cas à une détection correcte de ce bandeau, mais additionné de toute la partie de l'image qui n'a pu être atteinte par le bouchage de trous. Cette première situation d'échec de détection est présentée dans la figure 6.3, (a). Une autre situation plus problèmatique de non détection de l'incrustation intervient lorsque l'étape de gradient ne fournit pas de contours fermés pour l'incrustation. C'est par exemple le cas de l'image (b), toujours dans la figure 6.3, pour laquelle l'incrustation noire ne se distingue pas suffisamment des régions sombres de l'image. Les contours obtenus n'étant pas fermés, l'étape de bouchage de trous ne restitue pas les incrustations. Enfin, le choix d'effectuer un bouchage de trous à partir du point central repose sur l'hypothèse qu'une incrustation n'est jamais située à cet endroit. Si cette situation est en effet fort peu probable, il est cependant possible que le point central tombe malgré tout à l'intérieur d'une zone fermée, comme cela aurait été le cas si le présentateur de l'image (d), figure 6.1 avait été au centre de l'image. Dans ce cas le bouchage de trous produit une image entièrement blanche exceptée pour la zone centrale, et en aucun cas, ne fournit les incrustations recherchées.

Ces trois cas particuliers sont suffisamment importants pour qu'il soit donc nécessaire d'envisager d'affiner l'étape de bouchage de trous, par une étape préalable de fermeture des contours, et par une reconstruction issue non plus à partir du point du milieu de l'image, mais à partir d'autres points supplémentaires, à définir. S'il ne permet pas de résoudre les problèmes de faibles contours entre deux zones de couleurs similaires, le calcul d'un gradient directement couleur peut également améliorer les contours des incrustations, bien qu'il ne nous semble pas qu'il s'agisse là de la solution aux problèmes évoqués. Les incrustations sont en général suffisamment contrastées pour ressortir de l'image originale, qu'on la considère en couleur ou en niveaux de gris, et comme nous venons de l'évoquer les problèmes de faibles contours ne seront pas pour autant résolus.

Ces situations particulières mises à part, nous sommes à présent en mesure d'extraire les incrustations et bandeaux de forme rectangulaire des images et ceci, dans le but de lancer, sur ces zones particulières, des traitements dédiés et généralement nécessitant une plus lourde mise en œuvre. Ainsi il est envisageable de chercher à segmenter plus finement les imagettes, ou encore de détecter le texte des bandeaux. Dans ce dernier cas cependant, la détection de texte n'est pas toujours réalisable directement sur la zone correspondant au bandeau, sans une étape supplémentaire de filtrage préalable. Le texte n'est en effet pas toujours sur un fond de couleur uniforme, qu'il soit d'ailleurs à l'intérieur d'un bandeau ou directement dans l'image. Dans ces situations, l'application directe d'un logiciel de reconnaissance optique de caractères (ROC) est impossible. Nous proposons donc dans la section suivante un second outil d'extraction de zones textuelles.

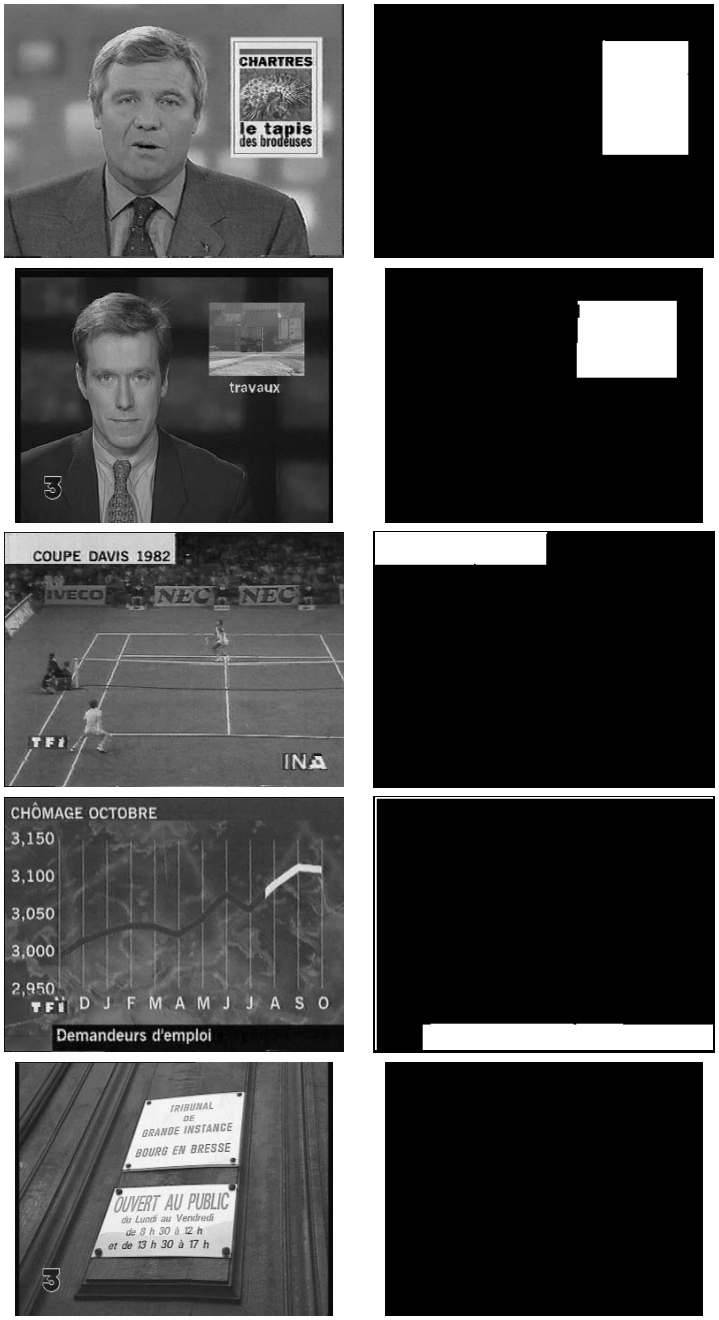


FIG. 6.2: Exemples de résultats de détection d'incrustations sur des images clés. On fournit pour chaque image originale, le masque des incrustations. Lorsque ce masque est totalement noir, aucune incrustation n'a été détectée.

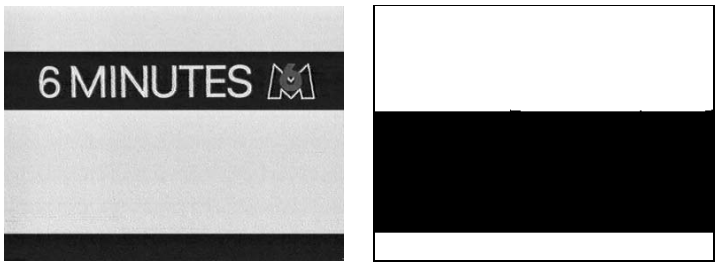


Image originale Masque des incrustations

(a)

Image originale Gradient

(b)

FIG. 6.3: Cas particuliers d'échec de détection des incrustations : (a) le bandeau sépare l'image en deux parties distinctes ; (b) le gradient ne produit pas de contours fermés.

6.3 Extraction de zones textuelles

L'outil détaillé ici possède un domaine d'application bien défini. Nous cherchons à extraire les zones de texte uniquement lorsque celui-ci est véritablement en situation d'être lu par le spectateur. Cette restriction, toujours dans le cadre de notre application aux journaux télévisés, regroupe malgré tout l'ensemble du texte, à caractère sémantique pour notre application d'indexation, présent dans les images. En effet, seul le texte suffisamment bien contrasté, d'une taille conséquente et relativement horizontal ou vertical peut fournir une information sémantique rapidement au spectateur. Ces trois contraintes de contraste, de taille et d'orientation des caractères s'appliquent en outre en règle générale à la majorité des bandeaux de texte.

En comparaison des nombreux travaux existants dans le domaine de détection de texte (le lecteur se reportera à [89, 20], pour quelques références et états de l'art), la méthodologie que nous décrivons ici fait suite aux travaux développés dans [16] et s'appuie essentiellement sur des opérateurs morphologiques (cf. annexe B), de façon à construire encore une fois un outil très simple. Le but de cet outil est en outre restreint à une détection des zones où, par la suite, il serait intéressant de lancer un moteur de reconnaissance de caractères. Nous ne visons par contre pas la réalisation de cette reconnaissance. Idéalement le produit final de l'algorithme de détection de texte que nous présentons est une image très nettoyée où seuls les caractères de texte sont présents, correspondant aux contraintes que nous nous sommes fixées.

Le cadre d'obtention de l'opérateur étant posé, nous en proposons une description dans l'algorithme 5, restreint à la détection des caractères plutôt clairs dans les images. Un processus dual est mis en œuvre pour les caractères foncés, par utilisation d'un chapeau haut-de-forme noir au lieu d'un chapeau haut-de-forme blanc en première étape.

Cet algorithme se décompose en trois étapes distinctes, dont la seconde est la plus cruciale.

Algorithme 5 Algorithme de détection des caractères clairs contrastés, horizontaux et d'une certaine taille.

Entrée :

I_i image clé

$dimX$ et $dimY$ les dimensions en X et Y de I_i

Pré-condition : n l'épaisseur maximale d'un caractère

Extraction des petits détails clairs

Calcul de C_i, chapeau haut-de-forme blanc par reconstruction par un élément structurant carré de taille n de I_i

Obtention d'une zone de détails horizontaux

Calcul de FH_i, fermeture horizontale de taille n de C_i

Calcul de OH_i, ouverture horizontale de taille $3n$ de FH_i

Bouchage des trous éventuels B_i par reconstructions de OH_i

Calcul de R_i, fermeture par reconstruction rectiligne verticale de taille $n/2$ de B_i

Dilatation isotrope D_i de taille $n/2$ de R_i

Extraction des caractères clairs dans les zones horizontales extraites

Calcul de l'infimum F_i entre C_i et D_i

Sortie :

l'image F_i contenant essentiellement les caractères clairs extraits.

On commence par extraire l'ensemble des détails fins et clairs dans l'image, par un chapeau haut-de-forme de taille n, sur l'image de luminance. Le choix de travailler en niveaux de gris est à nouveau guidé par le fait que les caractères doivent se distinguer suffisamment pour qu'une extraction en luminance suffise. Le paramètre n, le seul de l'algorithme est à choisir en fonction de l'épaisseur maximale des traits des caractères à extraire : tout caractère d'épaisseur supérieure ne sera pas détecté lors de cette première étape. En pratique nous avons choisi une taille de 10 pixels, ce qui correspond déjà à des caractères relativement larges, pour des images à pleine résolution. La seconde étape, qui comme nous l'avons dit est la plus délicate, consiste à regrouper sous forme de bandes horizontales les détails extraits. Ceci correspond bien sûr au passage du caractère au mot de texte. Tout détail extrait précédemment par le chapeau haut-de-forme et qui ne peut être inclus dans une telle région est alors éliminé.

L'extraction de ces régions commence par une fermeture horizontale de taille n de façon à connecter les détails clairs entre eux, suivie d'une ouverture de taille $3n$, ceci pour supprimer les trop petites régions ainsi créées. Il est sous-entendu à ce stade que le texte à extraire est composé de plus de trois caractères à la suite. Les mots d'une taille inférieure ne sont donc détectés que s'ils appartiennent à une ligne de texte plus longue. On poursuit cette extraction par un bouchage des trous éventuels dans chaque région, puis par une fermeture par reconstruction verticale de taille $n/2$, qui a pour but, comme lors de l'extraction d'incrustation, d'égaliser les niveaux de gris à l'intérieur d'une même région. Enfin cette extraction se termine par une dilatation isotrope de taille $n/2$ de façon à englober parfaitement les caractères contenus dans la région extraite.

Les régions correspondant aux critères de zones de texte étant à présent détectées, il reste à conserver uniquement les caractères extraits par chapeau haut-de-forme et situés à l'intérieur de ces régions, tous les autres résidus du chapeau étant, par là-même, éliminés.

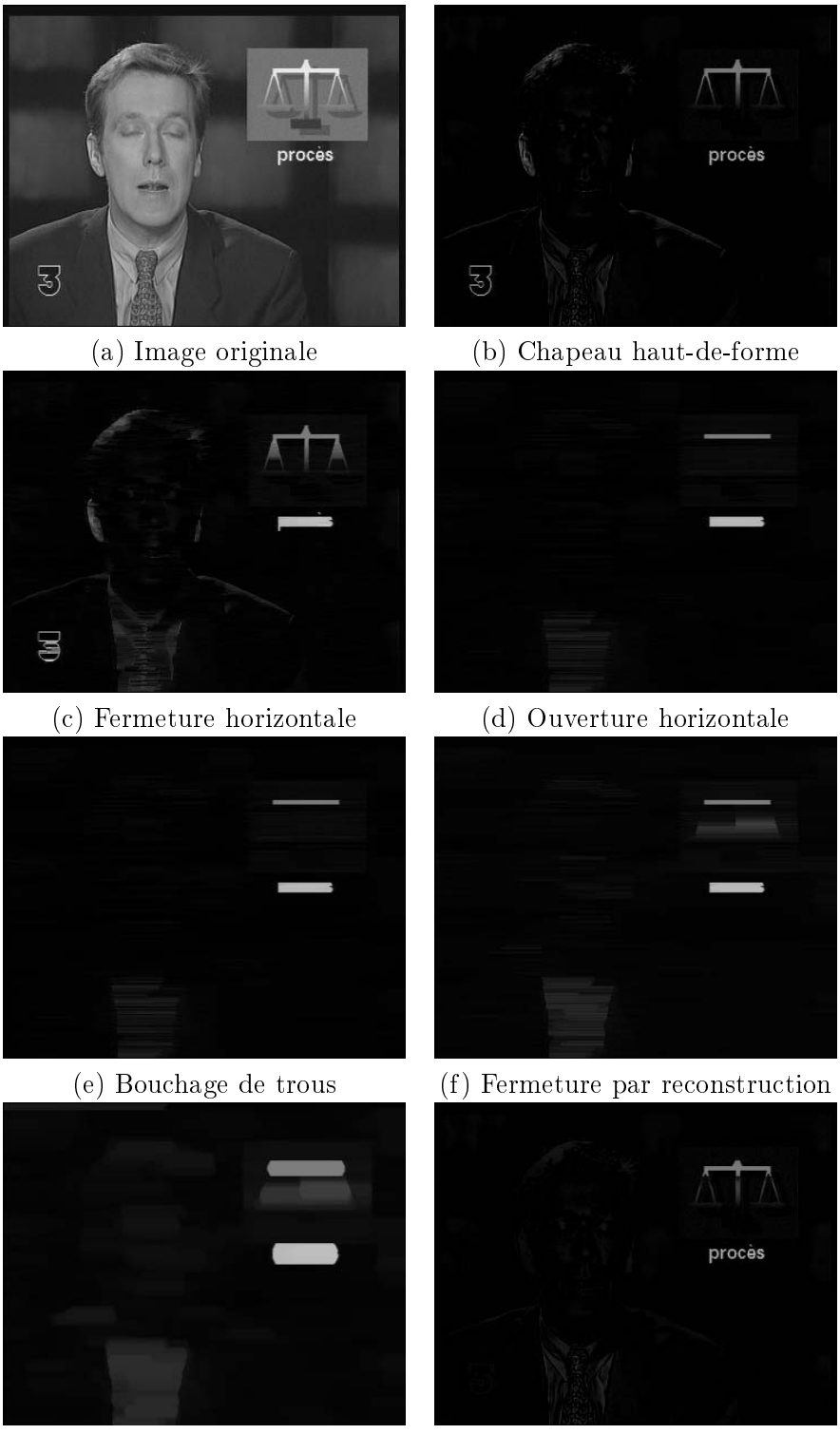
Nous présentons respectivement dans les figures 6.4 et 6.5, les différentes étapes de cette détection de texte sur une image clé donnée, ainsi que plusieurs autres résultats de détection correcte et d'oubli de texte. D'autres tests menés sur des images ne contenant aucun texte résultent par ailleurs - et heureusement ! - en une absence de détection.

Les images finales obtenues peuvent alors facilement être seuillées de façon à obtenir une image binaire, qui est alors fournie en entrée d'un logiciel de reconnaissance optique de caractères (ROC) classique[1], avec les résultats respectifs suivants :

- image (a), figure 6.4 : proces
- image (a), figure 6.5 : Jean Godfroitl, p'refet de la Dr-omg
- image (b), figure 6.5 : Bourg-en-Brosse Ain
- image (c), figure 6.5 : pas de détection
- image (d), figure 6.5 : EURO TNBIIEL LE TEST
- image (e), figure 6.5 : pas de détection

Du fait de l'orientation des caractères, l'image (c) ne donne pas lieu à une détection satisfaisante avec le logiciel utilisé, en dépit de la qualité de l'image d'entrée. L'image (e) par contre associe deux inconvénients : une piètre qualité de l'image de départ et une faible résolution,

[1] Il s'agit du logiciel WOCAR, disponible gratuitement à l'adresse "http://persoweb.francenet.fr/~cambien". Notons tout particulièrement que ce logiciel fait appel à une connaissance de la forme des polices de caractères. Or celles utilisées dans les journaux télévisés ne correspondent pas forcément à celles de la base de données de WOCAR.



(a) Image originale (b) Chapeau haut-de-forme

(c) Fermeture horizontale (d) Ouverture horizontale

(e) Bouchage de trous (f) Fermeture par reconstruction

(g) Dilatation isotrope (h) Infimum

FIG. 6.4: Etapes de la détection de zones de texte dans une image clé.

FIG. 6.5: Exemples de détection de texte, sur diverses images clés. On présente successivement l'image originale, le résultat de l'algorithme et un seuillage de ce résultat, utilisé comme entrée d'un ROC.

qui résultent également en une absence de reconnaissance du texte incrusté. Notons cependant, pour chacun des autres exemples, le niveau de détection tout à fait correct obtenu par l'algorithme proposé, alors que le logiciel de reconnaissance de caractères appliqué directement sur les images brutes sans notre étape de filtrage préalable ne fournit aucun résultat.

Au titre des avantages de l'algorithme que nous venons de présenter, citons encore une fois sa grande simplicité et son automatisme complet, quelle que soit la taille de l'image d'entrée (le seul paramètre de largeur maximale des caractères étant fixé à l'avance, et ce, en fonction des contraintes de la catégorie de texte que nous cherchons à extraire).

Comme pour l'outil de détection d'incrustations, certaines étapes de la méthode nécessitent certainement des améliorations. Il serait par exemple appréciable d'ajouter une étape d'égalisation du fond, avant le chapeau haut-de-forme initial dans les situations, malgré tout assez courantes, où le texte se trouve à cheval sur deux zones différentes, créant ainsi des paliers de contraste (cf. image (a), figure 6.5). L'utilisation d'un chapeau haut-de-forme inf (cf. section 3.2.5.1) comme première étape d'extraction des détails clairs constitue alors une piste possible d'amélioration. Un tel outil permet en effet d'extraire le dénivelé maximal entre un point et tous ses voisins. Dans l'exemple de la figure 6.5 (a), le texte possède un effet d'ombrage. L'application du chapeau haut-de-forme inf restituerait dans ce cas, pour toute la partie de texte sur fond clair, le dénivelé dans la zone d'ombre, réduisant ainsi l'inhomogénéité du fond.

Cependant, le schéma de traitement établi a, dès à présent, prouvé son efficacité, tout en restant dans nos contraintes de simplicité de l'outil élaboré. En l'état actuel, en plus de fournir des mots clés utiles à l'indexation du document, il constitue en outre déjà une base concrète pour l'établissement de relations entre prises de vue contenant le même texte, sur le modèle de l'algorithme de mise en relation par persistance d'incrustations que nous proposons au paragraphe 7.3.2.

La présentation de ce deuxième outil d'extraction d'objets étant à présent terminée, nous poursuivons ce chapitre par un troisième et dernier outil, également source d'informations intéressantes, permettant l'étude spectrale de diverses régions.

6.4 Outil d'étude spectrale

L'objet de cette section est de fournir la base d'un outil, capable de classifier sans ambiguïté certaines régions ou pixels d'une image comme appartenant à une zone de végétation, ou à un visage, ou à tout autre objet. Cet outil repose sur une expression et une étude de la teinte des régions ou pixels. En ce sens, la technique développée ne propose pour l'instant qu'une classification des diverses régions en fonction de leur teinte, plutôt bleue, verte ou couleur peau, et que par extension nous associons à des étiquettes à caractère sémantique : ciel, végétation, peau. Un tel étiquetage ne se justifiera pleinement qu'avec l'ajout d'autres caractéristiques de texture, forme ou autre, dans une perspective de perfection de l'outil.

A ce stade de notre étude, nous nous cantonnons à l'espace des teintes, au demeurant déjà source d'informations pertinentes pour l'établissement de relations (cf. chapitre 7). Notons que le terme d'"étude spectrale" est alors un abus de langage, destiné à faciliter le discours, abus de langage que nous poursuivrons par l'emploi du mot spectre, notamment dans la section 6.4.2, pour désigner l'histogramme des teintes, en référence à l'interprétation physique de la couleur comme une longueur d'onde.

Dans la suite de cette section, nous commençons par expliciter la méthodologie employée

dans le paragraphe 6.4.1. Puis nous illustrons l'utilisation qu'il est possible d'en faire dans le cadre de deux applications (sections 6.4.2 et 6.4.3), avant de conclure sur les améliorations à apporter et d'élargir le champ des applications.

6.4.1 Méthodologie : couleur dominante et étiquette

Avant d'expliciter ce que nous mettons derrière les notions de couleur dominante et d'étiquette, il est important de souligner que la dualité existant entre une région d'une segmentation et un objet directement extrait s'applique ici encore. En effet la méthodologie développée prend deux directions distinctes suivant que l'on cherche à extraire des caractéristiques de teinte d'une région donnée d'une segmentation (et donc regroupant à l'origine des pixels déjà homogènes suivant un critère donné), ou suivant que l'on vise l'extraction directe d'objets comme ensemble de pixels ayant certaines caractéristiques de teinte fixées a priori.

Pour les deux démarches, l'objectif est le même : il s'agit d'arriver au final à une classification des divers objets ou régions en fonction de leur teinte.

Cependant les deux processus sont opposés dans leur ordre logique. Dans le premier cas, on part d'une région, donc d'une certaine classification, pour aboutir à un étiquetage de cette région en fonction de sa teinte, l'étiquette attribuée étant a priori inconnue. Dans le second cas, on choisit une étiquette au départ, donc un certain intervalle de teintes, et on extrait l'ensemble des pixels correspondant aux caractéristiques de cette étiquette. Le résultat fournit une ou plusieurs régions dans l'image originale.

Quant à la notion de couleur dominante, elle prend naturellement le qualificatif de couleur dominante *d'une région donnée*. Avec l'hypothèse que la segmentation réalisée est à l'origine de régions homogènes en couleur, la couleur dominante s'exprime alors simplement comme la médiane de l'ensemble des couleurs présentes dans la région étudiée. Le choix de la couleur médiane (cf. section 5.4.3) plutôt que de la moyenne permet de s'assurer que la couleur résultante est présente au départ dans la région.

Ici il est essentiel de rappeler qu'une couleur est définie par ses trois paramètres de luminance, teinte et saturation. Même si nous calculons effectivement une couleur dominante pour chaque région, l'étude spectrale que nous proposons dans ce paragraphe n'en est pas moins uniquement réservée au paramètre de teinte.

Si la notion de couleur dominante se trouve exclusivement réservée à l'usage des régions, la notion d'étiquette la remplace au niveau pixel. Cette étiquette, qu'il est donc possible d'attribuer à chaque pixel, est définie pratiquement comme un intervalle de teintes relevant de l'expérience. Ainsi les teintes comprises entre $]0, 60]$, avec la définition de la teinte de l'espace HSV, correspond à l'étiquette "couleur de la peau humaine", comme cela sera détaillé de façon plus approfondie dans la section 6.4.3.

Nous fournissons ci-dessous un résumé des deux processus mis en œuvre en pratique :

Classification d'une région. Pour une région donnée d'une segmentation, on calcule sa couleur dominante. La teinte de cette couleur dominante est ensuite comparée à l'ensemble des étiquettes, représentées sous la forme d'intervalles de teintes, de façon à extraire celle à laquelle elle correspond. L'étiquette extraite est attribuée à la région étudiée.

Détection des zones correspondant à une étiquette donnée. On représente l'ensemble des teintes des pixels de l'image sous la forme d'un histogramme des teintes. Seuls les pixels dont la teinte appartient à l'intervalle de l'étiquette de départ sont conservés dans l'image originale.

Ces deux méthodologies distinctes étant explicitées, nous fournissons à présent deux exemples d'applications aboutissant à une classification des régions en fonction de leur teinte.

6.4.2 Distinction des classes d'objets : végétation, ciel, eau

L'application développée ici se propose de montrer que l'information de teinte permet, avec quelques restrictions cependant, une première classification ou distinction entre diverses classes d'objets, et plus particulièrement les régions contenant de la végétation et les zones de ciel.

Si nous nous sommes restreints à la végétation et au ciel, c'est tout d'abord parce que leur teinte est caractéristique de ces deux objets. D'autre part la présence de végétation et de zones de ciel sont deux informations essentielles à la caractérisation d'une scène extérieure. La classification de ces deux catégories d'objets doit donc nous permettre de répondre à la question d'un niveau sémantique plus élevé : s'agit-il d'une scène extérieure (de jour) ?

Mais l'élaboration de cette réponse étant plus précisément détaillée en fin de section, revenons à présent sur la distinction entre végétation et ciel. A partir de l'image fournie dans la figure 6.6, (a), il apparaît clairement que la teinte diffère lorsqu'on se trouve dans une zone de végétation ou dans une zone de ciel. Il suffit pour s'en persuader de tracer les histogrammes des teintes correspondant à deux régions de végétation et de ciel extraites manuellement (notons l'étendue de la gamme des teintes représentée, allant de 0 à 360 puisqu'il s'agit d'un angle). Ces deux histogrammes sont également présentés dans la figure 6.6, images (b) et (c). Pour chacun, les zones concernées du spectre des teintes sont très différentes.

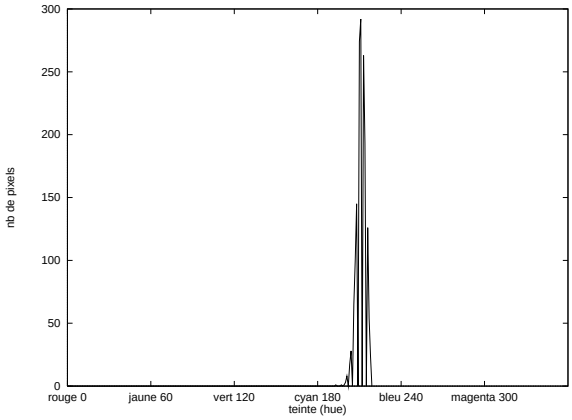
L'information de teinte n'est cependant pas suffisante dans d'autres cas. Il est ainsi impossible, sans information supplémentaire différente de la teinte, de distinguer a priori une zone de ciel d'une zone d'eau (comparer les spectres (b) et (d) de la figure 6.6). Citons en premier lieu, au nombre des informations supplémentaires, la texture des régions étudiées, ou bien leur forme [47].

Nous présentons dans la figure 6.7 les résultats de l'outil d'étude spectrale, appliqué aux régions issues d'une segmentation de l'image originale proposée dans la figure 6.6, (a). Cette segmentation, obtenue par l'algorithme discuté dans le chapitre 5, ainsi que le résultat des étiquetages respectifs en tant que régions de ciel ou de végétation, sont fournis dans les images (a), (c) et (d) de la figure 6.7. On propose également les valeurs des couleurs dominantes calculées pour chaque région dans l'image (b).

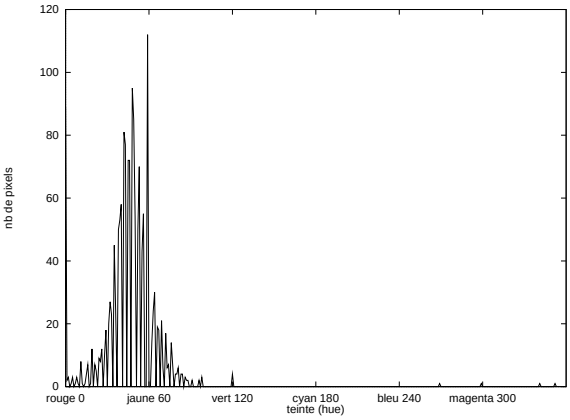
Nous sommes donc à présent en mesure de distinguer certaines régions d'une image à partir de leur réponse spectrale uniquement. Et même si cette information n'est pas suffisante, comme par exemple lors de la distinction ciel/eau, nous nous sommes basés uniquement sur cet outil pour développer une méthode d'analyse d'une image nous permettant de répondre à la question de présence d'une scène extérieure, ou du moins d'obtenir un premier indice de réponse à cette question. D'autres critères, développés dans le cadre d'une poursuite de ces travaux, tels que le positionnement des ombres et des sources de lumière [13, 77], et fusionnés à ce premier élément de réponse, permettront sans aucun doute un renforcement de la décision prise.

Ce premier critère, que nous nous proposons de détailler, repose sur la constatation que, pour des scènes extérieures naturelles et de jour, les couleurs dites "froides", i.e. correspondant à de faibles longueurs d'ondes (bleu), se trouvent en général dans le haut des images, alors que les couleurs chaudes (rouge, marron, jaune) se rencontrent majoritairement dans le bas des images. La vérification de cette hypothèse sur une image donnée indique que la probabilité

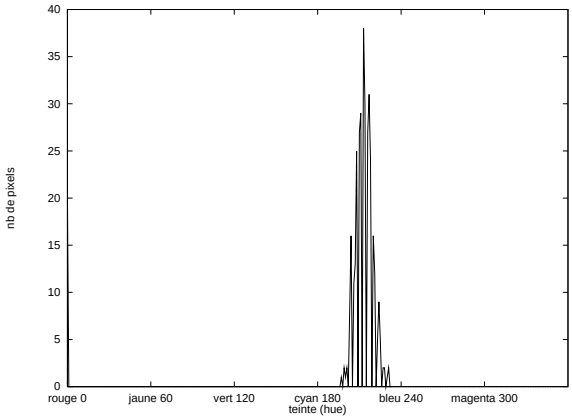
(a) Image originale ©KODAK



(b) Spectre de la zone de ciel

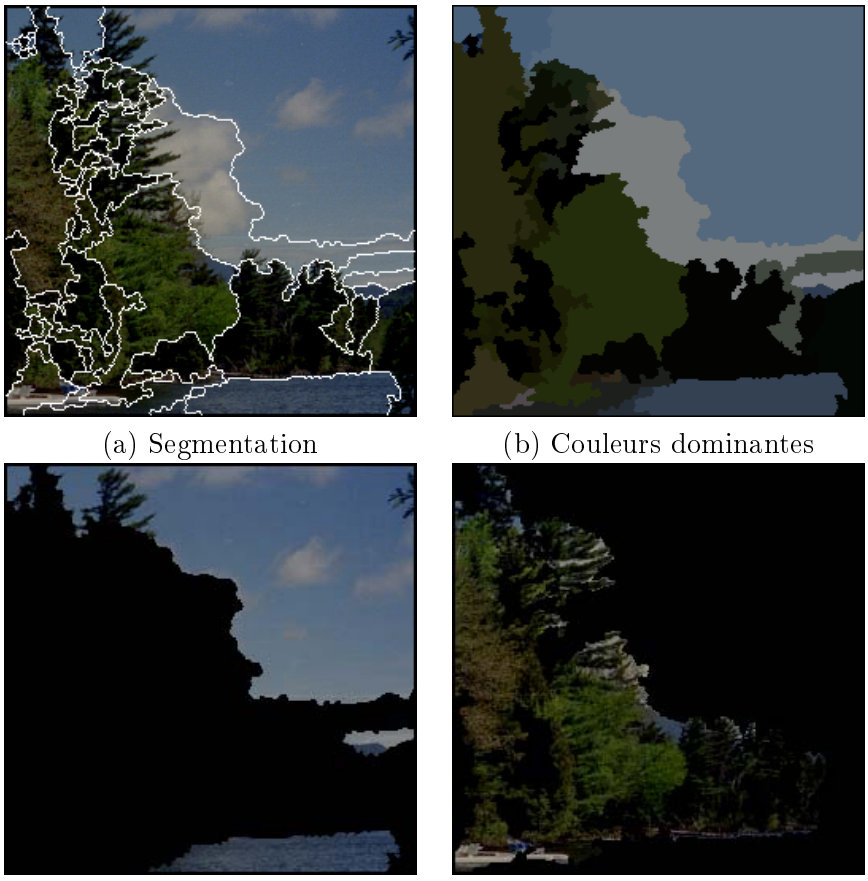


(c) Spectre de la végétation



(d) Spectre de l'eau

FIG. 6.6: Etude du spectre de différentes régions d'une même image.



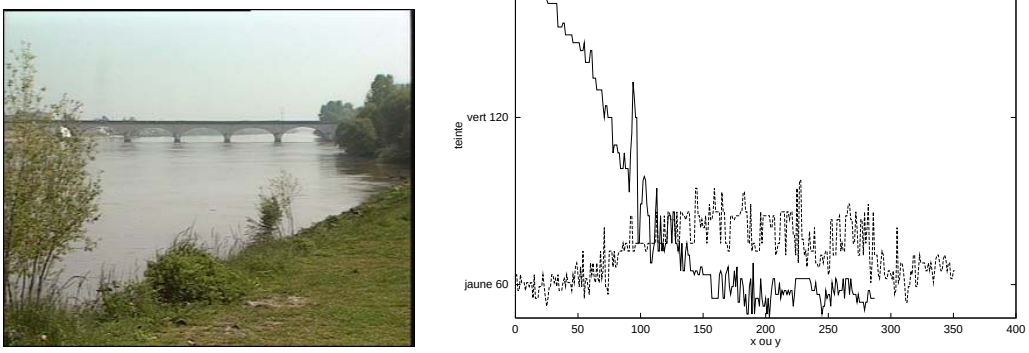
(a) Segmentation (b) Couleurs dominantes

(c) Régions étiquetées ciel (d) Régions étiquetées végétation

FIG. 6.7: Résultat de l'étude spectrale appliquée aux régions d'une segmentation. Notons l'erreur d'étiquetage de la zone de lac, l'eau ne pouvant être distinguée du ciel uniquement par une étude de sa teinte. Il convient également de remarquer l'intrusion de la cime de sapins dans la zone de ciel. Ceci est dû à une mauvaise segmentation initiale.

d'avoir affaire à une scène extérieure est élévée. Cependant, en aucun cas, l'invalidation de l'hypothèse n'induit la présence d'une scène intérieure. Avec cette hypothèse, une analyse de la répartition des teintes du haut vers le bas dans une image fournit donc une information intéressante. Dans un premier temps, cette analyse est menée de la façon suivante. Pour chaque ligne de l'image étudiée, on calcule la couleur dominante. On trace ainsi une courbe d'évolution de la teinte de cette couleur dominante en chaque ligne, en partant du haut de l'image. Un exemple d'une telle courbe est fourni dans la figure 6.8, (b) pour l'image originale présentée en (a) dans cette même figure.

Pour comparaison, l'évolution des teintes toujours ligne à ligne, mais cette fois-ci de gauche à droite, est également proposée. Au vu de ces deux courbes, la répartition annoncée est vérifiée. Une telle répartition est en outre caractéristique de la direction verticale. En ce sens, elle est également un indice précieux de l'orientation correcte d'une image [13]



(a) Image originale (b) Evolution de la teinte

FIG. 6.8: Courbes d'évolution de la teinte de chaque ligne, du haut vers le bas, et de chaque colonne, de gauche à droite (b), pour l'image originale (a).

La décision de classifier l'image de départ comme scène extérieure est alors prise en fonction d'une évaluation du dénivelé des teintes lors du parcours haut-bas. Cette évaluation est pour l'instant pour le moins grossière puisqu'elle repose sur la variation d'amplitude entre le premier point et le dernier point de la courbe d'une part (ce que nous appelons *amplitude relative*) et sur la variation d'amplitude entre le maximum et le minimum de la courbe (i.e. *amplitude absolue*), d'autre part.

La décision de présence d'une scène extérieure est alors conditionnée à la vérification des deux inéquations suivantes :

$$|Amplitude\_Relative| > \frac{2}{3} \times Amplitude\_Absolue \quad (6.1)$$

$$Amplitude\_Absolue > Seuil\_Dénivelé \quad (6.2)$$

où *Seuil_ Dénivelé* permet de définir un dénivelé minimal, fixé par expérience à la valeur 60 qui correspond à un changement de teinte dans l'espace HSV. L'exemple de la figure 6.8 est ainsi classifié en scène extérieure.

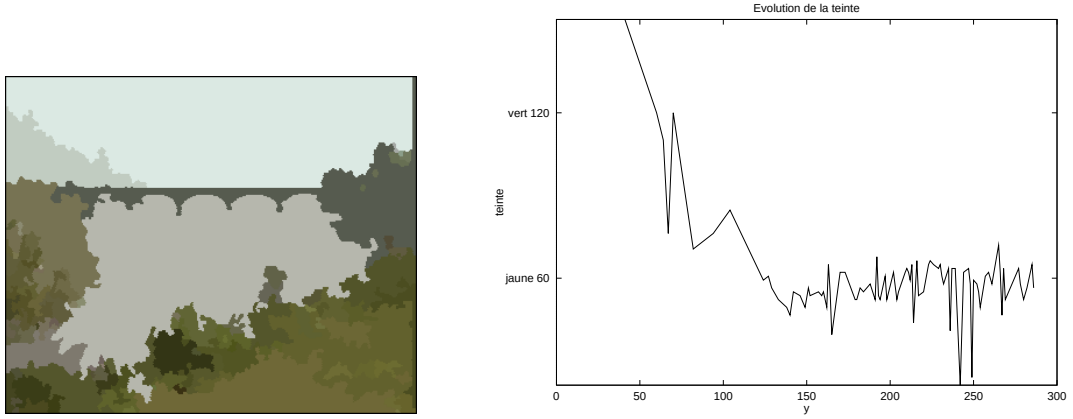
L'étude de l'évolution de la teinte ligne par ligne implique le calcul d'une couleur médiane pour chaque ligne de l'image, parmi des couleurs qui ne sont pas forcément proches les unes

de autres. En effet ce découpage pour le moins arbitraire ne tient à aucun moment compte des régions existant dans l'image de départ. Bien sûr dans la plupart des cas la couleur médiane calculée correspond à une couleur majoritairement présente sur la ligne courante et la plus proche des autres couleurs de cette ligne. Cependant il peut arriver que la couleur médiane sélectionnée, si elle correspond toujours à une couleur représentée, ne soit pas une couleur représentative d'une population de couleurs majoritaire dans la ligne étudiée, du fait du mélange de plusieurs populations différentes. Pour cette raison, nous avons choisi d'améliorer l'algorithme précédent en effectuant toujours une étude des teintes, mais cette fois-ci suivant les régions issues d'une segmentation préalable de l'image. Le découpage n'étant plus arbitraire, les couleurs médianes extraites devraient ainsi être plus représentatives des couleurs de chaque région. La segmentation utilisée découle bien sûr de l'algorithme proposé au chapitre 5.

L'évolution de la teinte devant toujours être étudiée dans la direction haut-bas, nous proposons de relier chaque région à l'ordonnée de son barycentre de façon à obtenir une classification adéquate. Pour chaque région, on calcule donc la teinte médiane et l'ordonnée de son barycentre. L'ensemble des points (ordonnée du barycentre, teinte médiane) constituent une courbe d'évolution des teintes qu'il reste à étudier de la même façon que précédemment.

Le choix de prendre l'ordonnée du barycentre de chaque région de façon à déterminer l'ordre de répartition haut-bas de ces régions peut cependant être à l'origine d'indécision : deux régions différentes peuvent avoir un barycentre de même ordonnée. Dans ce cas on effectue une première comparaison de leur taille, la plus grande région étant conservée, au détriment de la plus petite. Et dans le cas extrême où les deux régions concurrentes sont de la même taille, on conserve, de façon arbitraire, la région de valeur de teinte la plus élevée.

Pour l'image originale de la figure 6.8, on présente successivement dans la figure 6.9 la segmentation de départ, chaque région étant représentée par sa couleur médiane, et la courbe d'évolution des teintes extraites. La validation des équations 6.2 donne également lieu à la classification de l'image de départ en scène extérieure.



(a) Segmentation et couleurs médianes (b) Evolution des teintes

FIG. 6.9: Courbe d'évolution des teintes, région par région et de haut en bas (b), en fonction de l'ordonnée y de leur barycentre, à partir de la segmentation proposée en (a).

L'étude de la répartition des teintes région par région, si elle procure des valeurs de teintes médianes plus représentatives des divers objets présents dans les images, est cependant dépendante d'une segmentation correcte en régions. Cette étape de segmentation et le calcul des

couleurs médianes de chaque région sont en outre plus longs que l'étude ligne par ligne. Dans les deux cas cependant, l'évolution des teintes dans la direction verticale d'une image fournit un premier indice important sur la présence d'une scène extérieure.

Ce premier élément de réponse à une question d'un niveau sémantique déjà élevé est une première preuve de l'importance et de la validité d'une étude spectrale des divers pixels ou régions d'une image, dans un processus de structuration. Nous en proposons un deuxième exemple dans le paragraphe suivant sous la forme d'un outil de détection de régions de couleur peau.

6.4.3 Distinction de la couleur peau

Il s'agit ici de continuer l'étude spectrale d'une image donnée de façon à distinguer dans cette image les pixels de couleur peau. En aucun cas nous n'aboutissons ici à un détecteur de visages, par exemple. De fait, au vu du critère appliqué (sélection de certaines teintes), le résultat final obtenu est un ensemble de pixels ayant une teinte similaire à celle de la peau humaine. Ces pixels constituent donc un sur-ensemble des pixels appartenant à une région de peau.

Concrètement cependant, cette étude spectrale de la peau servira de base à un outil de détection de visage dans le cadre de deux applications de détection d'interviews 8.3 et de présentateur de journaux télévisés 7.4.2 que nous détaillerons dans la suite de ce mémoire.

Le processus utilisé pour extraire les pixels de teinte proche de celle de la peau humaine est le même que celui précédemment décrit dans le cadre de l'extraction de zones de végétation ou de ciel. Il repose sur la constatation (cf. Forsyth et Fleck [46, 47]) que la peau humaine, quelle que soit sa couleur, présente un pic resserré dans le spectre des teintes. Ceci s'illustre aisément à l'aide de notre première transformation HSV (équation 5.7), appliquée à des régions de peau de couleurs différentes. Pour exemple, nous présentons dans la figure 6.10 deux images originales, les transformations HSV correspondantes et le spectre de deux régions spécifiques de peau, extraites manuellement. Dans la continuité de ce que nous avons proposé pour l'extraction de régions de végétation ou de ciel, un seuillage sur les teintes dans le spectre entier de l'image originale permet à l'inverse de ne conserver que les pixels dont la teinte correspond à celle de la peau humaine.

Nous conservons ainsi les pixels dont la teinte est comprise dans l'intervalle $]0, 60]$, cet intervalle correspondant grossièrement à des teintes rouge-orange-jaune.

Parmi les pixels extraits par ce seuillage et ne correspondant pas à des pixels de peau, se trouvent tous les pixels appartenant à des régions rouges de tout ordre, mais aussi des pixels, qui du fait de leur position près de l'axe de luminance dans l'espace HSV donnent lieu à des teintes aléatoires (cf. section 5.3). Si la détection des premiers points "parasites" est inhérente à notre technique qui se résume somme toute à un simple seuillage, les seconds peuvent être éliminés par une étape de transformation HSV améliorée. Le seuillage s'effectue alors non sur les teintes de l'image originale, mais sur les teintes de l'image transformée. Un tel rajout se heurte cependant à une limitation de taille : la peau noire, loin d'être considérée comme une région de couleur, est souvent assimilée à une zone grise ; sa teinte est donc supprimée et elle ne fait au final pas partie des pixels extraits ! Même une baisse du seuil de la transformation HSV améliorée ne permet de récupérer ces régions qu'au prix d'une classification erronée pour un nombre important de pixels.

Une autre technique de suppression des pixels parasites, de teinte indéterminée, basée sur leur caractère isolé (i.e. ils ne forment pas de grandes régions connexes), constituerait ainsi

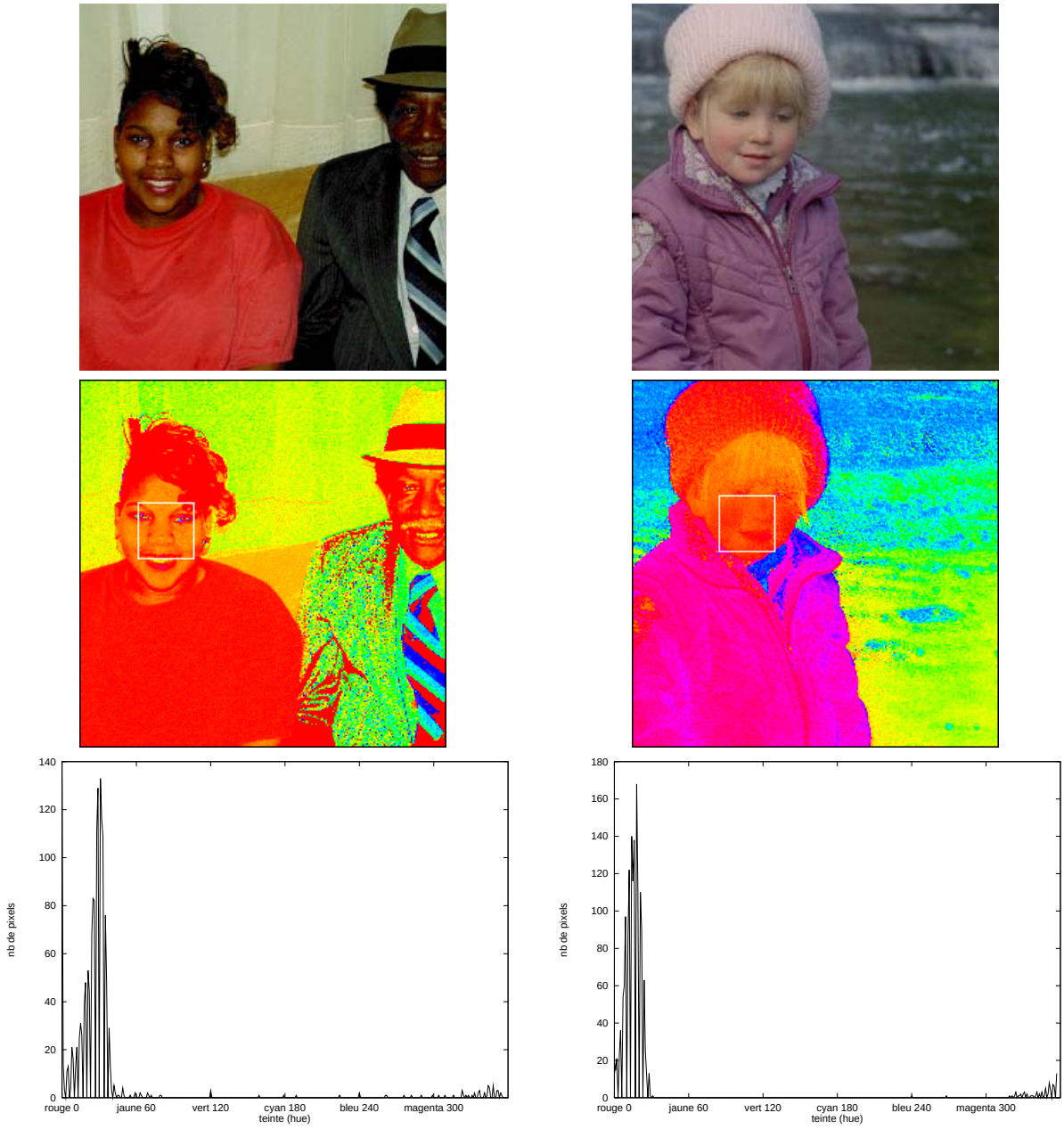


FIG. 6.10: Etude de la réponse spectrale de la peau humaine. A partir de deux images originales (©KODAK) contenant des exemples de peaux de couleurs différentes, on fournit les transformations HSV et les spectres obtenus sur des régions extraites manuellement. Notons qu'une région de couleur rouge mais ne contenant pas de peau (le pull de la première image par exemple) fournit également un spectre similaire.

une amélioration appréciable par rapport au simple seuillage, ou au seuillage précédé d'une étape de transformation HSV améliorée.

Conscients des limitations de l'utilisation de l'étape de transformation HSV, nous l'avons cependant conservée par la suite. Quelques exemples d'extraction de pixels de couleur peau sont présentés dans la figure 6.11. D'autres exemples pour un ensemble entier d'images clés d'une séquence seront proposés ultérieurement, notamment lors de la présentation de l'application de détection d'interview, dans la section 8.3.

Terminons l'exposé de cette deuxième application de l'outil d'étude spectral proposé par un rappel de la caractéristique de la classification obtenue, qui ne fournit qu'un sur-ensemble des regions contenant de la peau. Par la suite, cette classification présente l'avantage indéniable de permettre de restreindre à un petit nombre de régions l'application d'algorithmes plus spécifiques, par exemple d'extraction de visages.

Lorsque des hypothèses a priori sont disponibles sur la taille des visages recherchés, comme c'est le cas pour la détection de présentateur de journal télévisé, il est de plus possible d'utiliser d'autres critères également très simples, afin de réduire encore le champ de recherche (i.e. la taille des composantes extraites). De cette façon on évite le plus longtemps possible l'utilisation de critères plus complexes, tels que l'étude la forme, de la symétrie ou bien l'extraction des yeux, du nez et de la bouche.

En ce sens, ce troisième outil d'extraction d'objets particuliers rejoint les deux premiers présentés dans ce chapitre, permettant une première sur-sélection de zones d'incrustations et de texte dans les images clés. Nous proposons à présent, en guise de conclusion de ce chapitre, un bilan complet de ces trois outils.

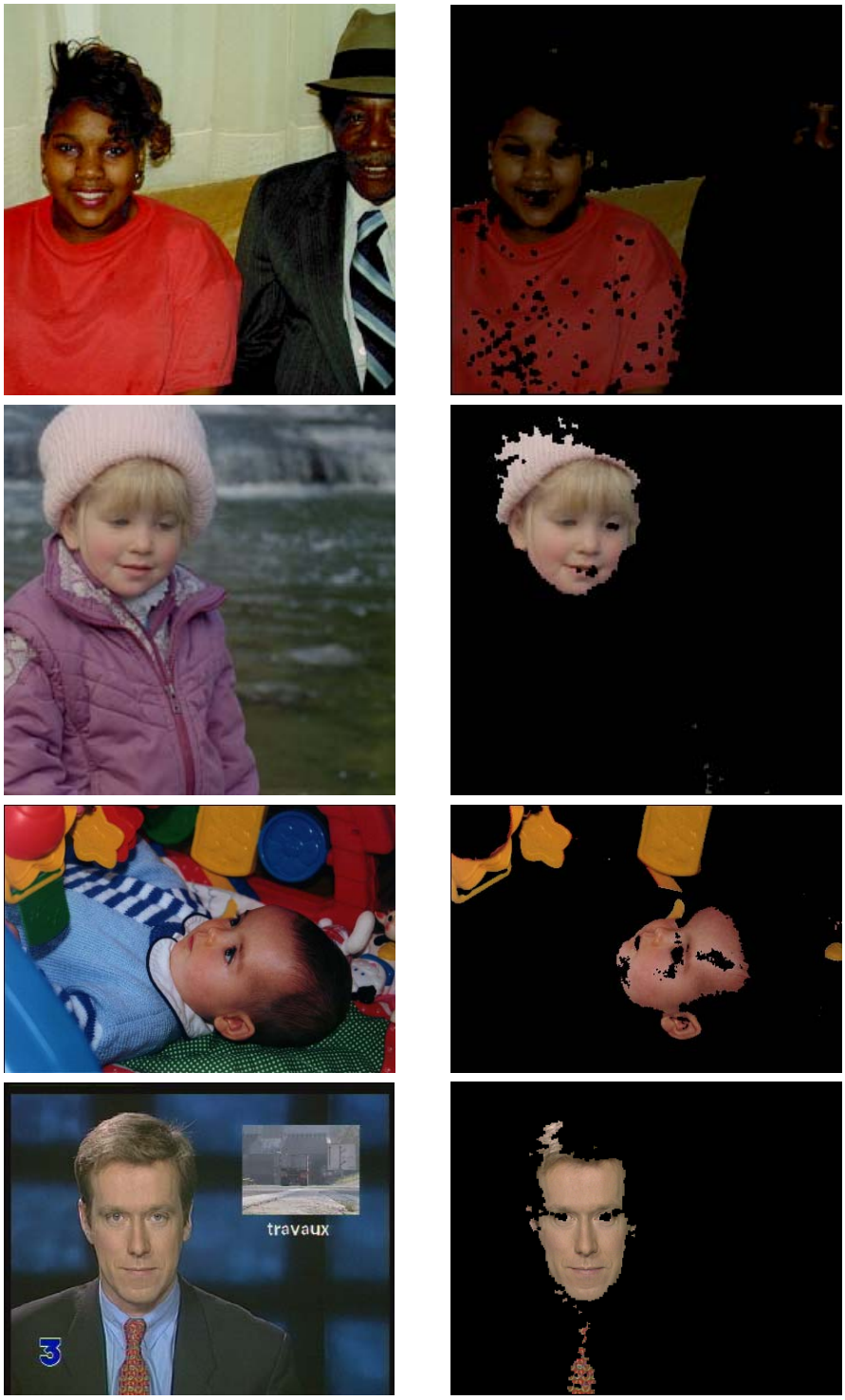
6.5 Conclusion

Nous venons de décrire trois outils destinés à l'extraction d'objets d'intérêt dans les images clés que sont les incrustations de bandeaux ou d'imagettes, le texte et les zones ayant une certaine réponse spectrale telles que le ciel ou la peau humaine.

Ces outils partagent tous les mêmes caractéristiques, désormais habituelles pour notre processus de structuration, d'automatisme et de simplicité de mise en œuvre : nous nous sommes en effet imposés de baser notre recherche sur des propriétés caractéristiques les plus simples possibles de ces objets, telles qu'une forme rectangulaire ou un alignement horizontal. Les trois méthodologies d'extraction auxquelles nous aboutissons fournissent en outre en général une première sélection des zones ayant une forte probabilité de contenir les objets recherchés. Ces zones constituent à leur tour une base de départ solide et restreinte sur laquelle lancer des algorithmes véritablement de reconnaissance (par exemple de texte ou de visage), automatiquement plus sophistiqués et plus lourds.

Dans leur état actuel, toutes les informations fournies par ces trois outils sont utilisées au chapitre 7 pour établir notre structure relationnelle. Nous verrons alors que le niveau de détection atteint permet d'ores et déjà l'établissement de liens sémantiques cohérents. Malgré tout, nous restons conscients que les méthodes développées n'ont pas encore atteint un parfait accomplissement. Aussi proposons-nous à présent quelques pistes d'améliorations de ces outils, qu'il serait intéressant de poursuivre.

Détection d'incrustations. L'utilisation de la morphologie mathématique par l'intermédaire de filtres dédiés à l'extraction de formes particulières trouve ici une application idéale. Deux étapes du processus restent cependant à améliorer, qui sont l'obtention



(a) Images originales (b) Résultats de l'extraction de peau

FIG. 6.11: Exemples d'extraction de pixels de couleur peau (b) à partir des images originales de la colonne (a).

de contours fermés et le bouchage de trous. Une étape supplémentaire de poursuite et fermeture de contours pour les zones de faible gradient (lorsque l'inscrustation est localement d'une couleur proche de celle du fond), est alors à envisager, à l'aide d'une des nombreuses techniques disponibles dans la littérature, en outre simplifiée du fait de la recherche de contours spécifiques, horizontaux ou verticaux. Quant au bouchage de trous à partir du point central de l'image, les tests effectués ont montré son insuffisance dans certains cas particuliers. Une solution consisterait alors à fusionner de façon "intelligente" les résultats de plusieurs bouchages de trous à partir de points différents, par exemple le centre et les quatre coins de l'image.

Enfin le modèle d'incrustation choisi est un rectangle parfait rencontrant ou non le bord de l'image. Or si la grande majorité des incrustations le vérifient, il est envisageable de l'assouplir pour obtenir la détection d'incrustations légèrement différentes, comme par exemple des rectangles parfaits sur certains côtés, et courbes sur les autres.

Détection de texte. Là encore les outils morphologiques et tout particulièrement le chapeau haut-de-forme ont prouvé leur grande efficacité. Dans l'hypothèse où nous ne cherchons à délimiter que les zones de texte suffisamment contrastées et en position relativement horizontale, un point reste à améliorer prioritairement : il s'agit, lorsque le texte est à cheval sur deux zones de luminances différentes, soit de faire précéder le chapeau haut-de-forme d'une égalisation du fond, soit d'inclure cette égalisation dans l'outil de chapeau haut-de-forme lui-même, en utilisant un chapeau modifié tel que, par exemple, le chapeau haut-de-forme inf présenté au chapitre 3.

Etude spectrale. Cette étude spectrale se résume à une sélection de certaines teintes dans les images, soit au niveau pixel, soit au niveau région d'une segmentation, basée sur la constatation que certains objets ont une teinte qui leur est propre. Cette étude fournit ainsi un premier indice d'importance pour l'élaboration d'une réponse à des questions d'un niveau sémantique élevé comme la présence de scène extérieure. Bien sûr une telle réponse ne peut découler uniquement d'une étude de la répartition des teintes dans une image, mais d'autres critères doivent être également extraits, puis fusionnés en une seule probabilité résultante, renforcée de par les réponses positives et successives à chaque critère individuel et indépendant. Il reste donc à incorporer notre outil dans un tel processus, dont le principe général - élaboration de critères simples et distincts, puis fusion - sera proposé à plusieurs reprises dans la suite de ce mémoire.

Nous venons de présenter trois exemples d'extraction d'objets particuliers. La liste n'est bien sûr pas exhaustive, et l'utilisation d'information supplémentaire, comme par exemple le mouvement dominant [11], conduirait à l'extraction de nouveaux objets, également source d'information sémantique.

L'extraction d'objets particuliers des images clés s'inscrit dans la deuxième phase de ce que nous avons appelé la structuration spatiale d'un document vidéo. Avec cette structuration spatiale s'achève également toute la partie linéaire de la structuration. Nous poursuivons l'exposé de nos travaux par la description de la structuration relationnelle cette fois-ci, i.e. l'établissement de relations entre toutes les entités que nous sommes à présent en mesure d'extraire : prises de vue, transitions, morceaux de prises de vue, images clés, régions et objets. Ces relations seront établies grâce à la mise en évidence de similarité de contenu entre ces diverses entités et tout particulièrement entre les images clés et les objets que nous venons d'extraire.

Troisième partie

Structuration relationnelle

Chapitre 7

Structuration relationnelle et hiérarchisation du document

7.1 Introduction

Avec ce chapitre débute la dernière étape de la structuration d'un document vidéo telle que nous l'avons décrite au chapitre 2 : après un découpage linéaire, tant temporel que spatial, ayant permis d'aboutir à une organisation en entités variées (prises de vue, transitions, morceaux de prises de vue, images clés, objets et régions), il reste à établir la structure relationnelle existant entre ces diverses entités. Cette structure relationnelle, introduite dans la section 2.3.2, correspond à l'établissement de relations de tout ordre entre deux entités quelconques d'un document vidéo.

Dans le cas général, les entités concernées par une mise en relation ne sont pas forcément adjacentes à l'intérieur du document vidéo. Cependant, l'établissement d'une relation entre deux entités voisines n'est bien sûr pas interdit. Et si ces relations particulières participent pleinement à la structuration relationnelle, elles sont également à l'origine, tout particulièrement lorsque les entités concernées sont des prises de vue, du regroupement de ces entités en d'autres entités d'un plus haut niveau syntaxique, les scènes et séquences (cf. section 2.3.1.1).

Cette extraction de scènes et séquences participe alors également à la partie linéaire temporelle de la structuration, dans un processus que nous avons appelé *hiérarchisation*. Cette notion est sans aucun doute à relier avec celle de stratification établie par Smith *et al.* [90, 91] et Davenport *et al.* [29], dans laquelle les auteurs développent une structure en prises de vue et scènes, basée sur une notion classique d'héritage entre les divers niveaux de la stratification. Yeung et Liu [102] proposent, quant à eux, une structure d'arbre, qui s'éloigne, de par son obtention, de la stratification et de la hiérarchisation. Les auteurs procèdent par classification pour regrouper les prises de vue les plus similaires. L'arbre obtenu peut alors être coupé à un certain niveau, correspondant par exemple à la plus forte valeur de dissemblance, et aboutissant à l'obtention d'une certaine partition des prises de vue. Que ce soit sous forme d'arbre, sous le nom de stratification, ou de hiérarchisation, cette étape supplémentaire de la structuration aboutissant aux scènes et séquences, représente de plus l'accès à un niveau sémantique élevé, à partir duquel la réponse à de simples requêtes telles que l'"extraction de tous les reportages d'un journal télévisé" est envisageable.

Si la structuration relationnelle, permet le processus linéaire de hiérarchisation lorsque des entités voisines sont mises en relation, elle ne se limite pas à cette étape essentielle certes,

mais qui reste un cas particulier. L'établissement de relations entre entités quelconques, de types variés et non plus toujours voisines, participe également pleinement à la représentation du contenu sémantique d'un document.

Et c'est à partir de cette représentation, véritable support de l'indexation, que la réponse à des requêtes plus complexes peut avoir lieu, comme par exemple la recherche des parties d'un document contenant un même personnage ou toute référence à ce personnage.

Après l'introduction de quelques définitions et notions essentielles à la compréhension du processus de structuration relationnelle, l'objet de ce chapitre sera donc d'exposer plusieurs techniques de mise en relation d'entités quelconques.

Nous détaillerons ainsi successivement des outils permettant d'extraire des relations d'ordre général (cf. section 7.3), et d'autres plus spécifiques à un type particulier de documents vidéo, que sont les journaux télévisés (cf. section 7.4). Dans un cas comme dans l'autre, nous aurons à cœur de souligner dans quelle mesure les outils développés dans le cadre de la structuration relationnelle, contribuent également à l'étape de hiérarchisation, i.e. au découpage en scènes et séquences.

7.2 Structuration relationnelle : définitions

Lors de la présentation de l'ensemble du processus de structuration d'un document (cf. chapitre 2), il était apparu que la structure de graphe permettait de représenter au mieux la partie relationnelle de cette structuration.

Pour un document vidéo donné, un tel graphe relie deux entités de types quelconques (correspondant aux nœuds du graphes) et trouvées en relation, par un arc symbolisant cette relation. En d'autres termes, deux entités d'un document vidéo donné appartiennent au graphe de relation dès qu'il existe une relation de tout ordre entre elles. En ce sens le graphe de relation n'est pas spécifique à un type de relation donné, mais représente aussi bien les relations "contient un même personnage" que "appartient à une même scène" par exemple.

A ce stade, deux remarques s'imposent dans le cadre d'une vision plus globale de l'apport de nos travaux au problème de l'indexation et de la recherche d'information sur des documents indexés. Tout d'abord c'est sur ce réseau, et en suivant le faisceau de relations tissé, que l'on peut envisager de répondre aux requêtes utilisateur : à chaque requête, certains arcs du graphe, i.e. certaines relations, pourront être suivies parce que jugées en adéquation avec la requête, et d'autres seront éliminées. Il est également envisageable d'affecter à chaque arc un degré d'adéquation, permettant d'instaurer des hiérarchies au sein même des relations établies, en fonction de la requête.

La deuxième remarque qu'il convient d'effectuer concerne l'étendue de ce graphe à d'autres documents. Des relations inter-documents peuvent en effet être établies au même titre que les relations intra-document, et donner lieu à des réponses multiples pour une même requête.

Ces deux remarques d'ordre général étant faites, revenons à présent concrètement sur la construction du graphe de relations. Pour un document donné, on ne sait, et parfois même on ne peut, définir "l'ensemble de la sémantique" contenue dans ce document, ne serait-ce que parce que cette dernière peut être multiple. Dans ces conditions, il n'est pas envisageable de construire un seul graphe capable de représenter l'ensemble de cette, ou ces, sémantique(s). Il est cependant possible d'associer un graphe donné à tout niveau de pénétration de la sémantique, même lorsque ces niveaux restent superficiels.

Nous ne prétendons donc pas fournir ici d'exemple de graphe complet mais simplement des

prémices de construction de tels graphes. Plus particulièrement, les outils que nous proposons par la suite ne permettent à chaque fois d'établir qu'un seul type de relations.

Puisqu'un seul et unique type de relations est considéré à chaque étape, on est alors amené à définir des groupes de relations :

Définition 20. Groupe de relations *Un groupe de relations est un sous-graphe de relations pour lequel un seul type de relation existe entre deux entités du sous-graphe (par exemple, l'ensemble des prises de vue en relation par comparaison de leurs images clés).*

Suivant que l'on considère que toutes les prises de vue d'un même groupe doivent être en relation deux à deux, ou non, deux types différents de groupes de relations sont ensuite construits. Ceci revient par ailleurs à considérer que la propriété de transitivité de la relation courante est vérifiée ou non :

Définition 21. Groupe strict de relations *: Une famille d'entités est un groupe strict de relations si, et seulement si, chacune des entités de cette famille est en relation avec toutes les autres entités de la famille. La relation est transitive.*

Définition 22. Groupe large de relations *: Une famille d'entités est un groupe large de relations si, et seulement si, chacune des entités de cette famille est au moins en relation avec une autre entité de la famille. La propriété de transitivité n'est pas vérifiée.*

De ces deux définitions découle directement l'inclusion de l'ensemble des groupes stricts dans l'ensemble des groupes larges. Par la suite, chacun des outils détaillés dans ce chapitre fournit au final un résultat sous forme de groupes de relations soit stricts, soit larges.

Avant de passer à leur exposé, nous souhaitons dès à présent insister à la fois sur la nature des relations extraites, qui appartiennent à un niveau sémantique déjà élevé, et sur celle des outils proposés, qui relèvent encore une fois de traitements bas niveau et simples à mettre en œuvre.

Le passage entre ces outils de traitements d'images proprement dits et la sémantique extraite des groupes de relations sera alors effectué par l'intermédiaire d'une étude de l'ensemble des propriétés bas niveau, nécessaire et suffisant à la caractérisation d'une notion sémantique donnée. Plus qu'une seule de ces propriétés, c'est une réponse positive à chacune d'entre elles qui permet d'établir la caractérisation sémantique recherchée.

Tous les outils que nous proposons à présent sont autant d'illustrations de cette méthodologie d'extraction et de fusion de plusieurs caractéristiques.

7.3 Relations d'ordre général

La mise en relation s'effectue naturellement par extraction de points communs entre deux entités données. Par "point commun", on choisit volontairement un terme très vague, pouvant s'appliquer à un large domaine de notions, telles que deux images clés similaires, ou un même objet reconnu dans deux scènes différentes. Il est même possible d'établir des relations reposant sur des informations plus "virtuelles", ou du moins moins concrètes. Le regroupement de prises de vue, non pas parce qu'elles contiennent un élément commun, mais parce qu'elles sont situées entre deux prises de vue particulières et elles-mêmes en relation, en est un exemple.

Nous proposons donc ici de détailler deux outils de détection de relations bien ciblées, que sont l'extraction d'images clés similaires et la persistance d'incrustation d'une entité à une autre. Toutes les relations établies par ces outils possèdent la caractéristique d'être assez générales pour pouvoir être appliquées à tout type de document vidéo.

7.3.1 Utilisation des images clés

L'objet de cette section est d'exposer l'outil de comparaison d'images clés que nous avons mis en place afin d'établir des relations entre diverses prises de vue (ou morceaux de prises de vue).

Une technique similaire de mise en relation de prises de vue, par comparaison de leurs deux uniques images clés, a par ailleurs été proposée par Arman *et al.* [7]. Notre méthode va plus loin dans la mesure où chaque prise de vue peut posséder plus d'une image clé. Le critère de comparaison est donc évalué pour toute paire d'images clés, provenant respectivement de l'une et de l'autre prise de vue, augmentant ainsi le nombre de possibilités d'établissement de relations, au détriment du temps de calcul, qui est alors proportionnel au nombre de comparaisons effectuées.

Yeung et Liu [102] proposent également une technique d'extraction de relations entre prises de vue par comparaison de leurs images clés, proches de notre méthode. La définition qu'ils donnent à la relation établie est en effet exactement équivalente à notre notion :

Définition 23. Relation par image clé *Soient deux prises de vue P_1 et P_2 pour lesquelles on a extrait les ensembles d'images clés respectifs $\{I_1^i\}_{i=d_1 \dots f_1}$ et $\{I_2^i\}_{i=d_2 \dots f_2}$. P_1 et P_2 sont en relation, par comparaison de leurs images clés, si, et seulement si, il existe au moins une paire d'images clés (I_1^n, I_2^m) similaires, au sens du critère de similarité choisi, pour la comparaison de deux images.*

Le critère de similarité entre les deux prises de vue, noté $C(P_1, P_2)$, s'exprime alors simplement comme le minimum des valeurs du critère de dissemblance obtenues pour toutes les paires d'images clés, issues de P_1 et P_2 :

$$C(P_1, P_2) = min_{i,j}(C(I_1^i, I_2^j)),\ d_1 \leq i \leq f_1,\ d_2 \leq i \leq f_2 \tag{7.1}$$

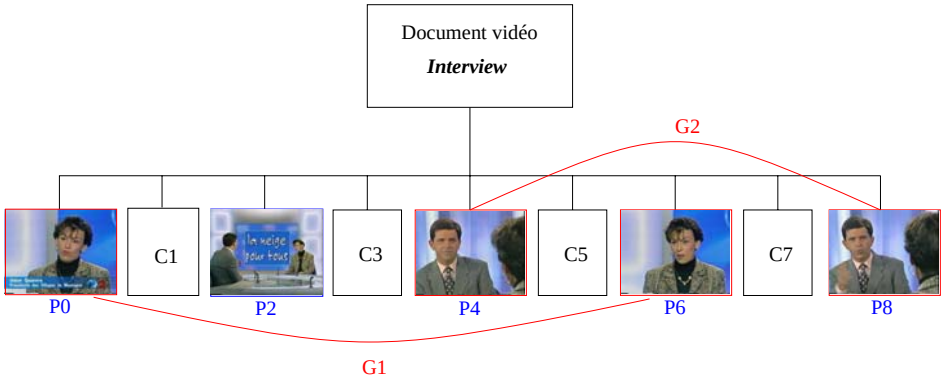


FIG. 7.1: Groupes de relations, G_1 et G_2, obtenus par comparaison des images clés pour le document *interview*, contenant 5 prises de vue P_0, P_2, ... P_8 séparées par les coupures C_1, ... , C_7. Seules les premières images clés de chaque prise de vue sont représentées.

La différence se situe alors dans le choix du critère de similarité utilisé pour les images clés. Yeung *et al.* proposent une double comparaison basée sur un critère de luminance et un critère de couleur. Dans notre cas, et toujours dans le souci de limiter la complexité des calculs mis en œuvre, nous avons repris le critère de similarité entre images, élaboré pour l'algorithme de détection de coupures. Ce critère, basé sur un calcul local (i.e. par blocs) de distance moyenne

pixel à pixel dans l'espace RGB, a par ailleurs déjà prouvé son efficacité à deux reprises, lors de la détection de transitions de type coupures, et lors de la décision d'existence de changements à l'intérieur d'une même prise de vue. Comme pour les deux utilisations précédentes, la décision d'existence d'une similarité entre deux images clés est obtenue par comparaison de la valeur du critère à un nouveau seuil, appelé *seuil de relation*. Ce seuil est encore une fois fixé à une valeur constante pour l'ensemble de la base de données dont nous disposons, et découlant du seuil global choisi pour la détection de coupures (cf. section 3.2.6), soit deux fois ce seuil global. Il convient de remarquer à ce stade, que le seuil de relation est donc égal pour l'ensemble de nos tests au seuil de changement (cf. section 4.4), ce qui est logique dans la mesure où les deux notions sont duales l'une de l'autre : deux images similaires, i.e. en deçà du seuil, sont dites en relation, et au contraire deux images de contenus différents, i.e. au delà du seuil, sont considérées comme significatives d'un changement. La différence se situe alors dans le fait que, pour l'opérateur de changement, seules les images clés d'une même prise de vue sont comparées deux à deux, alors que pour l'opérateur de mise en relation, il s'agit d'images clés provenant de prises de vue différentes.

Quant à la décision globale d'existence d'une relation entre deux prises de vue comparées, elle découle directement de la mise en relation d'une unique paire de leurs images clés, comme pour l'opérateur de changement, où la décision d'un changement interne à une prise de vue donnée dérivait directement de la détection d'un changement entre deux de ses images clés.

La méthode complète de mise en relation de prises de vue ou morceaux de prises de vue est alors proposée sous la forme de l'algorithme 6.

Algorithme 6 Algorithme de mise en relation de prises de vue ou morceaux de prises de vue par comparaison de leurs images clés.

Pré-condition : s_r le seuil de relation

pour tout (P_i, P_j) couple de prises de vue ou morceaux de prise de vue **faire**
  pour tout (I_i^k, I_j^l) couple d'images clés **faire**
    Calcul du critère de similarité $C(I_i^k, I_j^l)$
    si $C(I_i^k, I_j^l) \leq s_r$ **alors**
      P_i et P_j sont en relation
      Sortie de la boucle sur les images clés
    fin si
  fin pour
fin pour

La force de cet algorithme repose donc à la fois sur sa grande simplicité et sur sa relative robustesse dans la mesure où la valeur du seuil de relation peut être choisie une fois pour toutes et pour l'ensemble des documents vidéo testés. Les résultats obtenus sont alors globalement corrects, comme le prouvent les deux groupes de prises de vue en relation établis dans la figure 7.1 pour le document *interview*. Notons qu'il s'agit, pour ce cas trivial, de groupes stricts, puisque seules deux prises de vue appartiennent à chaque groupe.

Un autre exemple des relations obtenues est par ailleurs proposé dans la figure 7.2. Au travers de ce deuxième exemple, il est à noter que pour le choix de seuil effectué, certaines

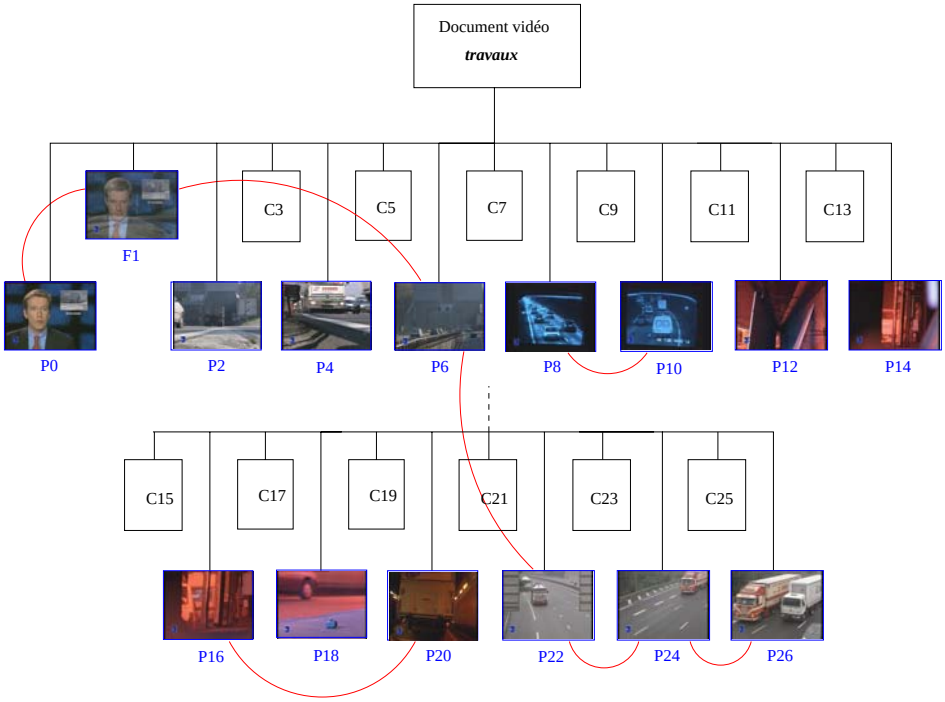


FIG. 7.2: Relations extraites par comparaison des images clés pour le document vidéo *travaux*, contenant 14 prises de vue $P_0 \ldots P_{26}$, séparées par un fondu F_1 et 12 coupures $C_3, \ldots, C_{25}$. Certaines relations extraites sont erronées, comme par exemple entre le fondu F_1 et la prise de vue P_6.

relations établies n'ont manifestement pas lieu d'être d'un point de vue sémantique. Ceci provient du caractère très simple de la comparaison effectuée, ne prenant en compte qu'une distance de couleur entre deux images. Un choix de seuil plus bas, i.e. une fois et demie le seuil global, et toujours constant pour l'ensemble de la base de données, permet de réduire considérablement le taux de fausses alarmes.

En terme de rapidité, le temps de calcul du critère proprement dit, déjà évalué lors de la détection de coupures, représente un coût faible. L'ensemble du processus dépend donc directement du nombre d'images clés sélectionnées par prise de vue. Deux prises de vue ayant respectivement n_1 et n_2 images clés seront donc déclarées en relation au pire au prix de $n_1 \times n_2$ comparaisons. Pour un document vidéo complet comprenant p prises de vue, C_p^2 couples de prises de vue doivent être comparées. L'intérêt de ne conserver qu'une ou deux images clés, pour la majorité des prises de vue sans changement, apparaît alors de façon cruciale pour cette application de mise en relation de prises de vue.

Le choix de favoriser la simplicité du critère s'effectue en outre au détriment de la détection de certaines relations, dont les plus évidentes sont sans doute celles existant entre deux images clés de contenus similaires, mais pour lesquelles l'une d'elles a subi un mouvement important. De par son expression, le critère de dissemblance appliqué n'est pas en mesure d'établir de relation dans ce cas, et ceci quelle que soit la valeur du seuil de relation choisie. L'établissement de ces relations supplémentaires doit faire l'objet d'une autre procédure, plus lourde et plus coûteuse en temps de calcul, et reposant sur une compensation du mouvement de la scène. Avec les contraintes que nous nous sommes imposées de simplicité et rapidité des outils développés, l'algorithme de mise en relation par comparaison des images clés, présenté ici, est cependant robuste aux petits changements de scènes et aux petits mouvements d'objets, comme l'illustrent les deux exemples de graphes de relations obtenus.

Enfin nous terminons l'exposé de cet outil par un rapide tour d'horizon de ses applications éventuelles. Certaines ont déjà été développées et d'autres le seront dans le cours de ce manuscrit, aussi nous ne fournissons ici qu'une liste, par ailleurs non exhaustive, sans illustrer nos propos :

- Cet outil a pour première application directe la correction de la détection de transitions effectuée (application intitulée *validation de la structure linéaire*, au chapitre 8).
- Il permet en outre de gérer la présence des flashes (cf. micro-découpage temporel, chapitre 4 et section 8.2, chapitre 8).
- Enfin, et il ne s'agit pas de la moindre des applications possibles, l'algorithme de mise en relation est le premier outil développé dans le cadre de ce mémoire, visant à l'extraction d'un niveau supplémentaire, dans la hiérarchie des entités d'un document vidéo, que sont les scènes (cf. section 8.4).

Nous proposons à présent un autre outil d'établissement de relations, toujours entre prises de vue, reposant cette fois-ci sur la détection d'incrustations et de bandeaux proposée au chapitre 6.

7.3.2 Persistance d'incrustations et de bandeaux

Nous commençons l'exposé de ce deuxième outil par un extrait de structure linéaire établie pour le document vidéo *Lille*, et proposée dans la figure 7.3. Une incrustation apparaît en haut à gauche au cours de la prise de vue 2, incrustation qui, loin de disparaître lorsque la prise

de vue est terminée, se poursuit dans les deux prises de vue suivantes. Un tel événement est évidemment synonyme d'un lien sémantique entre ces trois prises de vue.

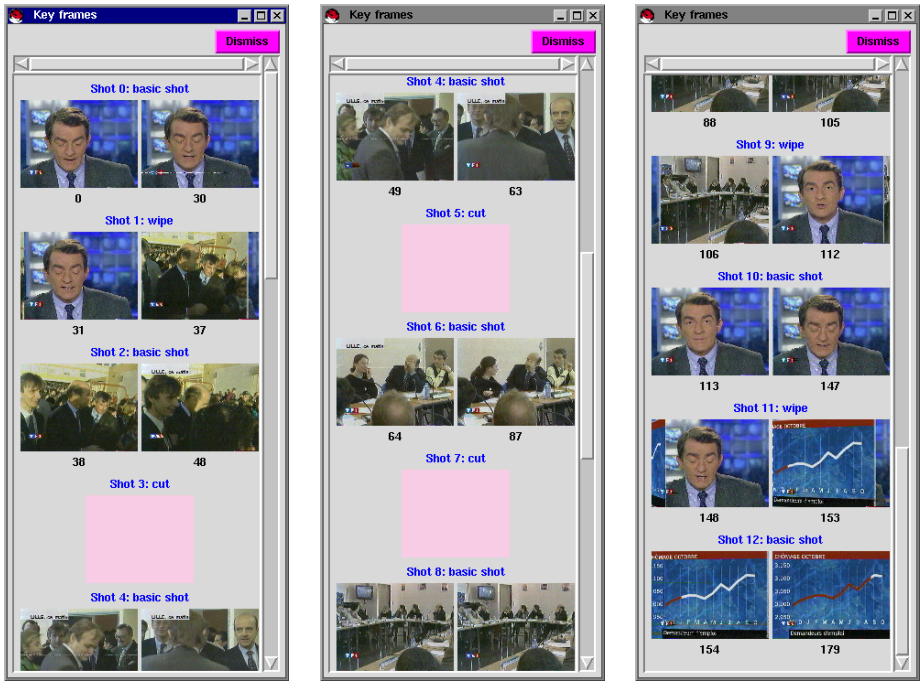


FIG. 7.3: Structure linéaire établie pour le document vidéo *Lille*, comportant une incrustation persistante entre les prises de vue 2, 4 et 6 (en haut à gauche dans les images).

L'objet de cette section est donc de détecter ce type d'événement, i.e. la poursuite d'une même incrustation d'une prise de vue à la suivante, de façon à établir un nouveau type de relation entre les deux prises de vue concernées. Dans un deuxième temps, et ceci est à mettre au nombre des perspectives d'amélioration de nos travaux, il est envisageable de détecter, non plus la persistance d'incrustation d'une prise de vue à la suivante, mais des similarités d'incrustation (du moins de leur forme) entre deux prises de vue quelconques. Ce type de relation reposant alors sur le fait que la forme, ou plutôt le format, des bandeaux et incrustations est spécifique à un type de scènes ou de séquences particulier. Ainsi à l'intérieur d'un même journal par exemple, les reportages contiendront un format et une période d'apparition des bandeaux spécifiques au journal étudié. Ce type spécifique, qui peut alors être considéré comme une véritable signature, symbolise de par son apparition un changement de sujet. Il est également susceptible de varier en fonction du type de scène ou de séquence rencontré (reportage, prise de vue de présentateur, interview, etc.)

Ces remarques sont autant de preuves de l'apport de ces relations dans un but d'extraction de scènes et de séquences, mais aussi de construction de lien sémantique entre diverses prises de vue éloignées, dans le temps.

L'outil proposé se décompose en trois étapes, que nous décrivons maintenant au travers de l'exemple du document *Lille*.

La première de ces étapes consiste en la détection des incrustations sur les images clés de chacune des prises de vue disponibles, ce qui fournit pour le document *Lille* le résultat présenté dans la figure 7.4. Pour cette étape, nous réutilisons bien sûr l'algorithme décrit dans

la section 6.2.

FIG. 7.4: Résultat de la détection d'incrustations et de bandeaux sur le document *Lille*. La non détection d'incrustation dans la prise de vue 12 est due à une piètre qualité, engendrant un décalage de la dernière ligne de l'image vers la gauche. L'incrustation n'est alors plus parfaitement rectangulaire.

Une fois ces incrustations extraites, il s'agit de les comparer. Ainsi que nous l'avons déjà évoqué, cette comparaison pourrait dans le futur être effectuée entre incrustations de prises de vue quelconques. Nous nous limitons ici à une comparaison entre incrustations de prises de vue voisines, une fois les coupures exclues. Plus exactement, la comparaison n'a lieu que dans le cadre restreint de la détection d'une incrustation sur la dernière des images clés d'une prise de vue et sur la première des images clés de la prise de vue suivante, ceci dans le souci d'introduire une contrainte de persistance temporelle de l'incrustation, à cheval sur deux prises de vue.

La décision de similarité entre deux incrustations que nous qualifierons désormais de "voisines" combine alors un critère de position de l'incrustation dans l'image et un critère de similarité de contenu. La probabilité que les deux incrustations soient positionnées au même endroit dans l'image, notée $P_{place\_incrust}$, est ainsi mesurée par le rapport de la surface commune aux deux incrustations, notées $incrust_1$ et $incrust_2$, sur la plus grande des surfaces

des deux incrustations :

$$P_{place\_incrust}(incrust_1, incrust_2) = \frac{\text{Aire}(incrust_1 \cap incrust_2)}{\max(\text{Aire}(incrust_1), \text{Aire}(incrust_2))} \tag{7.2}$$

Quant à la comparaison de leur contenu, elle est effectuée une fois encore par le critère de dissemblance mis au point pour la détection de coupures. Ce critère est cependant cette fois-ci appliqué globalement (et non plus par blocs, de par la faible surface des incrustations), et sur l'intersection des deux incrustations, une fois superposées. Cette superposition, qui ne tient absolument pas compte de leurs positions respectives dans les images clés de départ, revient à extraire les incrustations et à les considérer comme des images à part entière : le critère est calculé sur la surface commune entre ces deux nouvelles images, et correspond toujours à la moyenne des distances couleur pixel à pixel sur cette surface.

La troisième étape de l'opérateur consiste en une comparaison à la fois de la probabilité $P_{place\_incrust}$ avec un seuil que nous avons fixé à 90%, et du critère de dissemblance de contenu avec un autre seuil cette fois-ci fixé à 0.4. Notons que les deux seuils jouent un rôle inverse, dans la mesure où les deux incrustations sont considérées comme ayant des positions similaires, lorsque $P_{place\_incrust}$ atteint une valeur supérieure au seuil de 90%, alors que leurs contenus sont jugés identiques lorsque le critère de dissemblance prend une valeur cette fois-ci inférieure au seuil de 0.4.

Dans le cas particulier, peu fréquent, où plusieurs incrustations seraient détectées sur une même image clé, la recherche d'incrustations similaires est alors effectuée pour chaque incrustation, i.e. pour chaque composante connexe séparemment. Enfin, on autorise une poursuite de la persistance d'une même incrustation sur plus de deux prises de vue successives. Ceci s'effectue simplement par la vérification, sur les prises de vue intermédiaires, i.e. situées entre la prise de vue où apparaît l'incrustation et celle où disparaît l'incrustation, de la présence d'une même incrustation au début et en fin de prise de vue.

Revenons à présent au document *Lille*, que nous avons choisi pour illustrer les résultats de notre algorithme. Du fait d'une probabilité $P_{place\_incrust}$ élevée (supérieure à 98%) et d'une dissemblance de contenu estimée à $0.2 - 0.3$, des relations sont établies entre les prises de vue 2, 4 et 6, ce qui résulte pour ce même document en un groupe de relations de type strict, correspondant à la relation "persistance d'incrustation", représenté dans la figure 7.5. Le caractère strict des groupes extraits est cohérent avec la définition de la relation telle que nous l'avons établie : une relation n'est définie qu'entre deux prises de vue successives. Dans le cas d'une persistance d'une même incrustation sur un intervalle de temps de plus de deux prises de vue, les relations établies sont également de type strict. Il en est de même dans le cadre d'une étendue de ce critère de mise en relation à des prises de vue non plus successives mais à des endroits quelconques du document vidéo.

Nous terminons l'examen de ce nouvel outil par une remarque sur la transposition possible de l'ensemble de l'algorithme développé à un autre critère de mise en relation de prises de vue qu'est la persistance de texte, typiquement extrait des bandeaux, d'une prise de vue à une autre.

L'exposé des deux outils d'établissement de relations d'ordre général entre prises de vue, développés dans le cadre de nos travaux, étant achevé, nous poursuivons par la description de deux algorithmes de mise en relation, d'ordre plus spécifique cette fois-ci, car adaptés à la classe des documents de type journaux télévisés.

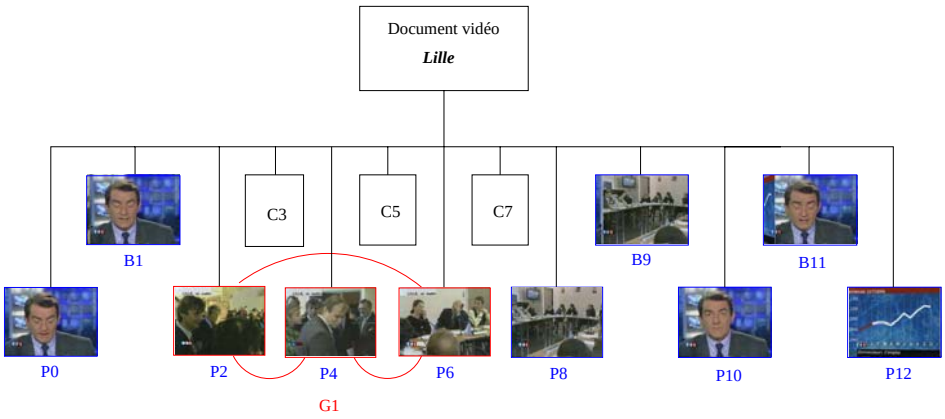


FIG. 7.5: Groupe de relations résultant de la détection d'incrustations persistants d'une prise de vue à la suivante, sur le document *Lille*.

7.4 Relations spécifiques

Dans le cadre de l'exposé de relations spécifiques à des documents vidéo particuliers, deux techniques de mise en relations, dédiées aux journaux télévisés, sont présentées dans cette section. Nous détaillons ainsi tout d'abord les relations découlant directement de la détection de transitions de types spécifiques. Puis nous proposons un algorithme complet de détection des prises de vue de présentateur, débouchant sur deux types de relations duales l'une de l'autre : toutes les prises de vue contenant le présentateur sont en relation, et par complémentarité sur l'ensemble du document vidéo, l'ensemble des prises de vue comprises entre deux prises de vue de présentateur, et ne le contenant pas, sont également considérées en relation.

7.4.1 Détection de transitions spécifiques

Depuis l'exposé de nos algorithmes de structuration linéaire temporelle, nous sommes en mesure de détecter les transitions de type fondu et de les étiqueter comme telles. Or ces transitions jouent couramment un rôle particulier dans l'organisation d'un journal télévisé : alors que deux prises de vue successives d'un même reportage sont séparées classiquement par une coupure, le passage du présentateur à un reportage est lui symboliquement représenté par une transition de type fondu. Bien sûr ce symbolisme des fondus, s'il est classique, n'est pas automatique et il arrive que d'autres types de transitions jouent ce rôle de délimitation. D'autre part, si la séparation prise de vue de présentateur / reportage est typiquement de l'ordre d'un découpage en séquences, le type de transitions peut également varier lors d'un passage d'une scène à une autre, comme c'est par exemple le cas pour le document vidéo *Lille*, dans lequel les balayages délimitent les diverses scènes (cf. figure 7.6).

L'outil que nous proposons ici est uniquement axé sur la détection d'un type particulier de transitions comme passages obligatoires entre deux séquences ou scènes. Comme exemple concret de ce partitionnement en scènes et séquences, citons le passage d'une prise de vue de présentateur à un reportage dans un journal télévisé qui se symbolise par un fondu ou un balayage. Cet outil se formalise par ailleurs relativement aisément par la règle de mise en relation suivante, où T_i, T_j sont deux transitions spécifiques, par exemple des balayages, et

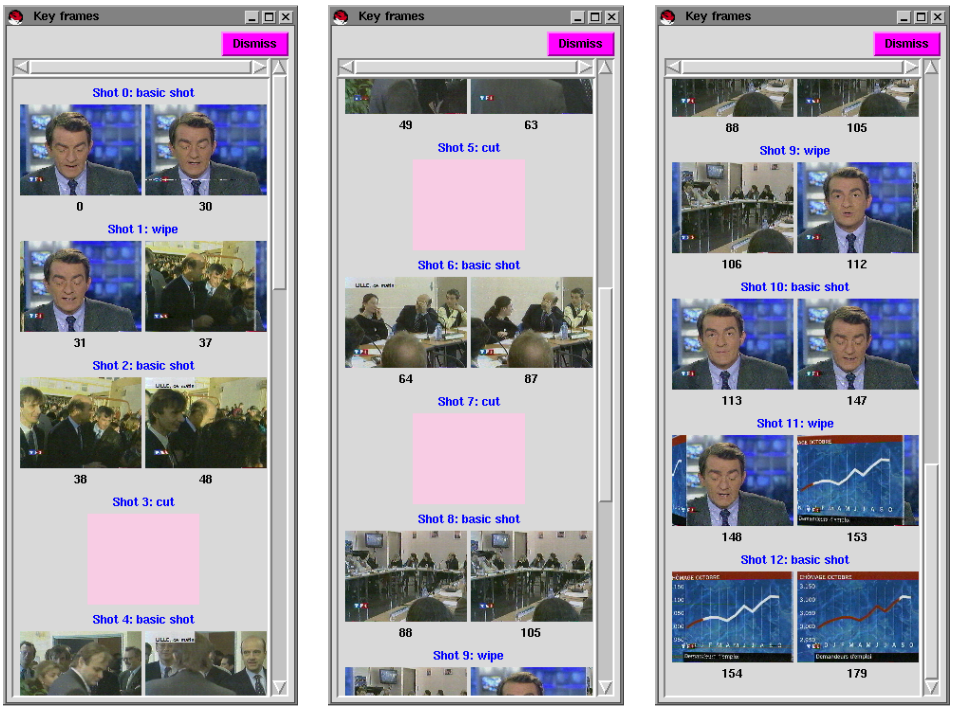


FIG. 7.6: Structure linéaire temporelle du document *Lille*.

successives d'un document vidéo de type journal télévisé (i.e. $i < j$) :

$$\forall (P_k, P_l) \text{ prises de vue, } \forall (k,l) \in]i,j[^2,\ P_k\ \mathcal{R}\ P_l \tag{7.3}$$

De par cette définition, les groupes de relations créés sont intrinsèquement de type strict.

L'application de cette règle aux documents de la base de données, extraits de journaux télévisés, donne lieu dans tous les cas à l'établissement de relations correctes et à un découpage satisfaisant (correspondant aux reportages) en différentes séquences. Nous en donnons une illustration dans la figure 7.7, pour le document *Lille* présenté ci-dessus.

La description de ce premier outil de détection de relation spécifique liée au type de transition utilisé est à présent terminée. Nous proposons dans la section suivante un deuxième exemple, plus sophistiqué, de mise en relation de prises de vue reposant sur l'extraction, toujours dans un journal télévisé, des prises de vue contenant le présentateur.

7.4.2 Détection de présentateur

Etre capable d'extraire, d'un journal télévisé, les prises de vue particulières dans lesquelles le présentateur apparaît, présente plusieurs intérêts. Ces prises de vue rythment en effet la succession des divers sujets d'actualité, et en ce sens, ont un rôle de délimiteur, tout comme les transitions de type fondus dans la section précédente. En effet, le fait d'apposer l'étiquette présentateur à certaines prises de vue permet par dualité de dire que toutes les prises de vue, dans le cas d'un journal télévisé, entre deux apparitions du présentateur font partie du même reportage, et donc portent sur un même sujet d'actualité.

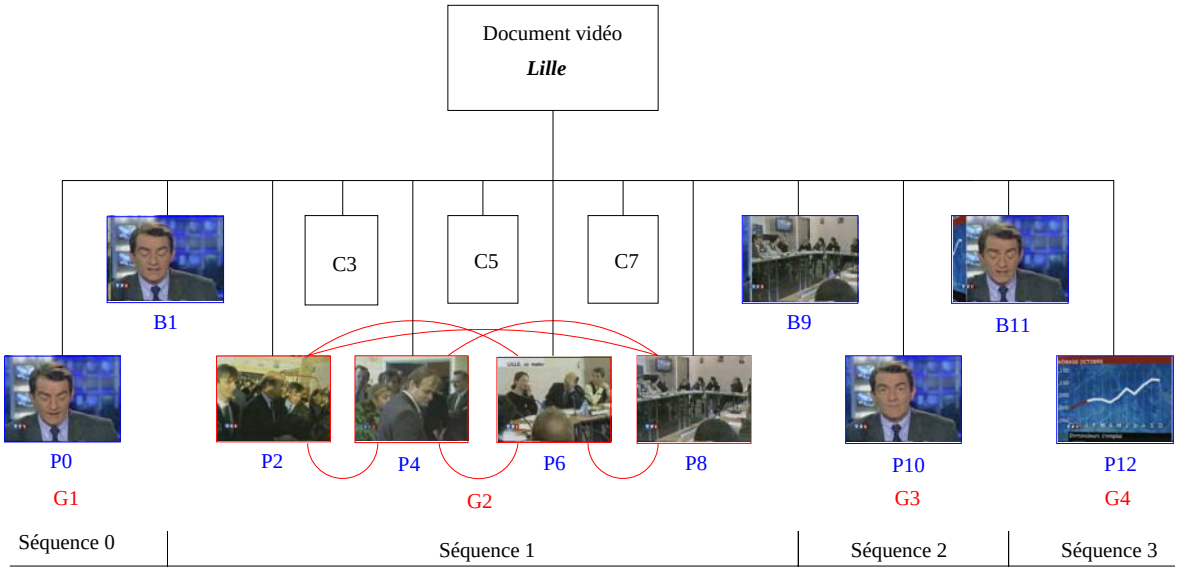


FIG. 7.7: Groupes stricts de relations obtenus par la détection de transitions spécifiques et découpage en séquences correspondant, pour le document *Lille*. Seules les premières images clés de chaque prise de vue sont représentées. Le document est composé de sept prises de vue séparées par trois balayages B1, B9 et B11 et trois coupures C3, C5 et C7.

D'autre part, puisque le rôle du présentateur est, en autres, de présenter les divers reportages, ses paroles doivent contenir les mots clés qu'il est important de relever pour comprendre le sujet de ces reportages. Outre ces mots clés, la présence d'imagettes, incrustées dans l'image même de la scène contenant le présentateur, a déjà pour but de donner une illustration en images du reportage à venir. Détecter les prises de vue de présentateur fournit donc de plus un indicateur des prises de vue dans lesquelles il sera primordial par la suite de lancer des outils de recherche de mots clés ou bien d'extraction d'incrustations.

Pour toutes ces raisons, dont la liste n'est pas exhaustive, nous avons donc réalisé, sur le principe d'agrégation des résultats de plusieurs outils bas niveau et de règles de cohérence, une technique de caractérisation des groupes de prises de vue, ayant la plus forte probabilité de correspondre au présentateur.

Nous proposons tout d'abord d'étudier diverses propriétés caractéristiques des prises de vue de présentateurs, qui restent somme toute similaires quelle que soit la chaîne, le pays, etc.

La première de ces propriétés concerne la réapparition de façon relativement périodique de ces prises de vue au cours du journal télévisé. Elles ont d'autre part, par essence, des contenus similaires.

La troisième propriété caractéristique des prises de vue de type présentateur nécessite l'hypothèse supplémentaire que le document vidéo traité correspond à un journal télévisé entier. Avec cette hypothèse, il est fréquent de commencer et de terminer le journal par une prise de vue de présentateur. Si cette prise de vue n'est pas toujours exactement la première et la dernière du journal, du moins est-elle située dans les toutes premières et les toutes dernières du document.

Par définition une prise de vue de présentateur contient (au moins) une personne, qui est généralement de face, globalement au milieu de l'écran et à une distance moyenne de la

caméra.

Enfin une telle prise de vue contient généralement assez peu de mouvement et le fond de la scène est particulièrement immobile.

Face à ces constats, quatre critères individuels ont été bâtis, puis fusionnés en une seule probabilité globale d'avoir affaire à une prise de vue de présentateur. Nous en proposons à présent une description plus détaillée.

Critère de sélection du groupe maximal de prises de vue en relation. Ce critère repose sur les deux propriétés d'apparition régulière du présentateur au cours du journal télévisé et de similarité de contenu des prises de vue de présentateur. L'ensemble de ces deux points nous fait donc rechercher le groupe maximal de prises de vue en relation dans le document, i.e. le groupe contenant le plus grand nombre de prises de vue en relation. A ce stade, un point doit être conservé en mémoire : le caractère périodique de l'apparition du présentateur n'a pas été mis en œuvre dans l'outil que nous proposons.

Pour extraire du document à traiter les groupes de prises de vue en relation, l'outil d'établissement de relations entre prises de vue par comparaison de leurs images clés est utilisé. La similarité que l'on cherche à établir entre les images de présentateur est en effet directement liée à une comparaison globale de leur contenu, aucune compensation de mouvement par exemple n'est nécessaire, dans la mesure où il est probable que la caméra à l'origine de ces prises de vue est fixe et où les mouvements du présentateur sont minimes.

La constitution de groupes stricts a d'autre part été retenue pour deux raisons. La première vise à renforcer le poids des relations établies au sein d'un groupe strict. En effet, on a tendance à considérer plus sûres les relations existant à l'intérieur d'un groupe strict, puisqu'elles se confirment les unes les autres. En outre, dans le cas d'un journal télévisé, on doit être capable d'extraire une relation entre deux prises de vue de présentateur quelconques, en considération de leur similarité de contenu. La deuxième raison guidant notre choix de groupes stricts est somme toute peu différente. Il s'agit de ne pas propager certaines relations pouvant exister entre une prise de vue de présentateur et une prise de vue autre. Ces relations sont par exemple établies lorsque la transition entre le présentateur et la prise de vue suivante est progressive, de type fondu ou balayage. Dans l'hypothèse d'existence de telles relations, la constitution de groupes larges conduirait au regroupement à la fois des prises de vue de présentateur, mais aussi par propagation des transitions présentateur-reportage et de la première prise de vue de chaque reportage, elle-aussi en relation avec la transition intermédiaire.

Par la suite, et pour ces deux raisons, nous proposons donc les résultats de la détection de groupes maximaux sous la forme de groupes stricts.

Concrètement une évaluation de ce deuxième critère est fournie par le rapport, pour un groupe de relations donné, entre le nombre de relations contenues dans ce groupe et le nombre de relations contenues dans le groupe maximal.

Critère de sélection des groupes de relations contenant des prises de vue en début et en fin de document. Pour l'établissement de ce deuxième critère, rappelons qu'il est nécessaire de faire l'hypothèse que le document vidéo traité correspond au journal télévisé entier. En d'autres mots, si le document ne représente qu'une partie d'un journal télévisé, ce critère ne sera pas validé positivement, et la décision finale qu'il s'agit d'un journal télévisé sera pénalisée. Nous supposons d'autre part que le générique du journal ne donne lieu qu'à l'extraction d'une seule prise de vue. Un générique formé de

plus d'une prise de vue ou générant la détection de fausses alarmes invalide ce troisième critère.

Il s'agit ici d'attribuer une probabilité à chacun des groupes de relations extraits précédemment en fonction de leur contenance ou non d'une prise de vue en début et d'une autre en fin de journal. Le critère choisi est alors très simple ; pour chaque groupe, il oscille entre trois valeurs :

- 1.0 (valeur maximale), si le groupe contient à la fois, la première ou la seconde, et, la dernière ou l'avant-dernière prises de vue du document.
- 0.5, si le groupe vérifie l'une seulement des deux conditions ; une des deux premières ou une des deux dernières prises de vue appartiennent au groupe.
- 0, si aucune des conditions n'est vérifiée. Aucune des deux premières, ni des deux dernières prises de vue n'appartient au groupe.

Critère de détection du visage de présentateur. Ce critère est sans doute qualifié à tort de "détection de visage" de présentateur. Il ne s'agit pas en effet d'un détecteur de visage, mais d'une détection d'une région de couleur peau, d'une taille et à une position telles que, dans notre application, il ne puisse s'agir que du visage du présentateur.

Notons par ailleurs qu'aucune utilisation de la forme usuelle d'un visage n'est faite. Par la suite, nous verrons que le critère élaboré est suffisant pour cette détection de prises de vue de présentateur. Avec l'ajout de contraintes supplémentaires de type forme par exemple, ce critère pourrait toutefois servir de base à un véritable détecteur de visages, outil essentiel en indexation d'images.

La première étape de calcul de ce critère correspond directement à l'utilisation de l'outil de détection de régions de couleur peau, présenté dans le chapitre 6, sur les images clés de chacune des prises de vue du document vidéo.

Il s'agit ensuite de sélectionner, parmi les composantes connexes de couleur peau extraites pour une image clé donnée, celles susceptibles, de par leur surface et leur positionnement, d'avoir une adéquation maximale avec les caractéristiques moyennes extraites d'images de ce type (présentant le buste d'une personne en plan moyen). Cette adéquation est alors mesurée sous la forme d'une probabilité pour chaque composante connexe, calcul que nous détaillons à présent.

En ce qui concerne la place de chaque composante connexe dans l'image, on définit, de manière empirique, un cadre au centre de l'image, dans lequel la composante connexe doit se trouver. Ce cadre et ses valeurs limites sont donnés sur le schéma 7.8.

La probabilité d'être plus ou moins bien placé, notée $P_{position\_correcte}$, sera alors mesurée par le rapport entre l'aire de la partie de la composante connexe à l'intérieur du cadre et son aire totale, aboutissant ainsi à une équation similaire à l'équation 7.2 de mesure de probabilité de placement correct des incrustations d'une image clé à une autre :

$$P_{position\_correcte}(CC) = \frac{\text{Aire}(CC \cap Cadre)}{\text{Aire}(CC)} \tag{7.4}$$

où CC désigne la composante connexe testée.

Quant à la taille de la composante connexe, elle intervient également de la façon suivante. On calcule *TauxRecouvrement*, la proportion de la taille de la composante connexe par

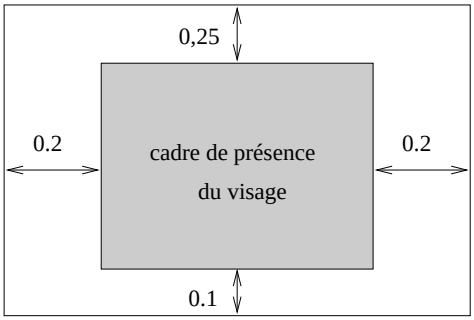


FIG. 7.8: Représentation du cadre dans lequel la composante connexe de peau doit se trouver pour avoir une forte probabilité d'appartenir au journaliste ou à l'invité.

rapport à la taille totale de l'image :

$$TauxRecouvrement(CC) = \frac{\text{Aire}(CC)}{\text{Taille}(image)} \tag{7.5}$$

Ce premier pourcentage est ensuite comparé à deux valeurs de seuil, fixées respectivement à 3% et 20% de façon expérimentale. Une nouvelle probabilité, $P_{taille\_correcte}$, est alors définie en fonction de la taille de la composante connexe, conformément à la fonction d'appartenancede la figure 7.9.

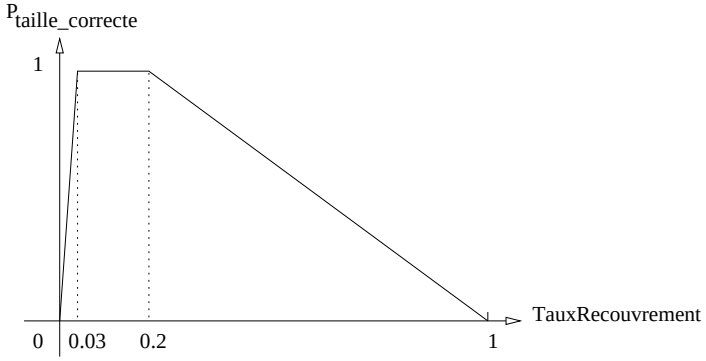


FIG. 7.9: Courbe de représentation de $P_{taille\_correcte}$ en fonction de *TauxRecouvrement*.

Cette fonction se traduit mathématiquement par :

$$f(x) \quad = \quad \begin{cases} 1 & \text{si } 0.03 \leq x \leq 0.2 \\ \frac{x}{0.03} & \text{si } x < 0.03 \\ \frac{1}{0.8}(1-x) & \text{si } 0.2 < x \end{cases} \tag{7.6}$$

avec $P_{taille\_correcte} = f(TauxRecouvrement)$.
L'expression de la probabilité de présence de peau résultante, P_{peau}, est alors tout

simplement la moyenne de $P_{taille\_correcte}$ et $P_{position\_correcte}$:

$$P_{peau} = \frac{P_{taille\_correcte} + P_{position\_correcte}}{2} \tag{7.7}$$

Une telle probabilité est évaluée pour chaque composante connexe d'une image clé donnée ; seule la valeur maximale est alors conservée comme représentative de la composante connexe d'intérêt. Enfin la moyenne des probabilités obtenues pour chaque image clé est affectée à la prise de vue.

Critère de détection d'un fond immobile. De même qu'un cadre a été défini de manière expérimentale comme région dans laquelle la composante connexe de visage doit se trouver, un autre cadre (cf. figure 7.10) est construit, toujours par expérience sur des images de présentateur, pour sélectionner la zone dans laquelle l'immobilité du fond sera testée.

Dans la zone extérieure au cadre, on applique le critère de différence d'images par distance dans l'espace des couleurs, utilisé lors de la détection de transitions de type coupures dans le chapitre 3, entre la première et la dernière images clés de la séquence. La probabilité, notée P_{fond}, est alors directement proportionnelle à la valeur du critère :

$$P_{fond} = \text{Distance}(\text{Image clé 1}, \text{Image clé 2}) \tag{7.8}$$

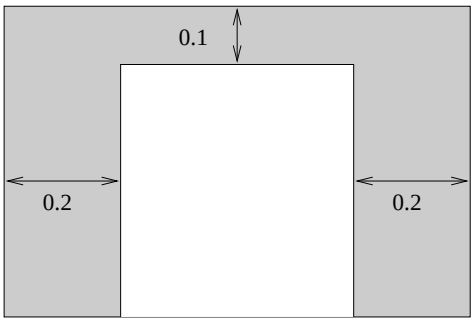


FIG. 7.10: Représentation de la zone dans laquelle on détermine si le fond de l'image est immobile.

Dans le cas où la prise de vue n'a qu'une seule image clé, le critère est mis à 1. On obtient ainsi directement une valeur de critère de fond immobile par prise de vue.

Appliqués à l'ensemble du document, ces quatre critères fournissent quatre valeurs de probabilités par groupe de relations extrait.

Plus qu'une seule d'entre elles, c'est l'ensemble des caractéristiques précédentes qui permet d'aboutir à la conclusion de présence d'un événement de type présentateur. En effet, une seule de ces contraintes, prise indépendamment, n'est pas suffisante pour aboutir à la conclusion de présence de présentateur, de par notamment son faible niveau d'extraction d'information sémantique. Cependant une réponse positive à l'ensemble de ces contraintes de bas niveau renforce, au contraire, la probabilité d'avoir affaire à cet événement qui est déjà d'un niveau sémantique élevé. Nous avons donc, dans cette application, une nouvelle illustration

du principe de base que nous avons constamment mis en œuvre dans nos travaux : accéder directement à de l'information d'un niveau sémantique élevé dans un document n'est pas une tâche aisée ; par contre, des traitements plus simples combinés à des règles logiques de cohérence permettent de contourner cette difficulté, et de répondre positivement à des questions sémantiques, à partir d'outils de bas niveau.

La fusion nécessaire des quatre critères extraits est réalisée par l'intermédiaire d'une simple moyenne des quatre probabilités obtenues. Il est par ailleurs envisageable, lors de cette fusion, de pondérer chaque critère par un facteur approprié, en fonction des variantes qui peuvent exister entre les divers journaux télévisés, dans l'hypothèse bien sûr où on a accès à l'information supplémentaire, par exemple, de provenance du document vidéo.

Mais pour notre application, une réponse positive à chacun d'entre eux et le calcul de la probabilité résultante par simple moyenne permettent la sélection des prises de vue de présentateur avec une probabilité élevée, comme l'illustrent la figure 7.11 et le tableau 7.12, contenant respectivement le groupe de relations stricts de probabilité maximale extrait pour le document *jtv1* de la base de données, et les probabilités obtenues pour l'ensemble des groupes extraits. Notons que les deux groupes extraits avec des probabilités élevées de plus de 80% correspondent effectivement à des prises de vue de présentateur. D'autre part, tout groupe dont la probabilité finale atteint 60% contient également une majorité de prises de vue de présentateur, associées à quelques prises de vue parasites.

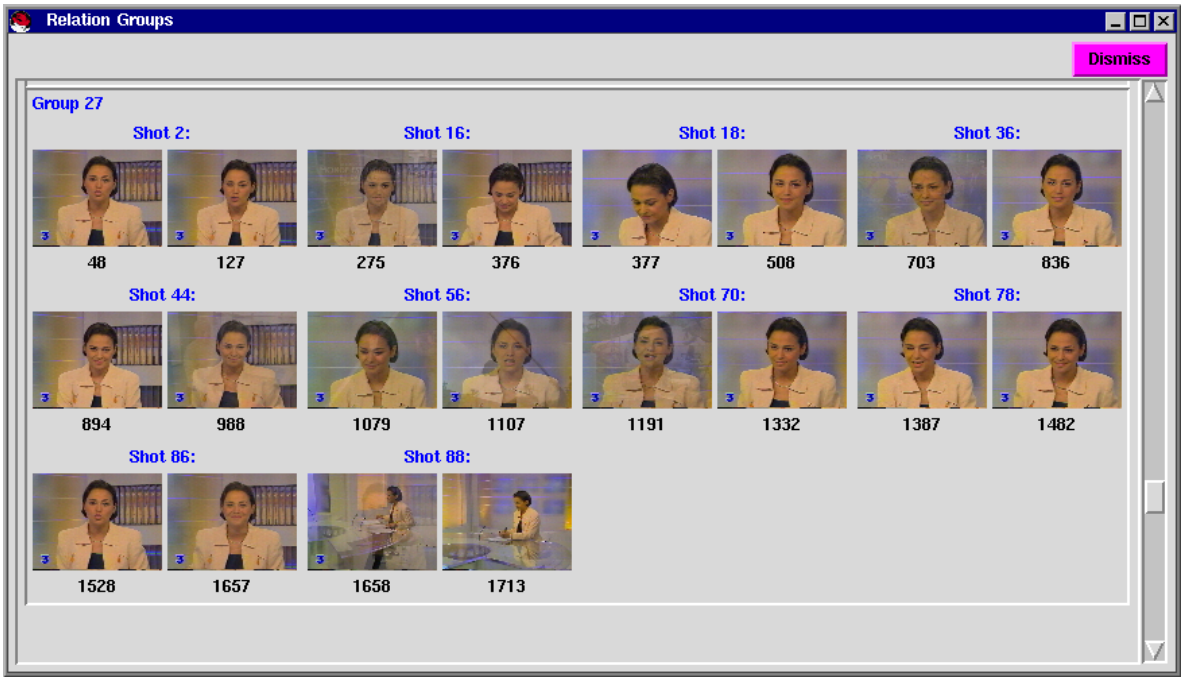


FIG. 7.11: Groupe strict de relations ayant la probabilité maximale de correspondre au groupe de prises de vue de présentateur, pour le document *jtv1*.

Terminons cette description par une dernière remarque sur le fait que cette application n'a été testée dans son entier que sur un seul document vidéo complet de journal télévisé. Cependant, l'algorithme appliqué aux autres séquences de la base de test n'engendre aucun faux positif et fournit à chaque fois des probabilités cohérentes avec le contenu des documents.

Groupe	**Détection de la peau**	**Immobilité du fond**	**Nombre de relations**	**Relations début-fin**	**% total**
0	54.8	89.1	20.0	0.0	41.0
1	62.6	85.6	20.0	0.0	42.0
2	65.6	82.4	20.0	0.0	42.0
3	65.9	92.6	20.0	0.0	44.6
4	57.7	84.5	40.0	0.0	45.5
5	72.1	92.7	20.0	0.0	46.2
6	76.7	89.0	20.0	0.0	46.3
7	63.9	93.0	30.0	0.0	46.7
8	76.7	83.7	30.0	0.0	47.6
9	94.0	81.1	20.0	0.0	48.8
10	66.2	92.8	40.0	0.0	49.7
11	66.1	92.8	40.0	0.0	49.7
12	94.0	89.0	20.0	0.0	50.7
13	82.4	82.0	40.0	0.0	51.1
14	73.9	91.9	40.0	0.0	51.4
15	70.2	96.0	40.0	0.0	51.5
16	83.6	84.6	40.0	0.0	52.1
17	78.3	91.4	40.0	0.0	52.4
18	74.6	88.6	50.0	0.0	53.3
19	62.7	82.8	50.0	50.0	61.4
20	95.9	87.1	20.0	50.0	63.3
21	70.1	87.6	70.0	50.0	69.4
22	73.1	85.8	70.0	50.0	69.7
23	64.4	88.6	80.0	50.0	70.8
24	93.4	76.6	20.0	100.0	72.1
25	66.7	85.9	100.0	50.0	75.6
26	63.8	89.2	80.0	100.0	83.2
27	**66.2**	**86.4**	**100.0**	**100.0**	**88.1**

FIG. 7.12: Valeurs des quatre critères pour les groupes stricts de relations obtenus pour le document *jtv1*.

Appliqué sur les documents *Interview* et *Lille* dont les structures linéaires extraites sont présentées respectivement dans les figures 7.1 et 7.6, l'outil de détection de présentateur conduit même à une détection correcte, malgré le fait que ces documents ne soient que des extraits de journaux télévisés. Cette dernière remarque est une preuve supplémentaire de la robustesse de notre outil, obtenue par la fusion de plusieurs critères bas niveau : si un critère n'est pas pleinement vérifié (en l'occurence pour ces deux documents, la répétition des prises de vue de présentateur et leur présence en début et fin de document), les autres critères restent cependant suffisamment pertinents, pour assurer une détection correcte.

Malgré ce nombre pour le moins restreint de tests, les résultats obtenus laissent à penser que la technique employée de sélection de critères de niveaux assez bas, et donc relativement simples à mettre en œuvre, combinés en une probabilité unique résultante, correspond à une voie de réponse correcte pour des questions de ce type. Ceci est par ailleurs corroboré par les travaux existants dans ce domaine, parmi lesquels ceux de Fisher *et al.* [45], qui proposent une reconnaissance du genre des films, en fonction de caractéristiques particulières et simples extraites des documents.

Une fois les groupes de relations stricts extraits, le groupe ayant la probabilité maximale est étiqueté groupe des prises de vue de présentateur, engendrant ainsi des relations entre ces prises de vue particulières et des relations par complémentation entre toutes les prises de vue comprises entre deux prises de vue de présentateur. Pour des raisons de taille des données et de facilité d'illustration, nous ne présentons pas le résultat de cette mise en relation sous la forme de graphe relationnel, sur le document *jtv1*. Un autre exemple plus court est proposé sous la forme de la figure 7.13, pour le document Lille. Pour ce document, il apparaît que certaines relations sont redondantes avec celles établies du fait de l'emploi de transitions spécifiques.

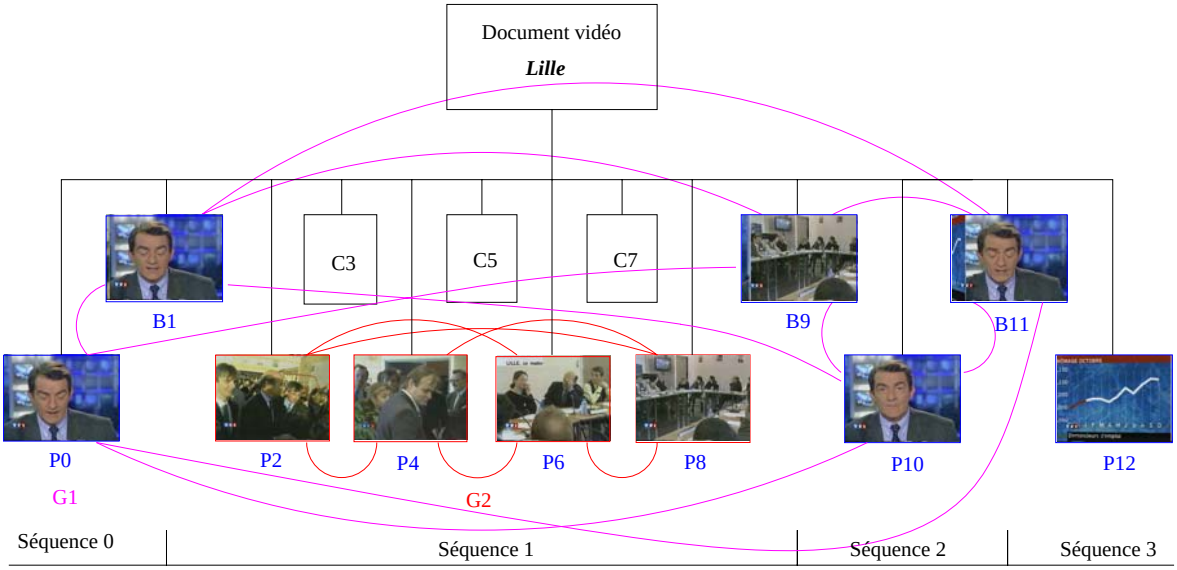


FIG. 7.13: Groupes de relations stricts obtenus par la détection de prises de vue de présentateur et découpage en séquences correspondant, pour le document *Lille*. Seules les premières images clés de chaque prise de vue sont représentées. Le document est composé de sept prises de vue séparées par trois balayages B1, B9 et B11 et trois coupures C3, C5 et C7.

Ceci termine l'exposé des outils de mises en relation développés dans le cadre de nos

travaux. Il convient à présent de conclure par un résumé de ces différents outils, conduisant à l'élaboration de la structure relationnelle d'un document vidéo.

7.5 Conclusion

Nous venons d'exposer quatre techniques différentes d'établissement de relations entre les diverses entités d'un document vidéo, applicables dans le cas général à un document vidéo quelconque pour certaines, ou bien au contraire dédiées à des documents spécifiques comme par exemple les journaux télévisés.

Quel que soit leur caractère, général ou spécifique, les relations obtenues partagent la même caractéristique d'être extraites à partir de traitements d'images relativement bas niveau et de l'application de règles logiques, exprimant le caractère sémantique proprement dit de la relation.

La notion d'"intelligence" que nous plaçons dans l'élaboration de la structure relationnelle d'un document vidéo ne se situe donc pas dans l'extraction directe des images d'informations de très haut niveau, mais dans la conjugaison d'informations bas niveau et de règles logiques, comme le décrit le schéma 7.14.

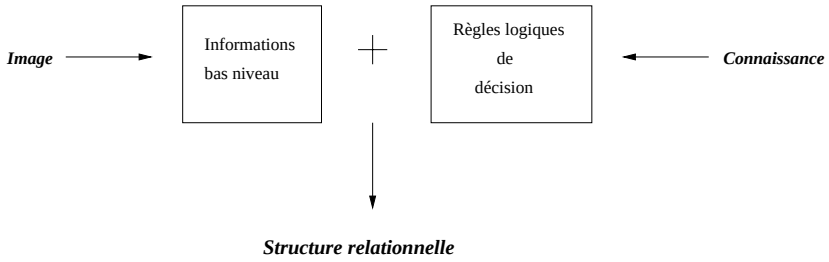


FIG. 7.14: Couplage entre les informations bas niveau issues du traitement d'images et des règles logiques de décision dérivant directement de la connaissance a priori sur un type de document, permettant l'établissement de la structure relationnelle.

Bien sûr la structure relationnelle, dont nous proposons quelques exemples dans ce chapitre, est loin d'être complète pour deux raisons principales qui constituent deux perspectives importantes de poursuite de nos travaux.

Tout d'abord, les exemples proposés ne représentent de façon évidente qu'un échantillon illustratif, mais restreint, de l'ensemble des relations qu'il est possible d'extraire. D'autres voies parmi lesquelles la reconnaissance de personnes, la classification scène de jour / scène de nuit [86], scène extérieure / scène intérieure (dont on a présenté un outil de détection dans le chapitre 6), l'extraction et la reconnaissance d'objets d'intérêt [11], par exemple par comparaison d'arbres de segmentation contenant des informations supplémentaires de forme, texture, position, etc. [48, 80], ou bien encore par l'utilisation d'histogrammes 3D, mériteraient ainsi d'être explorées, de façon à fournir des groupes de relations supplémentaires.

Attardons-nous quelques instants sur ce dernier exemple, qui à notre avis s'inscrit tout à fait dans la lignée des divers outils présentés dans ce mémoire. Par histogramme 3D, on entend une image 2D représentant sous forme concise l'ensemble des histogrammes d'un type donné (soit de luminance, soit de teinte, etc.) des images d'une même séquence d'images. Cette image 2D contient sur chacune de ses lignes l'histogramme de l'image de la séquence courante, à un

instant donné. L'étude de cette image par des techniques de traitement d'images classiques, par exemple morphologiques, donne accès à l'évolution temporelle de cet histogramme. Une première étude expérimentale a ainsi révélé l'existence de motifs répétitifs dans les histogrammes au cours du temps (par exemple un pic très marqué dans les teintes rouges), significatifs de la présence d'un même objet, mais dans des prises de vue différentes et non nécessairement voisines. Là encore, des traitements d'images simples (sur les images 2D produites), et tout particulièrement les opérateurs de filtrage disponibles en morphologie mathématique semblent tout indiqués pour l'extraction de ces motifs répétitifs, motifs qui contiennent une information de nature sémantique, donc intéressante à extraire, et accessible relativement aisément.

Dans un deuxième temps, et à partir de l'ensemble des groupes de relations résultant des quatre méthodes développées dans ce chapitre et d'autres techniques de mise en relation, il reste à établir ce que nous avons appelé des graphes d'entités en relation, mettant en jeu plusieurs types différents de relations, et non une seule. L'élaboration de tels graphes pour un document vidéo donné permet en effet une caractérisation optimale du contenu sémantique de ce document. C'est en outre sur ces graphes que seront appliqués les processus de recherche, i.e. les requêtes formulées sur une base de données de documents, but final de l'indexation. Notons qu'il est de plus envisageable de construire de tels graphes, non plus à l'intérieur d'un seul document, mais sur plusieurs, de façon à définir des relations inter-documents. Cette étape de construction de graphes, mettant en jeu des types divers de relations de façon à fournir des réponses à des requêtes précises, fera l'objet du chapitre suivant, où nous en détaillerons plusieurs exemples sous forme d'applications de l'ensemble de nos outils.

Comme dernière voie de l'élaboration de la structuration relationnelle, nous souhaitons souligner l'importance de la recherche restant à mettre en œuvre dans le cadre d'un découpage supplémentaire en scènes et séquences, découpage que nous avons présenté dans le chapitre 2, comme étant le dernier niveau de hiérarchie dans la partie linéaire de la structuration. Certaines des méthodes d'extraction de relations que nous venons de détailler fournissent d'ores et déjà des prémices de ce découpage. C'est par exemple le cas de la détection de présentateur ou de transitions d'un type particulier, qui délimitent les séquences, dans le cas des journaux télévisés par exemple.

Cependant ce niveau de découpage est loin d'être achevé et les applications développées au chapitre 8 s'efforceront également de poursuivre cette hiérarchisation.

Chapitre 8

Applications

8.1 Introduction

L'objet de ce chapitre est de proposer, sous la forme de trois applications concrètes, des exemples du processus entier de structuration, qu'elle soit linéaire ou relationnelle. Ces applications sont également l'occasion de mettre en pratique, sur les entités extraites, les outils de caractérisation sémantique détaillés dans l'ensemble des chapitres précédents.

Nous proposons ainsi une première application de suppression des fausses alarmes ou d'erreurs de structuration, issues de l'étape de structuration linéaire. Cette application sera détaillée dans le paragraphe 8.2.

Les journaux télévisés étant le type de document vidéo que nous avons privilégié dans le cadre de nos travaux, les deux applications suivantes leur sont donc plus spécifiquement dédiées. La section 8.3 présente ainsi un outil de détection d'interviews télévisés.

Cette seconde application sera suivie, dans la section 8.4, d'un exemple du processus global de structuration et d'indexation proposé dans l'ensemble de ce mémoire, pour un document vidéo de type journal télévisé. Cette dernière application agira ainsi comme un résumé concret de l'ensemble des outils dont nous disposons à présent et sera l'occasion une fois encore de montrer qu'il est possible d'atteindre un niveau sémantique déjà élévé, par l'application de nos outils.

Les deux premières applications seront bien sûr illustrées, dans la mesure où il est possible d'illustrer des séquences temporelles à partir de support papier fixe (!), à partir d'exemples choisis parmi les documents de notre base de données. Du fait de la longueur du fichier vidéo choisi et pour contourner la difficulté d'illustrer des séquences à partir d'images fixes, nous proposerons la troisième et dernière application sous la forme de pages HTML, stockées soit sur un CD, soit à l'adresse suivante :

http ://cmm.ensmp.fr/~demarty/THESE/chap8/

Mais commençons à présent par ce que nous avons appelé *validation de la structure linéaire*.

8.2 Validation de la structure linéaire

La structuration linéaire temporelle, mise en œuvre dans les chapitres 3 et 4, n'est pas exempte de fausses alarmes entraînant des erreurs de découpage, ou un mauvais étiquetage de certaines entités. Si de telles situations étaient relativement rares suite à la détection de

coupures, le fort taux de fausses alarmes (65.4%) de l'algorithme de détection de transitions chromatiques telles que les fondus ne peut rester sans solution.

Aussi, cette première application très directe des résultats de la hiérarchisation introduite au chapitre précédent est-elle l'occasion de revenir sur la structure linéaire établie, pour la valider, ou au contraire l'invalider. Nous détaillons ici trois cas de découpages erronés, que nous sommes en mesure de corriger :

- fausse détection de coupures ;
- détection de flash comme prise de vue à part entière ;
- erreur de détection de fondu.

Pour chacune de ces trois situations, des règles simples de cohérence, directement issues des relations qu'il a été possible d'établir entre les entités, sont à la base de toute modification de la structure linéaire. Nul doute que d'autres cas particuliers de fausses alarmes, non décrits ici, puissent être également corrigés par l'utilisation de règles du même ordre.

FIG. 8.1: Structuration linéaire issue du macro-découpage temporel pour le document *vieux tennis*. Une fausse détection a eu lieu entre les prises de vue 8 et 10.

Fausse coupure. Commençons donc par le cas d'une fausse détection de coupure dans un document vidéo, dont nous donnons un exemple dans la structure linéaire de la figure 8.1. Pour le document présenté, une erreur de découpage s'est manifestement produite entre les prises de vue P_8 et P_{10}, avec pour conséquence l'apparition d'une coupure n'ayant

pas lieu d'être. Le diagnostic, que nous effectuons après découpage, pour décider de la présence d'une erreur dans ce cas, est la similarité de contenu entre les prises de vue P_8 et P_{10}. Cette notion correspond exactement à notre outil d'établissement de relations entre prises de vue à partir des images clés, détaillé au paragraphe 7.3.1. Et effectivement, la recherche de relations entre prises de vue sur cet exemple exhibe une relation entre P_8 et P_{10}. On aboutit donc tout naturellement à la règle suivante :

$$P_i \mathcal{R} P_{i+2} \Rightarrow \begin{cases} \text{Suppression de la coupure entre } (P_i, P_{i+2}). \\ \text{Modification de l'étiquette de } P_i \text{ et } P_{i+2} \text{ de } \textbf{prises de vue} \\ \text{à } \textbf{morceaux} \text{ d'une même prise de vue.} \end{cases} \quad (8.1)$$

où $\mathcal{R}$ représente la relation établie par comparaison des images clés.
Une fois appliquée à l'exemple de la figure 8.1, on aboutit à la nouvelle structuration linéaire présentée à la figure 8.2. L'application de cette première règle de validation donne lieu à la correction de 7 fausses coupures sur l'ensemble de notre base de données (effets de bords compris), ce qui représente une baisse du taux de fausses alarmes de 3.7% à 3%.

FIG. 8.2: Structuration linéaire après validation à partir des relations entre prises de vue établies par comparaison du contenu des images clés. La fausse détection entre les prises de vue 8 et 9 a été corrigée et leur étiquette modifiée en **morceau de prises de vue**.

Un tel algorithme de suppression de fausses coupures est également responsable de la concaténation de deux prises de vue voisines et en relation, même si la coupure entre les

deux est réelle. Sur l'ensemble de la base de données, on réalise ainsi 8 concaténations de prises de vue parasites. Cependant deux seulement provoquent la réunion de prises de vue de contenus très différents. Dans les autres cas on assiste à la réunion de prises de vue proches sémantiquement ; quelques exemples sont fournis dans la figure 8.3.

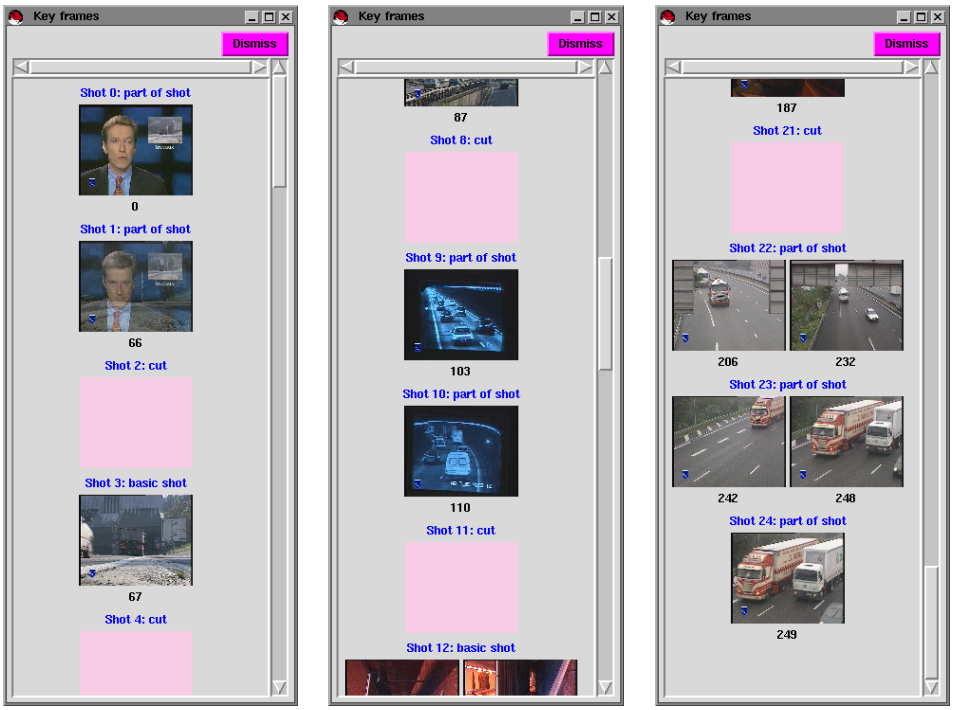


FIG. 8.3: Exemples de concaténation parasite de deux prises de vue voisines et en relation, mais distinctes, du fait de la correction des fausses coupures. Extraits de la structure du document *travaux*.

Detection de flash . Lors du bilan de l'algorithme de détection de coupures (section 3.2.8), nous avions relevé la présence de fausses alarmes dues aux événements particuliers des journaux télévisés que sont les flashes d'appareil photo. Si détecter les flashes comme prises de vue à part entière constitue une erreur manifeste, elle peut être corrigée par un changement d'étiquette de l'entité flash, qui passe de **prise de vue** à **morceau de prise de vue**. Une telle modification a déjà été exposée, dans le cadre du micro-découpage temporel, et plus particulièrement dans la section 4.2.5, aussi nous ne nous attarderons pas sur ce deuxième exemple de validation de la structure linéaire. Rappelons simplement la règle qui est alors utilisée. On considère trois prises de vue successives P_i, P_{i+2} et P_{i+4}, séparées par deux coupures C_{i+1} et C_{i+3}. Suite à l'étape de hiérarchisation, et dans l'éventualité où P_{i+2} est un morceau de prise de vue de type flash, l'établissement de relations à partir de la comparaison du contenu des images clés doit aboutir à la conclusion d'une relation entre P_i et P_{i+4}. Dans ce cas, et si de plus P_{i+2} vérifie les caractéristiques d'un flash proposées précédemment, on modifie l'étiquetage de la façon

suivante :

$$\left\{\begin{array}{l} P_i \mathcal{R} P_{i+4} \\ P_{i+2} \text{ vérifie les caractéristiques d'un } \textit{flash} \end{array}\right. \Rightarrow \left\{\begin{array}{l} \text{Suppression des coupures entre } (P_i, P_{i+2}) \text{ et} \\ (P_{i+2}, P_{i+4}). \\ \text{Modification de l'étiquetage de } P_i,\ P_{i+2} \text{ et} \\ P_{i+4} \text{ de } \textbf{prises de vue} \text{ à } \textbf{morceaux} \text{ d'une} \\ \text{même prise de vue.} \\ \text{Ajout de l'étiquette } \textbf{flash} \text{ pour } P_{i+2}. \end{array}\right. \tag{8.2}$$

où $Carac.(P_i)$ représente l'ensemble des caractéristiques extraites de la prise de vue P_i. Pour des illustrations sur des documents test, nous renvoyons le lecteur au paragraphe 4.2.5 du chapitre 4. En terme de taux de fausses alarmes, on aboutit après correction des événements correspondant à des flashes à une baisse de 3% (taux après correction de fausses coupures) à 0.7%.

Erreur de détection de fondu. Ce troisième exemple de retour et validation de la structure linéaire établie par macro-découpage est sans aucun doute le plus intéressant, dans la mesure où il permet de corriger le taux élevé de fausses alarmes de l'algorithme de détection de fondus, taux qui représentait le point faible de l'algorithme. Une fois encore les règles mises en œuvre pour mener à bien la correction des erreurs de détection sont basées sur l'examen des caractéristiques d'une transition de ce type.

Sauf dans la situation très particulière d'un fondu entre deux prises de vue du contenu similaire (cas rare en pratique), on est en effet en droit d'attendre d'une part une détection de changement entre le début et la fin de la transition de type fondu. D'autre part, il est naturel de détecter des relations, toujours par comparaison des images clés, entre la prise de vue avant fondu et le fondu, et entre la prise de vue après et le fondu. Si l'une ou l'autre de ces caractéristiques ne sont pas présentes, on est naturellement amené à reconsidérer l'étiquette **fondu** attribuée à la prise de vue considérée. Ceci se traduit par la règle suivante, où P_i, P_{i+1} et P_{i+2} sont trois prises de vue successives :

$$\left\{\begin{array}{c} P_{i+1} = fondu \\ et \\ P_i \ \mathcal{R} P_{i+1} \\ ou \\ P_{i+1} \ \mathcal{R} P_{i+2} \\ ou \\ \text{Changement}(P_{i+1}) = non \end{array}\right\} \Rightarrow \left\{\begin{array}{l} \text{Ajout de coupures entre } (P_i, P_{i+1}) \text{ et entre } (P_{i+1}, P_{i+2}). \\ \text{Modification de l'étiquette } \textbf{fondu} \text{ de } P_{i+1} \\ \text{en } \textbf{prise de vue}. \end{array}\right. \tag{8.3}$$

Il est même envisageable de fusionner soit deux des trois prises de vue, ou l'ensemble des trois, si la présence de relations l'autorise. On revient alors au cas de suppression de fausses coupures, décrite dans le premier point de ce paragraphe.

Si l'application directe de cette règle corrige une très grande majorité des fausses alarmes, elle conduit cependant également à la suppression de fondus réels, mais pour lesquels aucun changement interne n'a été détecté, ou bien des relations avec les prises de vue voisines manquent. La règle 8.3 conduisant donc à un ensemble de contraintes trop fortes, nous l'avons assouplie en pratique, de façon à conserver un maximum des fondus

réels détectés, au détriment d'un nombre moindre de fausses alarmes corrigées. Cette nouvelle règle, qui correspond à rechercher les fausses alarmes parmi les prises de vue sans relation avec leurs voisins, ou entourées de deux voisins en relation, s'exprime alors de la façon suivante :

$$\left\{ \begin{array}{c} P_{i+1} = fondu \\ et \\ \left\{ \begin{array}{c} P_i \ \mathcal{R} P_{i+1} \\ et \\ P_{i+1} \ \mathcal{R} P_{i+2} \end{array} \right\} \\ ou \\ P_i \mathcal{R} P_{i+2} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \text{Ajout de coupures entre } (P_i, P_{i+1}) \text{ et entre } (P_{i+1}, P_{i+2}). \\ \text{Modification de l'étiquette } \textbf{fondu} \text{ de } P_{i+1} \\ \text{en } \textbf{prise de vue}. \end{array} \right. \tag{8.4}$$

Un exemple de la correction de détection de fondus est présenté dans la figure 8.4. Sur l'ensemble de la base de données de test, l'utilisation de cette deuxième règle de validation permet une baisse du taux de fausses alarmes de 65.4% à 33.3% (soit la correction de 23 fausses alarmes sur 32) ; un seul et unique fondu réel, pour lequel aucune relation n'a pu être établie avec les prises de vue de par et d'autre, est alors supprimé.

Pour chacun de ces trois exemples, la baisse effective du taux de fausses alarmes est une preuve de l'amélioration du découpage linéaire effectué. Comme nous le notions en introduction de ce paragraphe, d'autres règles logiques de cohérence entre caractéristiques des diverses entités extraites d'un document vidéo peuvent être établies de façon similaire. Il est ainsi envisageable, dans le cas d'une prise de vue qualifiée de "avec changement" par l'outil détaillé dans la section 4.4, de revenir sur les images de cette prise de vue, de façon à déterminer si aucune transition n'aurait été oubliée. Une étude approfondie du mouvement peut alors intervenir pour départager les cas d'oublis de transitions des prises de vue contenant du mouvement. La détermination de perte de transition, obtenue par une telle règle, peut ainsi être le point de départ qui nous manquait afin de déterminer la présence de transitions géométriques. Rappelons en effet, que l'étude du masque de transition ne nous permet une classification de ces transitions, que dans l'hypothèse où les début et fin de transitions sont connus.

Cette dernière remarque, qui n'a pas été mise en œuvre dans le cadre de ce mémoire, est à conserver parmi les perspectives d'amélioration de la structuration obtenue. Avec elle, se termine l'exposé de cette première application concrète de validation de la structuration extraite pour un document vidéo donné. Nous poursuivons à présent par une deuxième application directement liée aux journaux télévisés : la détection d'interviews.

8.3 Extraction d'interviews

Nous proposons pour cette application de réaliser la détection et l'extraction, à l'intérieur d'un document vidéo, d'un ensemble de prises de vue successives, correspondant à un interview d'une personne invitée, par un journaliste télévisé.

Un tel événement *interview* se caractérise trivialement par la présence alternée de deux ou trois plans uniques, le premier représentant le journaliste, le deuxième la personne invitée et le troisième éventuel, correspondant à un plan des deux personnages ensemble. D'autre part, les deux personnes sont prises à une distance moyenne de la caméra.

FIG. 8.4: Structuration linéaire avec fausse alarme de détection de fondu pour la prise de vue 9 (a), et après validation et correction (b) pour le document *nantes*.

Sur la base de ces caractéristiques, l'établissement de relations entre prises de vue grâce à l'etude de leurs images clés permet d'extraire, pour le document vidéo *interview* de notre base de données, dont on présente par ailleurs à titre de référence la structure linéaire établie pour ce document dans la figure 8.5, les relations suivantes :

$$P_0 \mathcal{R} P_6 \quad \text{et} \quad P_4 \mathcal{R} P_8 \tag{8.5}$$

On présente par ailleurs à titre de référence la structure linéaire établie pour ce document dans la figure 8.5.

FIG. 8.5: Structuration linéaire du document vidéo *interview*.

Cette situation correspond à la première caractéristique d'entrelacement de prises de vue au contenu similaire. Le document vidéo *interview* devient donc un candidat éventuel de type interview. Il reste à vérifier la présence de personnes dans chacune des prises de vue. Cette dernière étape est évidemment réalisée grâce au même outil que celui utilisé pour la détection du visage du présentateur du journal télévisé, dans la section 7.4.2. les plans du journaliste ou de la personne invitée présentent en effet des caractéristiques proches, en termes de distance et

position de la caméra par rapport à la personne filmée, de ceux utilisés pour le présentateur. La valeur du critère correspond directement à l'expression de PeauPb, fournie dans l'équation 7.7.

Les régions de couleur peau résultant de l'application de l'outil développé dans le chapitre 6 et sur lesquelles les probabilités d'avoir affaire au visage soit du journaliste, soit de la personne invitée seront calculées par cette équation, pour le document *interview*, sont proposées dans la figure 8.6. Notons l'extraction correcte des visages sur toutes les images clés sauf une. Cette erreur est due à une erreur de positionnement du seuil automatique intervenant dans la classification des pixels couleur et gris, de la transformation HSV améliorée, première étape de l'extraction des zones de couleur peau.

FIG. 8.6: Résultat de l'extraction de régions de couleur peau sur les images clés du document *interview*.

On aboutit ainsi, dans le cadre de l'exemple du document *interview*, aux probabilités par prise de vue résumées dans le tableau 8.1. Notons que les probabilités sont très élevées (proches et même égales à 100%), sauf dans les deux cas des prises de vue 2 et 6. La prise de vue 2 représente en effet le plan plus éloigné contenant les deux protagonistes de l'interview, prise de vue qui ne correspond évidemment pas au modèle d'une seule composante connexe de la taille et à la position choisies ici. Ce troisième plan possible lors d'une interview n'a pas été

modélisé dans le cadre de cette application, dans la mesure où il n'est pas toujours présent. Par ailleurs les résultats obtenus pour les autres prises de vue sont suffisamment caractéristiques pour que la détection d'interview se fasse sans peine, malgré ce choix. La prise de vue 6 pour laquelle une des images clés ne présentait pas une détection correcte des régions de couleur peau possède également une probabilité plus faible, mais tout de même supérieure à 50%, ce qui pour deux images clés dans la prise de vue, dont une sans région de couleur peau, conduit à la conclusion d'une probabilité proche de la valeur maximale de 100% pour l'image clé possédant une détection correcte.

Prise de vue	**Probabilité (%) de détection de peau**
0	100
1	coupure
2	46.5
3	coupure
4	98.1
5	coupure
6	51.5
7	coupure
8	93.6

TAB. 8.1: Probabilités de présence de régions de couleur peau correspondant soit au journaliste, soit à l'invité, obtenues pour le document *interview*.

Encore une fois, aucun de ces critères (prises de vue de contenu similaire entrelacées, présence de régions de couleur peau d'une certaine taille et avec une certaine position), pris seul, n'est suffisant pour caractériser un événement de type *interview*, mais une fusion de l'ensemble s'impose.

Pour notre application, cette nécessité de réponse positive à l'ensemble des caractéristiques est testée successivement : on commence par extraire les groupes de prises de vue entrelacées ; sur ces prises de vue uniquement, on détecte les régions de couleur peau, et sur ces régions on établit des valeurs de probabilités en fonction de leur taille et de leur positionnement. Cette dernière probabilité est alors représentative pour chaque prise de vue de son degré d'adéquation au modèle d'*interview*.

L'application que nous venons de présenter, de détection de groupes de prises de vue correspondant à des interviews, constitue un premier exemple du niveau sémantique qu'il est possible d'atteindre par combinaison de la structuration linéaire et relationnelle établie et de simples règles logiques. Nous proposons donc à présent de poursuivre par l'exposé d'une deuxième application aboutissant à l'extraction d'information véritablement sémantique, mais toujours à partir de critères très simples : la détection de présentateur de journaux télévisés.

8.4 Structuration et caractérisation du contenu d'un journal télévisé

Ainsi que nous l'avons déjà évoqué au début de ce mémoire, les journaux télévisés constituent la première source d'information réutilisée actuellement. Ce type de document associe

en outre, à une structure relativement similaire quel que soit le pays d'origine, un contenu très varié de par la diversité des sujets d'actualité traités. En ce sens, le journal télévisé est souvent choisi comme application, parmi les nombreux travaux existants d'indexation [61, 94, 95, 42].

Nous ne faillons pas à la règle, puisque nous avons également sélectionné ce type de document tout au long de ce mémoire et pour cette dernière application, à travers laquelle nous visons une illustration complète de la structuration et de la caractérisation du contenu que nous sommes en mesure d'obtenir pour un tel document pris en son entier.

Notre document original n'est cette fois-ci pas issu de notre base de données test, mais de la base de données établie dans le cadre de l'élaboration de la norme MPEG7 (*MPEG7 content set*). Il s'agit du fichier *jornaldanoite2* présent sur le CD $n°15$, qui reprend un journal de la télévision portugaise, RTV.

Afin de rendre compte au mieux du caractère "en mouvement" d'un tel document, ainsi que pour tenir compte du volume des données originales (49 minutes de vidéo), nous avons choisi de ne pas utiliser pour cette dernière application le support papier, mais de mettre à la disposition du lecteur, toutes les ressources multimédia actuellement disponibles. Pour cette raison nous ne rapportons ici que les grandes lignes des traitements effectués sur le document, ce dernier, ainsi que les résultats obtenus, étant disponibles sur le CD joint à ce mémoire, ou bien à l'adresse internet suivante : http ://cmm.ensmp.fr/∼demarty/THESE/chap8/.

Les pages HTML disponibles comportent alors :

- un récapitulatif de ce que nous venons d'énoncer et des différents traitements effectués ;
- les caractéristiques générales du document ;
- les résultats des traitements suivants :
  - étude du masque de transition ;
  - détection de peau ;
  - détection d'incrustations ;
  - détection de texte ;

  Du fait du volume des données, cette première partie des résultats n'est réalisée que sur quelques images clés originales.
- le résultat automatique de la structuration linéaire et relationnelle, présentée sous une forme plus "interactive", ainsi qu'un découpage en reportages.

8.5 Conclusion

Ce chapitre a été l'occasion d'illustrer dans le cadre d'applications d'un niveau sémantique plus élevé, l'ensemble des outils bâtis et présentés dans ce mémoire. Toutes ont montré la puissance et la robustesse des algorithmes élaborés, qui tout en restant à des niveaux d'extraction d'information assez bas, permettent de répondre à des questions complexes. Nous ne nous attarderons pas plus longtemps ici sur les conclusions de nos travaux, le chapitre suivant ayant pour rôle de résumer nos apports dans le domaine de l'indexation de documents vidéo, et d'apporter des perspectives de poursuite et d'amélioration de la structuration que nous sommes à présent en mesure de construire.

Chapitre 9

Conclusion

Face au problème de l'indexation de documents vidéo, nous avons proposé, tout au long de ce mémoire, de nombreux outils de structuration d'un document, structuration qui intervient comme la première étape du processus d'indexation. Ce problème ayant suscité de multiples travaux, il convient à présent de situer les techniques que nous venons de présenter, en rappelant leurs caractéristiques principales qui constituent aussi, bien évidemment, leurs points forts.

9.1 Apports de cette thèse

Au terme de ce document, nous disposons de la description d'une double structure comme support de la représentation du contenu sémantique d'un document vidéo. Cette structure complète, tant linéaire que relationnelle se décompose en plusieurs étapes autour desquelles s'est organisé ce mémoire :

- une structuration linéaire à la fois temporelle - cette étape se subdivisant elle-même en un macro et un micro-découpage -, et spatiale,
- et une structuration relationnelle.

Pour l'élaboration de chacune de ces étapes, nous avons proposé une liste fournie d'outils, partageant les mêmes qualités de simplicité de mise en œuvre et de rapidité d'exécution, conjuguées à des taux de détection et des niveaux de résultats tout à fait satisfaisants. Dans un domaine où égaler le temps réel n'est plus suffisant, si on considère la multitude des documents déjà existants (il s'agit en effet d'indexer un siècle de documents !), cette efficacité est essentiellement due au choix de privilégier des traitements d'images simples couplés tantôt à des opérateurs puissants de morphologie mathématique, parfois spécialement élaborés pour notre application, tantôt à des règles logiques de décision, appartenant au domaine de l'intelligence artificielle. En ce sens, ce pari de n'effectuer que des traitements simples, combinés entre eux et/ou avec des règles logiques, plutôt que d'élaborer directement des traitements complexes destinés à extraire directement des informations sémantiques elles-aussi complexes, s'est révélé être une méthodologie cohérente, source d'informations sémantiques de haut niveau.

Ce choix de méthodologie nous a par ailleurs permis de montrer qu'il était faisable de réaliser, sinon un outil complet d'indexation, du moins un outil de structuration entièrement automatique et extrêmement rapide (plus que le temps réel).

De façon plus particulière, les apports de nos algorithmes propres à chaque étape de la structuration sont les suivants.

Structuration linéaire temporelle : macro-découpage
Cette première étape consiste en un découpage en prises de vue d'un fichier vidéo quelconque. Deux algorithmes de détection de transitions, basés sur le calcul d'un critère de dissemblance local et sur l'utilisation d'opérateurs de morphologie mathématique, ainsi qu'une technique de fusion des deux structures résultantes ont été élaborés. Parmi les opérateurs morphologiques utilisés, notons la construction d'une variante du chapeau haut-de-forme, le chapeau haut-de-forme inf, et l'utilisation d'un filtrage hiérarchique capables d'extraire de façon robuste et précise les événements correspondant aux transitions de types coupures, fondus et autres transitions chromatiques plus complexes. Nous sommes également en mesure de fournir une classification des transitions de type géométrique, une fois leur localisation réalisée, par le biais des masques de transition dérivant du calcul local du critère de dissemblance. Au nombre des points forts de des deux algorithmes présentés, notons les forts taux de détection atteints et le petit nombre de paramètres nécessaires, paramètres qui restent en outre très peu dépendants des données, permettant ainsi de classer nos algorithmes dans la catégorie des outils automatiques. Par leur simplicité et leur rapidité (temps d'exécution inférieurs au temps réel sans aucune optimisation), ces deux outils laissent en outre envisager une implantation hardware relativement aisée. Seul le fort taux de fausses alarmes de l'algorithme de détection de fondus reste un point méritant plus d'attention.

Structuration linéaire temporelle : micro-découpage
Dans le but d'extraire les entités de types morceaux de prises de vue et images clés, les mouvements de caméra internes aux prises de vue sont détectés par ce qui n'est qu'une retombée directe de l'algorithme de détection des fondus. Si la classification obtenue n'est pas totale (aucune information de direction du mouvement n'est accessible), l'outil proposé ne nécessite aucun temps de calcul supplémentaire. Cette première classification fournit en outre un excellent point de départ, sous la forme d'un support restreint de recherche, pour le lancement d'algorithmes d'estimation du mouvement plus complexes. En parallèle des mouvements de caméra, nous avons de plus prouvé la validité de notre outil de détection et de correction des événements de type flash.

Le processus de sélection des images clés élaboré au cours de nos travaux s'effectue également à moindre coût, puisque les résultats de l'algorithme de détection de coupures sont cette fois-ci réutilisés : les images clés extraites correspondent aux micro-changements intervenant au sein des prises de vue ; en ce sens, nous proposons ici un critère véritable de sélection, reposant sur une base plus solide que le choix arbitraire de la première, la dernière et l'image clé du milieu de chaque prise de vue. Les taux de rappel importants de notre technique indiquent la sélection de la quasi-totalité de l'information sémantique présente, au détriment d'une redondance importante de cette information. En réponse à cet inconvénient nous avons mis en place un outil de détection des changements intervenant dans les prises de vue, qui conduit à une baisse significative du nombre d'images clés redondantes pour chaque prise de vue.

Structuration linéaire spatiale : segmentation et extraction d'objets

Deux voies de structuration spatiale ont été distinguées menant à deux types d'outils. Un processus complet de segmentation couleur morphologique des images clés a ainsi été bâti, visant l'obtention de grandes régions homogènes en couleur. Au nombre des propriétés essentielles de cet outil, citons la généralité du processus quelle que soit l'image d'entrée, son automaticité, et l'originalité des prétraitements effectués, appliqués séparément sur deux classes de pixels distinctes suivant la présence réelle d'une information colorée ou non, et menant à une simplification extrême de l'image à segmenter. Cette classification inédite est automatiquement réalisée grâce à notre opérateur de transformation HSV améliorée. Comme dans toute segmentation morphologique, cette série de prétraitements est suivie d'une étape de gradient couleur, que nous avons voulu différent suivant les deux classes de pixels. Trois cas de décalage de contours ayant été étudiés et catalogués, nous terminons enfin par une segmentation morphologique pour laquelle un nouvel algorithme sur graphe a été bâti, aucun critère d'arrêt n'étant nécessaire du fait de la simplification de l'image de départ.

L'autre volet de la structuration spatiale, mise en œuvre sous la forme de trois outils, est l'extraction d'objets d'intérêt, que sont le texte, les incrustations et les régions d'une certaine teinte comme par exemple la peau. En plus de leur caractère automatique, ces outils prouvent encore une fois l'intérêt et la cohérence de la méthodologie adoptée pour atteindre des informations intéressantes, par l'utilisation de traitements d'images bas niveau, qui en l'occurence sont encore une fois basés sur des opérateurs de morphologie mathématique. S'ils doivent encore être améliorés significativement, notamment dans le but d'obtenir une meilleure robustesse des résultats, ces outils effectuent cependant une première sélection des zones des images sur lesquelles par la suite lancer des traitements dédiés et plus complexes. Ils ont en outre d'ores et déjà prouvé leur validité et leur utilité en l'état, pour l'établissement de la structure relationnelle.

Structuration relationnelle

Cette dernière étape a été mise en œuvre sous la forme d'outils variés, dédiés à chaque fois à l'extraction d'un type de relations donné. Nous sommes ainsi en mesure d'établir des relations essentiellement entre prises de vue, par comparaison de leurs images clés, par détection d'incrustations persistantes, de transitions particulières ou encore de présentateur de journal télévisé. Tous ces outils appuient à nouveau la fonctionnalité de la méthodologie adoptée, à savoir la combinaison de plusieurs critères bas niveau avec des règles logiques de décision. La validation des résultats obtenus se traduit alors par le niveau sémantique contenu dans les graphes de relations et les exemples de hiérarchisation des prises de vue construits.

L'ensemble du processus de structuration établi a au final été mis en œuvre dans diverses applications de complexité croissante, allant de la correction de la structuration linéaire par suppresion des fausses alarmes de détection de transitions, jusqu'à la structuration complète d'un journal télévisé. Cette dernière application fournit alors une illustration concrète du niveau de structuration atteint, permettant déjà de répondre à des requêtes sémantiques relativement complexes.

9.2 Perspectives

Si l'indexation demeure un vaste domaine encore sans solution idéale, les outils, et surtout la méthodologie, que nous venons de présenter auront participé, nous l'espérons, à un certain avancement vers cette solution. Et c'est naturellement que nos résultats s'inscrivent dans des perspectives de poursuite de la recherche dans ce domaine. Ces perspectives se classifient en deux catégories suivant qu'elles s'adressent à des points plus particuliers d'amélioration de nos algorithmes, ou bien qu'elles s'inscrivent dans un cadre plus général.

Au nombre des perspectives plus spécifiques, citons quelques exemples d'amélioration de nos outils de structuration. La localisation de transitions géométriques et la détection d'effets de zooms par exemple, qui n'ont pu être menés à leur terme dans le cadre de nos travaux, méritent en effet d'être poursuivies. De même, d'autres efforts apportés au processus de segmentation ou bien à la robustesse des outils d'extraction d'objets d'intérêt permettraient certainement de les améliorer. Une combinaison de plusieurs critères de texture, couleur, etc. sur lesquels baser la segmentation hiérarchique est par exemple envisageable. Une fois ces améliorations apportées, une étape de validation de l'ensemble des outils créés, grâce à l'utilisation d'un protocole de validation dédié, comme celui développé par exemple dans les travaux de Ruiloba [78], nous semble alors devenir nécessaire.

Si ces exemples propres à chaque algorithme sont essentiels à la poursuite de nos travaux et à l'obtention de résultats toujours plus complets, nous pensons également à d'autres voies plus générales d'extension de notre travail de thèse.

Il nous paraît ainsi important d'étudier l'extension possible de la méthodologie que nous avons adoptée, non plus aux seuls documents vidéo, mais à toute sorte de documents multimédia, la structuration que nous préconisons pouvant également être appliquée et adaptée à ce type de documents. Nous avons évoqué en introduction la situation "amont" des outils que nous avons élaborés par rapport à la norme MPEG7. Rappelons en effet que cette norme ne s'applique qu'aux descripteurs des objets multimédia. Dans ce contexte, il nous paraît intéressant d'étudier dans quelle mesure les entités que nous extrayons correspondent aux descripteurs d'un document vidéo.

Enfin, un dernier axe naturel de poursuite qu'il nous paraît important d'aborder est celui de l'adéquation de la structuration que nous avons bâtie avec le domaine des bases de données et la formulation concrète des requêtes de recherche d'information. Sans entrer réellement dans ce domaine qui n'est pas le nôtre, nous avons déjà évoqué la nécessité de choisir certaines des relations établies au travers du graphe de relations, pour répondre à une requête donnée. Le mécanisme de sélection de ces relations ne nous semble cependant pas trivial, alors que sa maîtrise intervient pleinement dans le processus d'indexation. Son étude nous apparaît donc comme essentielle par la suite.

En conclusion le travail ne manque pas !

Annexe A

Lexique et définitions

Nous regroupons ici, sous forme de lexique, les définitions des diverses notions essentielles à l'indexation.

A.1 Vocabulaire propre à la structuration :

Définition 24. Fichier vidéo *Un fichier vidéo, ou fichier de données, correspond au fichier physique ; il contient une série d'images acquises à une fréquence d'acquisition, un format et une taille donnés. Aucune information de nature sémantique ou structurelle sur l'organisation des images n'est disponible directement.*

Définition 25. Séquence d'images *La séquence d'images est une notion identique à celle de fichier vidéo.*

Définition 26. Document vidéo *Un document vidéo est constitué de prises de vue, scènes et séquences ayant un lien sémantique et pouvant être organisés en une structure hiérarchique. Un journal télévisé par exemple entre dans la catégorie des documents vidéo.*

Définition 27. Prise de vue (*shot*) *Une prise de vue est une série d'images acquises par une seule caméra, d'un seul tenant, et n'ayant donné lieu à aucun montage. Cet ensemble d'images représente une action continue dans le temps et dans l'espace.*

Définition 28. Scène (*scene*) *Une scène est constituée d'un ensemble de prises de vue ayant une même unité de lieu (divers points de vue par exemple).*

Définition 29. Séquence ou segment (*sequence*) *Une séquence regroupe diverses prises de vue et scènes ayant une même unité de sujet (par exemple un reportage).*

Définition 30. Image clé (*keyframe*) *Une image clé est une image représentative d'une partie du contenu informationnel d'une entité (prise de vue ou morceau de prise de vue) du document vidéo. L'ensemble des images clés extraites pour une entité doit permettre de reconstituer tout le contenu sémantique de cette entité, sans redondance, ni perte d'information. Il s'agit d'un ensemble minimal, en ce sens que la suppression d'une image clé conduit automatiquement à la perte d'informations sémantiques.*

Définition 31. Objet *Toute région d'une image ayant une certaine unité et indépendance et possédant une certaine sémantique correspond à la définition d'un objet.*

Définition 32. Macro-découpage temporel *Par macro-découpage temporel, on entend la partie du découpage temporel d'un fichier vidéo ne descendant pas au-dessous de la prise de vue, qui est alors l'unité atomique.*

Définition 33. Micro-découpage temporel *Le micro-découpage temporel consiste en un découpage temporel des prises de vue d'un document vidéo en sous-unités allant jusqu'à l'unité atomique constituée par l'image clé.*

Définition 34. Graphe de relations *Un graphe de relations correspond à la représentation de l'ensemble des relations sémantiques ou syntaxiques, de tout ordre, qui existent entre deux entités quelconques d'un même document.*

Définition 35. Groupe de relations *Un groupe de relations est un sous-graphe de relations pour lequel un seul type de relation existe entre deux entités du sous-graphe.*

Définition 36. Groupe strict de relations *: Un groupe d'entités est un groupe strict de relations si, et seulement si, chacune des entités de ce groupe est en relation avec toutes les autres entités du groupe. Ceci s'exprime aussi de la façon suivante : entre deux entités quelconques du groupe, il existe une relation.*

Définition 37. Groupe large de relations *: Un groupe d'entités est un groupe large de relations si, et seulement si, chacune des entités de ce groupe est au moins en relation avec une autre entité du groupe.*

Définition 38. Relation par image clé *Soient deux prises de vue P_1 et P_2 pour lesquelles on a extrait les ensembles d'images clés respectifs $\{I_1^i\}_{i=d_1...f_1}$ et $\{I_2^i\}_{i=d_2...f_2}$. P_1 et P_2 sont en relation, par comparaison de leurs images clés, si, et seulement si, il existe au moins une paire d'images clés (I_1^n, I_2^m) similaires.*

Le critère de similarité entre les deux prises de vue, noté $C(P_1, P_2)$, a pour expression :

$$C(P_1, P_2) = min_{i,j}(C(I_1^i, I_2^j)),\ d_1 \leq i \leq f_1,\ d_2 \leq i \leq f_2 \tag{A.1}$$

A.2 Vocabulaire des transitions :

Définition 39. Transition (*transition*) *On appelle transition tout effet de montage qui permet de mettre bout à bout deux prises de vue. Il s'agit d'une série artificielle d'images ajoutée lors du montage par un opérateur.*

Définition 40. Coupure (*cut*) *Une coupure est l'effet de transition qui consiste à accoler deux prises de vue successives.*

Définition 41. Fondu et fondu enchaîné (*fade, dissolve*) *Un fondu est une transition graduelle (linéaire) d'une scène vers une image constante (*fade-out *en anglais), ou bien d'une image constante vers une scène (*fade-in *en anglais). Le fondu enchaîné est constitué de la succession d'un* fade-out *et d'un* fade-in*, la transition est alors graduelle d'une scène à une autre. Les fondus et les fondus enchaînés n'agissent que sur la luminance des pixels et non sur leur agencement spatial.*

Définition 42. Balayage (*wipe*) *Lors d'une transition de type balayage, une droite se déplace sur l'image au cours du temps. En amont de cette droite, on conserve l'ancienne prise de vue et en aval, la nouvelle prise de vue apparaît.*

Définition 43. Page tournée (*page turn*) *On appelle page tournée l'ensemble des transitions entre deux prises de vue dont l'effet consiste à donner l'impression qu'on passe d'une prise de vue à la suivante comme si on tournait une page d'un livre.*

Définition 44. Etat de transition *L'état de transition d'un pixel p correspond à la situation particulière de ce pixel qui est modifié du fait de la transition. Cet état de transition du pixel*

possède une durée propre pendant laquelle p est modifié suivant une transformation donnée, modélisant la transition.

Définition 45. Modèle géométrique *Le modèle géométrique peut se définir comme l'ordre dans lequel les pixels passent par l'état de transition. Mathématiquement, si on note G le modèle géométrique d'une transition :*

$$G = \{G_t \,|t \in]t_0, t_1]\,\} \tag{A.2}$$

où G_t est un sous-ensemble des pixels de l'image à l'instant t. G_t peut être représenté comme une image binaire, les pixels à 1 correspondant aux pixels modifiés à l'instant t par rapport à l'instant $(t-1)$. Chaque ensemble G_t correspond donc à l'indicatrice de l'ensemble de définition de la fonction "état de transition", à l'instant t.

Définition 46. Masque de transition *Le masque de transition M_t entre deux images successives d'un document vidéo est une image à niveaux de gris, qui traduit le lieu et la valeur des changements intervenant entre les deux images. Le niveau de gris de chaque pixel de l'image correspond à une valeur de dissemblance entre les instants $(t-1)$ et t. Dans le cas d'une transition parfaite entre deux prises de vue fixes et sans bruit, le lieu des points du masque de transition ayant une valeur de dissemblance non nulle à l'instant t correspond à l'ensemble G_t de la définition du modèle géométrique (cf. définition 45).*

A.3 Vocabulaire lié à la caméra

Définition 47. Mouvement de caméra*[29] La caméra servant à l'acquisition des images peut avoir un mouvement propre, qu'il est intéressant de caractériser comme information supplémentaire nécessaire à l'indexation. On distingue plusieurs types de mouvements différents parmi lesquels :*

- *Mouvement panoramique (panning) : la caméra est en rotation autour d'un axe vertical.*
- *Tilting (tilting) : la caméra est en rotation autour d'un axe horizontal.*
- *Translation (travelling) : la caméra est en translation horizontale, selon un axe parallèle à la scène.*
- *Crane (crane) : la caméra est en translation verticale.*
- *Dolly (dolly) : la caméra se déplace sur un charriot en translation pour se rapprocher ou s'éloigner d'un objet d'intérêt.*
- *Caméra fixe (steadycam) : la caméra ne possède aucun mouvement.*
- *Tenue manuelle (handheld) : la caméra est tenue à la main par un opérateur, produisant une prise de vue sautillante, instable et chaotique.*

Définition 48. Longueur focale de la caméra *Les divers plans peuvent être acquis à des longueurs focales différentes, donnant un aperçu différent de la scène. On distingue ainsi trois types de prises de vue de longueur focale différente et constante : grand champ (*wide*), moyen (*medium*) et plan rapproché (*closeup*). Quant aux prises de vue de type zoom, pour lesquelles la longueur focale varie pour passer d'un de ces trois types à un autre, elles permettent à l'opérateur soit de concentrer l'attention du spectateur sur un détail d'une partie de la scène, ou contraire de révéler une plus grande partie de la scène, cachée auparavant.*

A.4 Vocabulaire divers

Définition 49. Traversée (*intercept) La traversée dans une image binaire correspond à la plus grande distance qu'il est possible de parcourir en ligne droite dans une direction donnée, en partant du bord de l'image, tout en restant dans une même phase.*

Définition 50. Précision *[75] La précision P est définie comme la proportion d'images pertinentes trouvées par rapport au nombre d'images trouvées :*

$$P = \frac{\textit{nombre d'images pertinentes trouvées}}{\textit{nombre d'images trouvées}} \tag{A.3}$$

Définition 51. Rappel *[75] Le rappel R est défini comme la proportion d'images pertinentes trouvées par rapport au nombre d'images pertinentes :*

$$P = \frac{\textit{nombre d'images pertinentes trouvées}}{\textit{nombre d'images pertinentes}} \tag{A.4}$$

Annexe B

Opérateurs de morphologie mathématique

Cette annexe a pour but de présenter de façon succincte les définitions des opérateurs de base de morphologie mathématique, nécessaires à la compréhension des outils morphologiques employés dans ce mémoire. Pour une introduction plus détaillée de cet outil puissant de traitement d'images qu'est la morphologie mathématique, le lecteur pourra se référer à [83, 82].

B.1 Dilatation et érosion

Tout opérateur morphologique est basé sur la comparaison de l'image à traiter, ou du moins de certaines de ses composantes connexes, avec un objet, appelé élément structurant, de taille et de forme choisies par l'utilisateur en fonction du résultat recherché. Un exemple typique d'élément structurant est la boule unité. Les deux opérateurs de base, la *dilatation* et l'*érosion*, sur lesquels la majorité des autres outils morphologiques reposent, ont alors la définition suivante :

Définition 52. Dilatation *La dilatation de l'image X par l'élément structurant B, notée $X \oplus \check{B}$, est l'ensemble des positions x pour lesquelles le translaté de B en x coupe X :*

$$X \oplus \check{B} = \{x, X \cap B_x \neq \emptyset\} \tag{B.1}$$

où $\check{B} = -B = \{-x, x \in B\}$, $\check{B}$ est appelé transposé de B.

Définition 53. Erosion *L'érosion de l'image X par l'élément structurant B, notée $X \ominus \check{B}$, est l'ensemble des positions x pour lesquelles le translaté de B en x est inclus dans X :*

$$X \ominus \check{B} = \{x, B_x \subset X\} \tag{B.2}$$

B.2 Ouverture et fermeture

Par composition des deux opérateurs de dilatation et d'érosion, on construit ensuite l'ouverture et la fermeture, qui appartiennent à la classe des filtres morphologiques, détaillée en B.5.

Définition 54. Ouverture *L'ouvert d'une image X par un élément structurant B, noté X_B, est constitué de l'union des translatés de l'élément structurant B, lorsque celui-ci est totalement inclus dans X :*

$$X_B = \bigcup_x \{B_x, B_x \subset X\} \tag{B.3}$$

L'ouverture correspond à la succession d'une érosion et d'un dilatation :

$$X_B = (X \ominus \check{B}) \oplus B \tag{B.4}$$

Définition 55. Fermeture *Le fermé d'une image X par un élément structurant B, noté X^B, est constitué de l'union des translatés de l'élément structurant B, lorsque ce dernier reste inclus dans le complémentaire de X :*

$$X^B = \bigcup_x \{B_x, B_x \cap X = \emptyset\} \tag{B.5}$$

La fermeture correspond cette fois-ci à une dilatation suivie d'une érosion :

$$X^B = (X \oplus \check{B}) \ominus B \tag{B.6}$$

B.3 Résidus : chapeaux haut de forme

A partir de l'outil morphologique d'ouverture, on définit le *chapeau haut-de-forme blanc* de la façon suivante :

Définition 56. Chapeau haut-de-forme blanc (*white top hat*) *Le chapeau haut-de-forme blanc par l'élément structurant B est l'opérateur morphologique qui consiste à effectuer la différence entre une image originale X et son ouvert X_B. On a ainsi l'équation :*

$$TH(X) \quad = \quad X - X_B \tag{B.7}$$

De façon duale, le *chapeau haut-de-forme noir* a pour définition :

Définition 57. Chapeau haut-de-forme noir (*black top hat*) *Le chapeau haut-de-forme noir par l'élément structurant B est l'opérateur morphologique qui consiste à effectuer la différence entre la fermeture d'une image originale X et elle-même. On a ainsi l'équation :*

$$TH(X) \quad = \quad X^B - X \tag{B.8}$$

où X^B est le fermé de l'image X par l'élément structurant B.

B.4 Géodésie et reconstruction

Afin d'introduire la notion d'opérateurs géodésiques, il est nécessaire de définir ce que l'on entend par distance géodésique.

Définition 58. Distance géodésique *La distance géodésique $d_X(x, y)$ entre deux points x et y dans l'ensemble X est la borne inférieure des longueurs des chemins allant de x à y et totalement inclus dans X. S'il n'existe pas de tel chemin, alors la distance géodésique entre x et y est infinie.*

Muni de cette distance, on définit alors la boule géodésique, comme élément structurant de base.

Définition 59. Boule géodésique *Soit X un ensemble fermé. La boule géodésique $B_X(x,r)$ de centre x et de rayon r a pour définition :*

$$B_X(x,r) = \{y \in X,\ d_X(x,y) \leq r\}$$

Avec cette boule géodésique comme élément structurant, on est à même de définir les opérateurs morphologiques suivants :

Définition 60. Dilatation géodésique *La dilatation géodésique de Y dans X de taille r est définie par :*

$$\begin{aligned} D_X^r(Y) &= \{x,\ B_X(x,r) \cap Y \neq \emptyset\} \\ &= \{x,\ d_X(x,Y) \leq r\} \end{aligned}$$

Définition 61. Erosion géodésique *L'érosion géodésique de Y dans X de taille r est définie par :*

$$E_X^r(Y) = \{x,\ B_X(x,r) \subset Y\}$$

Il est possible de poursuivre la dilatation ou l'érosion géodésiques de l'ensemble Y dans l'ensemble X pour une taille infinie. Dans le cas de la dilatation, on obtient ainsi pour résultat l'ensemble des composantes connexes de X qui contiennent au moins un point de Y. Cette opération prend alors le nom de reconstruction :

Définition 62. Reconstruction *La reconstruction de Y dans X correspond à la dilatation géodésique infinie $D_X^\infty(y)$.*

B.5 Filtres morphologiques

Parmi les nombreux filtres morphologiques existants, les plus classiques, outre l'ouverture et la fermeture, sont sans doute les filtres alternés séquentiels et les filtres par reconstruction.

Définition 63. Filtre alterné séquentiel *On appelle filtre alterné séquentiel de taille n, la transformation obtenue par applications successives sur l'image originale X d'ouvertures et de fermetures de tailles croissantes. Si on note $FAS_n(X)$ le résultat du filtre alterné séquentiel de taille n, on a donc l'expression suivante :*

$$FAS(n) = \phi_n\gamma_n \ldots \phi_2\gamma_2\phi_1\gamma_1 \tag{B.9}$$

où (γ_i) et (ϕ_i) sont respectivement une famille d'ouvertures et une famille de fermetures par un élément structurant de taille i.

La définition du filtre par reconstruction nécessite l'introduction de la notion de reconstruction, présentée dans le paragraphe précédent.

Définition 64. Filtre par reconstruction *On distingue deux types de filtres par reconstruction : par ouverture et par fermeture. Le filtre par reconstruction par ouverture est constitué de la composition d'une érosion de l'image originale X, suivie d'une reconstruction de l'érodé obtenue sous X. Le filtre par reconstruction par fermeture est l'opérateur dual du filtre par ouverture : il se décompose en une étape de dilatation de l'image originale, suivie d'une reconstruction de cette image originale sous son dilaté.*

B.6 Gradient morphologique

Trois gradients morphologiques sont couramment utilisés, avec les définitions suivantes :

Définition 65. Gradient morphologique *Le gradient morphologique se définit comme la différence entre le dilaté et l'érodé d'une image X.*

$$Grad(X) \quad = \quad X \oplus B - X \ominus \check{B} \tag{B.10}$$

où B est l'élément structurant.

Définition 66. Gradient morphologique interne *Le gradient morphologique interne se définit comme la différence entre l'image originale X et son érodé.*

$$Grad\_Int(X) \quad = \quad X - X \ominus \check{B} \tag{B.11}$$

où B est l'élément structurant.

Définition 67. Gradient morphologique externe *Le gradient morphologique se définit comme la différence entre le dilaté d'une image X et elle-même.*

$$Grad\_Ext(X) \quad = \quad X \oplus B - X \tag{B.12}$$

où B est l'élément structurant.

La première formulation donnant des lignes de gradient épaisses, les notions de gradients interne et externe ont été introduites. Ces deux derniers tirent leur nom du positionnement respectif des lignes de gradient, à l'intérieur ou à l'extérieur des objets, dans le cas d'une image binaire.

Annexe C

La transformation HSV améliorée : un filtre fort commutatif

Nous proposons ici une démonstration de la propriété suivante :

Propriété 1. *La transformation HSV améliorée est un filtre fort commutatif.*

Reprenons la définition de la transformation HSV améliorée :

$$\begin{array}{rcl} \text{Transformation HSV Améliorée : HSV} & \longrightarrow & \text{HSV} \\ (h,s,v) & \longrightarrow & (h',s',v') = \left\{ \begin{array}{cc} (h,S_{max},v) & \text{si } sv \geq s_0 \\ (0,0,v) & \text{si } sv < s_0 \end{array} \right. \end{array}$$

avec $S_{max} = v$.

Cette tranformation se décompose en un produit de deux opérations successives γ et ϕ définies de la façon suivante :

$$\begin{array}{rcl} \gamma : \text{HSV} & \longrightarrow & \text{HSV} \\ (h,s,v) & \longrightarrow & (h',s',v') = \left\{ \begin{array}{cc} (h,s,v) & \text{si } sv \geq s_0 \\ (0,0,v) & \text{si } sv < s_0 \end{array} \right. \end{array}$$

et :

$$\begin{array}{rcl} \phi : \text{HSV} & \longrightarrow & \text{HSV} \\ (h,s,v) & \longrightarrow & (h',s',v') = \left\{ \begin{array}{cc} (h,S_{max},v) & \text{si } sv \geq s_0 \\ (h,s,v) & \text{si } sv < s_0 \end{array} \right. \end{array}$$

γ et ϕ étant respectivement une ouverture et une fermeture, et comme, de plus, on a :

$$\text{Transform\_HSV} = \gamma\phi = \phi\gamma$$

il en résulte la propriété recherchée.

Bibliographie

[1] http ://www.rational.com/uml/.

[2] *Annales des télécommunications, numéro spécial "Compression and Image Processing*, volume 52, juillet-août 1997.

[3] Gukrukh Ahanger and Thomas D. C. Little. A survey of technologies for parsing and indexing digital video. *Journal of Visual Communication and Image Representation, special issue on Digital Library*, 7(1) :28–43, 1996.

[4] Philippe Aigrain and Philippe Joly. The automatic real-time analysis of film editing and transition effects and its applications. *Computer and Graphics*, 18(1) :93–103, 1994.

[5] Antonio Albiol, Valery Naranjo, and Jesus Angulo. Low complexity cut detection in the presence of flicker. In *International Conference on Image Processing ICIP'2000*, 2000.

[6] Edoardo Ardizzone, Giuseppe A. M. Gioiello, Marco La Cascia, and Davide Molinell. A real-time neural approach to scene cut detection. In *IS&T/SPIE - Storage & Retrieval for Image and Video Databases IV*, San Josè, CA, january 28 - february 2 1996.

[7] F. Arman, R. Depommier, A. Hsu, and M. Chiu. Content-based browsing of video sequences. In *ACM Multilmedia 94*, pages 97–103, San Francisco, CA, august 1994.

[8] Farshid Arman, Arding Hsu, and Ming-Yee Chiu. Image processing on compressed data for large video databases. *Proceedings of the ACM MultiMedia, California, USA. Association of Computing Machinery*, pages 267–272, june 1993.

[9] J. Astola, P.Haavisto, and Y. Neuvo. Vector median filters. *Proceedings of the IEEE*, 78(4) :678–689, April 1990.

[10] J. Aumont and M. Marie. *L'analyse de films.* Nathan, 1988. 2ème édition.

[11] Serge Benayoun, Hélène Bernard, Pascal Bertolino, Patrick Bouthemy, Marc Gelgon, Roger Mohr, Cordelia Schmid, and Fabien Spindler. Structuration de vidéo pour des interfaces de consultation avancées. *4èmes Journées d'Etudes et d'Echanges Compression et Représentation des Signaux Audio-Visuels, C0RESA'98*, 9/10 june 1998.

[12] S. Beucher. *Segmentation d'images et Morphologie Mathématique.* PhD thesis, Ecole des Mines de Paris, 1990.

[13] S. Beucher and C. H. Demarty. Premier rapport d'avancement, étude kodak. Technical report, Centre de Morphologie Mathématique, Ecole des Mines de Paris, 1997. Confidentiel.

[14] Serge Beucher. Watershed, hierarchical segmentation and waterfall algorithm. In J. Serra and P. Soille, editors, *Mathematical Morphology and its Applications to Image Processing, Proceedings ISMM'94*, pages 69–76. Kluwer Academic Publishers, 1994.

[15] Serge Beucher, Michel Bilodeau, Michel Gauthier, and René Peyrard. Contribution du cmm au rapport final proart/prolab/prometheus. Technical Report N-04/95/MM, Centre de Morphologie Mathématique, mars 1995.

[16] Serge Beucher, Serguei Kozyrev, and Dimitri Gorokhovik. Pré-traitement morphologique d'images de plis postaux. In *Actes du 4ème Colloque National sur l'Ecrit et le Document, CNED'96*, pages 133–140, Nantes, 3-5 juillet 1996.

[17] Serge Beucher and Fernand Meyer. The morphological approach to segmentation : The watershed transformation. In E. R. Dougherty, editor, *Mathematical Morphology in Image Processing*, pages 433–481. Marcel Dekker, Inc, New-York, 1993.

[18] J.S. Boreczky and L.A. Rowe. A comparison of video shot boundary detection techniques. In I.K. Sethi and R.C. Jain, editors, *Storage & Retrieval for Image and Video Databases IV*, volume 2670 of *SPIE Proceedings Series*, pages 170–179, 1995.

[19] P. Bouthemy and F. Ganansia. Video partitioning and camera motion characterization for content-based video indexing. In The Institute of Electrical and Inc. Electronics Engineers, editors, *Proceedings ICIP'96, IEEE International conference on image processing*, volume 1, pages 905–908, Lausanne, Switzerland, september, 16-19 1996.

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

[21] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8 :769–798, november 1986.

[22] Mehmet Celenk. A color clustering technique for image segmentation. *Computer Vision, Graphics and Image Processing*, 52 :145–170, 1990.

[23] Mehmet Celenk. Colour image segmentation by clustering. In *IEEE Proceedings*, volume 138, pages 368–376, september 1991.

[24] D. Chandler. The grammar of television and film. http ://www.aber.ac.uk/ dgc/gramtv.htm, 1994. UWA.

[25] Jacek Cichosz and Fernand Meyer. Morphological multiscale image segmentation. In *WIAMIS'97, Workshop on Image Analysis for Multimedia Interactive Services*, pages 161–166, Louvain-la-Neuve (Belgium), june 1997.

[26] Tim Clark and Rebecca Isaacs. Cut detection in rivl. Technical Report CS631 Fall 95 Project, CS Dept, Cornell University, december 1995.

[27] A. Dailianas, R. B. Allen, and P. England. Comparisons of automatic video segmentation algorithms. In *Proc. SPIE Photonics East'95 : Integration Issues in Large Commercia Media Delivery System*, Philadelphia, october 1995.

[28] C. J. Date. *An Introduction to Database Systems.* The Systems Programming Series. Addison-Wesley Publishing Company, 1975.

[29] G. Davenport, T. G. Aguierre Smith, and N. Pincevert. Cinematic primitives for multimedia. *IEEE Computer Graphics and Applications*, pages 67–74, july 1991.

[30] James Davis. Mosaics of scenes with moving objects. In *IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition (CVPR98)*, pages 354–359, 1998.

[31] M. Davis. Media streams : An iconic visual language for video annotation. In Bergen, editor, *Proc. Symposium on Visual Languages.* Norway, 1993.

[32] Marc Davis. Knowledge representation for video. In American Association of Artificial Intelligence, editor, *Working Notes : Workshop on Indexing and Reuse in Multimedia Systems*, pages 19–28, august 1994.

[33] E. Deardorff, T.D.C Little, J.D. Marshall, D. Venkatesh, and R. Walzer. Video scene decomposition with the motion picture parser. In *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging Science and Technology (Digital Video Compression and Processing on Personal Computer : Algorithms and Technologies.)*, volume 2187, pages 44–55, San Jose, february 1994.

[34] C. H. Demarty and S. Beucher. Color image segmentation using an hls transformation. In *International Symposium on Mathematical Morphology (ISMM'98)*, Amsterdam, The Netherlands, june 1998. Kluwer Academic Publishers.

[35] C.H. Demarty and S. Beucher. Cti ccett : Caractérisation automatique de documents vidéo - application aux journaux télévisés. Technical report, Centre de Morphologie Mathématique, Ecole des Mines de Paris, december 1997. Confidentiel.

[36] C.H. Demarty and S. Beucher. Cti ccett : Caractérisation automatique de documents vidéo - application aux journaux télévisés, deuxième rapport d'avancement. Technical report, Centre de Morphologie Mathématique, Ecole des Mines de Paris, december 1998. Confidentiel.

[37] C.H. Demarty and S. Beucher. Efficient morphological algorithms for video indexing. In *Content-Based and Multimedia Indexing, CBMI'99*, october 1999. To appear.

[38] C.H. Demarty and S. Beucher. Morphological tools for video indexing. In IEEE Computer Society, editor, *International Conference on Multimedia Computing and Systems, ICMCS'99*, volume 2, pages 991–992, Florence, Italy, june 1999.

[39] Rachid Deriche. Fast algorithms for low-level vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1) :78–87, 1990.

[40] S. M. Eisenstein. *Film form.* Harcourt Brace and Co., New York, 1949.

[41] C. Faloutsos et al. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3 :231–262, 1994.

[42] H. J. Zhang et al. Automatic parsing and indexing of news video. *Multimedia Systems*, 2(6) :256–266, 1995.

[43] Olivier Faugeras. *Three-dimensional Computer Vision. A Geometric Viewpoint*, chapter 4. The MIT press, Cambridge, Massachussetts, London, England, 1993.

[44] Jian Feng, Kwok-Tung Lo, and Hassan Mehrpour. Scene change detection algorithm for mpeg video sequence. In The Institute of Electrical and Inc. Electronics Engineers, editors, *Proceedings ICIP'96, IEEE International conference on image processing*, volume 2, pages 821–824, Lausanne, Switzerland, september, 16-19 1996.

[45] Stephan Fischer, Rainer Lienhart, and Wolfgang Effelsberg. Automatic recognition of film genres. In *ACM Multimedia 95*, San Francisco, California, november 1995.

[46] David A. Forsyth and Margaret M. Fleck. Identifying nude pictures. In *3rd IEEE Workshop on applications of computer vision*, Sarasota, Florida, USA, december, 2-4 1996.

[47] David A. Forsyth, Jitendra Malik, Margareth M. Fleck, Hayit Greenspan, Thomas Leung, Serge Belongie, Chad Carson, and Chris Bregler. Finding pictures of objects

in large collections of images. Technical Report CSD-96-905, UCBerkeley, CS Division, 1996.

[48] L. Garrido, P. Salembier, and J.R. Casas. Representing and retrieving regions using binary partition trees. In *International Conference on Image Processing (ICIP'99)*, Kobe (Japan), October 24-28 1999.

[49] Marc Gelgon and Patrick Bouthemy. Determinig s structured spatio-temporal representation of video content for efficient visualization and indexing. In *ECCV'98*, 1998.

[50] Michel Grimaud. *La géodésie numérique en morphologie mathématique. Application à la détection automatique de microcalcifications en mammographie numérique.* PhD thesis, Ecole des Mines de Paris, décembre 1991.

[51] Michel Grimaud. A new measure of contrast : dynamics. In *SPIE, Image algebra and morphological processing III*, volume 1796, pages 292–305, San Diego, july 1992.

[52] Riad Hammoud, Liming Chen, and Dominique Fontaine. An extensible spatial-temporal model for semantic video segmentation. In *1st International Forum on Multimedia and Image Processing (IFMIP'98)*, Anchorage, Alaska, may, 10-14 1998.

[53] Arun Hampapur. *Designing Video Data Management Systems.* PhD thesis, The University of Michigan, 1994.

[54] Arun Hampapur, Ramesh Jain, and Terry E. Weymouth. Digital video indexing in multimedia systems. *Proceedings of the workshop on Indexing and Reuses in Multimedia Systems. American Association of Artificial Intelligence*, august 1994.

[55] Arun Hampapur, Ramesh Jain, and Terry E. Weymouth. Digital video segmentation. In *Proceedings Second Annual ACM MultiMedia Conference and Exposition. Association of Computing Machinery*, pages 357–364, october 1994.

[56] Arun Hampapur, Ramesh Jain, and Terry E. Weymouth. Feature based digital video indexing. In *Proceedings of IFIP 2.6 Third Conference on Visual Database Systems VDB.3*, pages 29–31, Lausanne, Switzerland, march 1995.

[57] Arun Hampapur, Ramesh Jain, and Terry E. Weymouth. Production model based digital video segmentation. *Journal of Multimedia Tools and Applications*, 1(1) :9–46, march 1995.

[58] Frank Heckbert. Color image quantization for frame buffer display. In *SIGGRAPH'82 Proceedings*, page 297, 1982.

[59] Rune Hjelsvold. Video information contents and architecture. In *Proceedings of the 4th International Conference on Extending Database Technology*, Cambridge, UK, March 28-31 1994.

[60] Rune Hjelsvold, Stein Langorgen, Roger Midtstraum, and Olav Sandsta. Integrated video archive tools. *ACM Multimedia*, pages 283–293, 1995.

[61] Rune Hjelsvold and Roger Midtstraum. Modelling and querying video data. *Proceedings of the 20th VLDB conference*, 1994.

[62] Ramesh Jain and Arun Hampapur. Metadata in video databases. *Sigmod Record : Special Issue on Metadata for Digital Media. ACM : SIGMOD*, december 1994.

[63] Henry F. Korth and Abraham Silberschatz. *Database System Concepts.* McGraw Hill Book Company, 1986.

[64] Murat Kunt, Goesta Granlund, and Michael Kocher. *Traitement numérique d'images*, chapter 5, pages 139–170. Presses Polytechniques et Universitaires Romandes, 1993.

[65] Fabrice Lemonnier. *Architecture électronique dédiée aux algorithmes rapides de segmentation basés sur la morphologie mathématique*. PhD thesis, Ecole des Mines de Paris, 1996.

[66] Quang-Tuan Luong. La couleur en vision par ordinateur : 1. une revue. Technical Report 1251, INRIA, Sophia-Antipolis, june 1990.

[67] Beatriz Marcotegui. *Segmentation de séquences d'images en vue du codage*. PhD thesis, Ecole des Mines de Paris, 1996.

[68] D. Marr and E. Hildreth. Theory of edge detection. In *Proceedings of the Royal Society of London*, volume B 207, pages 187–217, 1980.

[69] J. Meng, Y. Juan, and S.-F. Chang. Scene change detection in a mpeg compressed video. In *Proceedings of the IS&T/SPIE Symposium on Electronic Imaging : Science & Technology*, San Josè, CA, February 1995.

[70] Fernand Meyer. Color image segmentation. In *4th Conf. Image Processing and Applications*, volume 354, pages 53–56, 1992.

[71] Fernand Meyer. The levelings. In H.J.A.M. Heijmans and J.B.T.M. Roerdink, editors, *Mathematical Morphology and its Applications to Image and Signal Processing, ISMM'98*, pages 199–206, Amsterdam, june 1998. Kluwer.

[72] Fernand Meyer. Graph based morphological segmentation. In *GBR'99*, pages 1–10, 1999.

[73] Fernand Meyer and Serge Beucher. Morphological segmentation. *J. of Visual Communication and Image Representation*, 1(1) :21–46, 1990.

[74] Akio Nagasaka and Yuzuru Tanaka. Automatic video indexing and full-video search for object appearances. In *2nd Working Conference on Visual Database Systems, IFIP WG 2.6.*, pages 119–133, Budapest, Hungary, october 1991.

[75] Chahab Nastar, Nozha Boujemaa, Matthias Mitschke, and Christophe Meilhac. Surfimage : Un système flexible d'indexation et de recherche d'images. In *4 èmes Journées d'Etudes et d'Echanges "Compression et Représentation des Signaux Audiovisuels, CORESA' 98*, Lannion, FRANCE, june, 9-10 1998.

[76] A. Pentland, R. Picard, and S. Sclaroff. Photobook : Tools for content-based manipulation of image databases. *Int'l Journal of Computer Vision*, 18(3), 1995.

[77] Valéry Risson. *Indexation des conditions d'éclairage d'images couleur*. PhD thesis, Ecole des Mines de Paris, En cours.

[78] R. Ruiloba, P. Joly, S. Marchand-Maillet, and G. Quénot. Towards a standard protocol for the evaluation of video-to-shots segmentation algorithms. In *First European Workshop on Content-Based Multimedia Indexing, CBMI'99*, pages 41–48, Toulouse, France, 1999.

[79] K. Saarinen. Watershed in color image segmentation. In *IEEE Workshop on Non Linear Signal and Image Processing*, pages 14–17, 1995.

[80] P. Salembier and L. Garrido. Binary partition tree as an efficient representation for filtering, segmentation and information retrieval. In *IEEE Int. Conference on Image Processing, ICIP'98*, Chicago (IL), USA, October 4-7 1998.

[81] Raimondo Schettini. A segmentation algorithm for color images. *Pattern Recognition Letters*, 14 :449–506, 1993.

[82] M. Schmitt and J. Mattioli. *Morphologie Mathématique.* Masson, Paris, 1994.

[83] J. Serra. *Image Analysis and Mathematical Morphology.* Academic Press, London, 1982.

[84] J. Serra. *Image Analysis and Mathematical Morphology Volume 2 : Theoritical Advances.* Academic Press, London, 1988.

[85] I. K. Sethi and N. Patel. A statistical approach to scene change detection. In *Proc. of SPIE Storage and Retrieval for Image and Video Databases III*, volume 2420, San Josè, CA, 1995.

[86] Claude Seyrat, Gwénaël Durand, and Pascal Faudemay. Méthodes d'indexation multimédia fondées sur les objets. In *4 èmes Journées d'Etudes et d'Echanges "Compression et Représentation des Signaux Audiovisuels, CORESA' 98*, Lannion, FRANCE, june, 9-10 1998.

[87] L. Shafarenko and M. Petrou. Automatic watershed segmentation of randomly textured color images. In *IEEE Trans. on Image Processing*, volume 6, pages 1530–1543, november 1997.

[88] T.Y. Shih. The reversibility of six geometric color spaces. *Photogrammetric Engineering & Remote Sensing*, 61(10) :1223–1232, October 1995.

[89] Michael A. Smith and Takeo Kanade. Video skimming for quick browsing based on audio and image characterization.

[90] T. G. Aguierre Smith and G. Davenport. The stratification system : A design environment for random access video. In *Proc. 3rd Intl. Workshop on Network and Operating System Support for Digital Audio and Video*, La Jolla, CA, november 1994.

[91] T. G. Aguierre Smith and N. C. Pincever. Parsing movies in context. In *Proc. Summer 1991 Usenix Conf.*, pages 157–168, Nashville, Tennessee, june 1991.

[92] I. Sobel. Neighbourhood coding of binary images for fast contour following and general array binary processing. *Computer Graphics and Image Processing*, 8 :127–135, 1978.

[93] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7 :11–32, 1991.

[94] Deborah Swanberg, Chiao-Fe Shu, and Ramesh Jain. Architecture of a multimedia information system for content-based retrieval. In *Audio Video Workshop*, San Diego, California, november 1992.

[95] Deborah Swanberg, Chiao-Fe Shu, and Ramesh Jain. Knowledge guided parsing in video databases. In *Electronic Imaging : Science and Technology, IST/SPIE*, San Josè, California, 1993.

[96] L. Teodosio and W. Bender. Salient video stills : Content and context preserved. In *Proc. ACM Multimedia 93*, pages 39–46, Anaheim, CA, 1993.

[97] Alain Tremeau and Nathalie Borel. A region growing and merging algorithm to color segmentation. *Pattern Recognition*, 30(7) :1191–1203, 1997.

[98] Corinne Vachier. *Extraction de caractéristiques, segmentation d'image et morphologie mathématique.* PhD thesis, Ecole des Mines de Paris, 1995.

[99] Howard D. Wactlar, Michael G. Christel, Yihong Gong, and Alexander G. Hauptmann. Lessons learned from building a terabyte digital video library. *Computer*, 32(2) :66–73, February 1999.

[100] B. Yeo and B. Liu. Rapid scene analysis on compressed videos. In *IEEE Transactions on Circuits and Systems for Video Technology*, 1995.

[101] B.-L. Yeo. *Efficient Processing of Compressed Images and Video.* PhD thesis, Dept. of Electrical Engineering, Princeton University, january 1996.

[102] Minerva M. Yeung and Bede Liu. Efficient matching and clustering of video shots. In *Proc. 2nd IEEE International Conference on Image Processing, ICIP'95*, pages 338–341, Washington, october 1995.

[103] Ramin Zabih, Justin Miller, and Kevin Mai. A feature-based algorithm for detecting and classifying scene breaks. *Proceedings of the Fourth ACM Conference on Multimedia*, november 1995.

[104] Ramin Zabih, Justin Miller, and Kevin Mai. Feature-based algorithms for detecting and classifying scene breaks. In *Proceedings of the 4th ACM International Conference on Multimedia*, San Francisco, California, USA, November 1995.

[105] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems 1(1)*, pages 10–28, 1993.

[106] H. J. Zhang, C. Y. Low, and S. W. Smoliar. Video parsing and browsing using compressed data. *Multimedia tools applications*, 1 :91–113, 1995.

[107] H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu. Video parsing, retrieval and browsing : An integrated and content-based solution. In *Proceedings of ACM Multimedia.*

Résumé : La multitude de documents multimédia déjà existants ou créés chaque jour nous confronte au problème de la recherche d'informations au sein de bases de données gigantesques, qui rendent toute volonté d'indexation entièrement manuelle impossible. Dans ce contexte, il est devenu nécessaire de concevoir et de construire des outils capables, sinon d'extraire tout le contenu sémantique d'un document donné, du moins d'en élaborer une première structuration de manière automatique.

En se restreignant aux documents vidéo, cette thèse se propose donc de bâtir des outils automatiques réalisant une structuration en deux étapes. Tout d'abord linéaire, elle aboutit à un découpage d'un document vidéo en entités allant de la scène à l'image clé en passant par la prise de vue et le morceau de prise de vue. Puis relationnelle, elle consiste en l'extraction de relations par la mise en évidence de liens, syntaxiques ou sémantiques, de tout ordre entre deux entités de types quelconques. En plus de leur caractère général et automatique, l'ensemble des outils que nous présentons sont en outre conçus dans le respect d'une méthodologie précise. Cette dernière consiste à n'utiliser que des critères simples et de bas niveau de traitements d'images, et tout particulièrement de morphologie mathématique, qui, combinés entre eux et avec des règles logiques de décision, permettent déjà d'atteindre une structuration cohérente, efficace et représentative d'un contenu informationnel de niveau sémantique élevé. Ce choix induit de plus une grande rapidité de nos outils puisque, dans leur ensemble, leur temps d'exécution est inférieur au temps réel. Leur validation est obtenue au travers de nombreux exemples et applications, appartenant essentiellement à la classe des journaux télévisés.

Mots clés : Indexation, Analyse d'images et de séquences, Morphologie mathématique, Segmentation couleur, Segmentation et structuration de vidéo, Structure relationnelle, Contenu sémantique.

Abstract : Due to the large amount of multimedia documents either already existing or produced daily, we are faced with the problem of retrieving information from gigantic databases, for which a purely manual indexing process is currently impossible. In this context, it has become necessary to design new techniques, if not to extract the whole semantic content of a given document, but at least to produce a first structure of it automatically.

Dealing only with video documents, this thesis therefore proposes to build automatic tools that create such a structure in two steps. As a result of the first linear part of the structuring, the video document is splitted into different entities that go from the scene to the object, through the shot, the part of shot and the key frame. The second and relational step consists in extracting relationships, by establishing all kinds of semantic and syntactic links, between the different types of entities. In addition to being general and automatic, the tools proposed were elaborated with respect to a precise methodology. According to this methodology we only use simple and low level image processing criteria, and in particular these from mathematical morphology. These criteria, when combined together and also with logical rules of decision, already enable us to obtain a structure which is consistent, efficient and representative of a content with a high semantic level. The choice of this methodology also leads to a very high speed, as our tools work faster than real time. These tools are validated through numerous examples and applications, based mainly on television news broadcasts documents.

Key words : Indexing, Image and sequence processing, Mathematical morphology, Colour segmentation , Video segmentation and structuring, Shot relation extraction, Semantic content.