Modélisation et gestion de concepts, en particulier temporels, pour l'assistance à la caractérisation de séquences d'images.

Alain Simac

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Sciences et Technologies de l'Information et de la Communication**

Arrêté ministériel : 7 août 2006

Présentée par

« Alain SIMAC »

Thèse dirigée par **« Patrick LAMBERT »**
et par **« Michèle ROMBAUT »**

préparée au sein du **Laboratoire LISTIC Polytech Annecy-Chambéry**
et au sein du **Laboratoire GIPSA-Lab Grenoble**
dans **l'École Doctorale SISEO**

Modélisation et gestion de concepts, en particulier temporels, pour l'assistance à la caractérisation de séquences d'images

Thèse soutenue publiquement le **« 14 juin 2011»**, devant le jury composé de :

Mr. Alain TREMEAU
Professeur à l'Université Jean Monnet, Saint-Etienne, *Président, Examinateur*

Mr. Pierre MORIZET-MAHOUDEAUX
Professeur à l'UTC, Compiègne, *Rapporteur*

Mr. Rémy MULLOT
Professeur à l'Université de La Rochelle, La Rochelle, *Rapporteur*

Mr. Stéphane BRES
Maître de Conférence à l'INSA, Lyon, *Examinateur*

Mr. Patrick LAMBERT
Professeur à l'Université de Savoie, Annecy, *Directeur de thèse*

M. Michèle ROMBAUT
Professeur à l'Université Joseph-Fourier, Grenoble, *Directrice de thèse*

Alain Simac

MODÉLISATION ET GESTION DE CONCEPTS, EN PARTICULIER TEMPORELS, POUR L'ASSISTANCE À LA CARACTÉRISATION DE SÉQUENCES D'IMAGES

Directeurs de thèse :
Michèle Rombaut, Professeur, *Université Joseph-Fourier*
Patrick Lambert, Professeur, *Université de Savoie*

Résumé

Les techniques habituelles d'indexation de vidéos passent généralement par une phase d'apprentissage qui nécessite préalablement la constitution d'une base d'apprentissage. Même si la taille de cette base est souvent réduite, la phase d'annotation réalisée par un expert de l'application est souvent longue et fastidieuse. Dans le cadre de cette thèse, nous avons développé un dispositif qui permet de pré-sélectionner un ensemble de prototypes susceptibles de contenir le concept qui doit apparaître dans la base d'apprentissage. Cette base réduite de prototypes sera ensuite annotée par l'expert.
Nous nous sommes intéressés à des concepts temporels, ce qui nous a amené à étudier particulièrement des caractéristiques liées au mouvement, comme les points d'intérêt spatio-temporels (STIP Spatial Temporal Interest Points). D'autres caractéristiques ont aussi été utilisées concernant la couleur et la présence de formes particulières. Ces caractéristiques sont ensuite exploitées pour structurer la base de vidéos en *briques* spatio-temporelles homogènes. Cette structuration correspond à une sorte de segmentation de la base en fonction de chacune des caractéristiques.
La liaison entre le concept à définir et les briques extraites de la base est en lien avec le *fossé sémantique* bien connu dans la problématique d'indexation automatique. La création de ce lien nécessite l'utilisation de la connaissance de l'expert de l'application sur le concept. Nous avons développé un système dans lequel cette connaissance est extraite par un système de questions/réponses. Les couples de questions/réponses permettent de sélectionner des briques répondant à la contrainte, de définir des relations entre certaines briques, et enfin de naviguer dans l'arborescence des questions.
Des tests ont été réalisés sur des bases de vidéos de provenances diverses telles que des vidéos provenant d'émissions de télévision, de films d'animation, ou encore des vidéos de laboratoire disponibles sur le net, ou réalisées par nos soins. Ces tests montrent les performances satisfaisantes mais aussi les limites de l'approche et ouvrent des perspectives intéressantes, particulièrement sur les aspects collaboratifs et les aspects adaptatifs qui permettraient de capitaliser les connaissances des experts applicatifs et rendraient le système plus efficient.

Alain Simac

MODELING AND MANAGEMENT OF TIME CONCEPTS TO SUPPORT THE CHARACTERIZATION OF IMAGE SEQUENCES

Supervisors :
Michèle Rombaut, Professor, *Université Joseph-Fourier*
Patrick Lambert, Professor, *Université de Savoie*

Abstract

The usual techniques of video indexing generally go through a learning phase that requires the prior establishment of a training database. Even if the size of the database is often reduced, the annotation phase by an expert of the application is often long and tedious. In this thesis, we developed a system that allows pre-selecting a set of prototypes that can contain the concept that must appear in the training set. This reduced base of prototypes will then be annotated by the expert.

We are interested in time concepts, which led us to study particular features related to movement, such as Spatial Temporal Interest Points (STIP). Other features have also been used concerning the color and the presence of particular shapes. These characteristics are then used to structure the video database in homogeneous space-time *blocks*. This structure corresponds to segmentation related to each characteristic.

The link between the concept to define and blocks extracted from the base corresponds to the well known problem of automatic indexing, the *semantic gap*. The definition of this link requires the introduction of the application expert's knowledge. We developed a system in which this knowledge is extracted by a questions/answers system. The couples of questions/answers allow the system to select blocks corresponding to the constraint, to define relationships between some blocks, and finally to navigate on the questions/answers tree.

Tests were performed on video databases from various sources such as videos from television shows, animated films, laboratory videos available on the net, or made by us. These tests show the satisfying performances but also the limitations of the approach and open interesting perspectives, particularly on the collaborative and adaptive aspects that would capitalize in the application expert knowledge and would make the system more efficient.

Je dédie cette thèse
aux personnes qui comptent
le plus dans la vie pour moi :
mes parents, ma femme, mes enfants.

JE tiens à remercier...

... ma direction de thèse, Michèle ROMBAUT et Patrick LAMBERT. Ils m'ont guidé tout au long de ma thèse par leur rigueur scientifique et leurs conseils. Leur encadrement scientifique, leur disponibilité mais aussi leur écoute et leur compréhension ont été très précieux pendant ces longs mois de travail. Au final, ils m'ont transmis une méthodologie de travail, une rigueur dans la rédaction, mais aussi une démarche scientifique. Je leur dois en grande partie la finalisation et la réussite de ces travaux.

... Alain TRÉMEAU du Laboratoire Hubert Curien de l'Université Jean Monnet de Saint-Etienne, de m'avoir fait l'honneur de présider mon jury de thèse.

... Pierre MORIZET-MAHOUDEAUX du laboratoire Heudiasyc de l'université de technologies de Compiègne et Rémy MULLOT, directeur du laboratoire Informatique, Image et Interaction de l'université de La Rochelle, pour avoir accepté d'expertiser mon manuscrit de thèse. Leurs remarques et critiques m'ont permis d'améliorer ce manuscrit et d'envisager différemment le contexte final de mon approche.

... Stéphane BRES du laboratoire LIRIS de l'Institut National des Sciences Appliquées, pour avoir examiné ma thèse.

... la Région RHÔNE-ALPES qui a financé ces travaux trois années durant.

... Sophie MARAT pour avoir travaillé avec moi pendant plusieurs semaines. Cette collaboration a été très enrichissante pour moi.

... Vincent GIRONDEL pour m'avoir aidé plusieurs jours durant en restant derrière la caméra, pour générer une base de données vidéo.

... Jean-Marc SACHE qui a été à de nombreuses reprises une aide précieuse pour disposer de mon Mac au sein du labo mais aussi pour déployer mon environnement sur les différents ordinateurs du labo.

...Florent BALDINI pour m'avoir aidé à construire une base de données vidéo, pour m'avoir épaulé lors de la récupération de mes données après leurs pertes (merci Hitachi) mais aussi pour nos nombreuses discussions.

... Martine GAUVIN pour les nombreuses réponses qu'elle a su m'apporter durant toute la thèse sur toutes les questions administratives qui se sont posées. Sans elle, certaines n'auraient sans doute jamais eu de réponse.

... Georges HABCHI pour ses nombreux conseils avisés dans les moments 'compliqués'. Ils m'auront notamment permis de réussir à finaliser ces travaux.

... Joëlle et Samia qui m'ont apporté une aide précieuse lors des nombreuses démarches administratives.

... les Doctorants du LISTIC et plus particulièrement Renaud, Abdellah, Florent, Yajing, Grégory, Olivier et Amory. Leurs contributions à la réussite de ce travail sont au-delà du caractère scientifique. Votre bonne humeur m'a permis de surmonter les difficultés. J'ai eu la chance de vous avoir à mes côtés ces années durant.

... les Doctorants du conseil de l'école doctorale SISEO, du conseil scientifique et du

conseil pédagogique du CIES. Nous avons partagé de nombreux moments ensemble. Un remerciement tout particulier à Christine pour son aide et son soutien dans la création de l'association des doctorants de l'école doctorale de l'Université de Savoie. En espérant que ce que nous avons bâti servira aux générations futures de doctorants et docteurs. Sans oublier également, Florent, Dorothée, Anne, Fabien.

... toute l'équipe d'enseignement en informatique du département SéréCom avec qui j'ai eu plaisir à travailler et en particulier, Thibault CARRON et Christophe COURTIN pour m'avoir conseillé sur l'enseignement auprès des étudiants et pour m'avoir laissé une grande liberté dans mes enseignements. J'ai beaucoup appris à leurs côtés.

... Jean-Christophe KLEIN pour m'avoir appris à gérer mon temps, à travailler et à manager une équipe et surtout pour m'avoir guidé dans la définition de mon projet professionnel. Ses formations ont été très intéressantes.

... William GOUBERT pour m'avoir accueilli dans son aventure tactile en me permettant ainsi de finir ma thèse dans de bonnes conditions. J'espère que l'avenir nous permettra de faire de grandes choses.

... Serge RIAZANOFF pour m'avoir transmis sa passion pour l'image et pour la planète Terre mais aussi pour m'avoir appris la rigueur au travail lors de mes stages de maitrise et de DEA.

... Jean-Charles MARTY pour m'avoir mis sur la voie de la recherche. Finalement, quelques années plus tard, je me rends compte à quel point il avait raison. Le chemin était particulièrement long et difficile mais tellement enrichissant.

... mes amis, Benoit, Christophe, JeaDea et Kévin pour m'avoir changé les idées à de nombreuses reprises et pour leurs encouragements.

... ma belle-famille, qui m'a beaucoup soutenu dans toutes les épreuves que j'ai dû franchir.

... mes parents, qui m'ont accompagné et ont cru en moi tout au long de ces années. Vous êtes un soutien sans faille et vous l'avez particulièrement montré pendant ces 'quelques' mois.

... ma femme. Je suis conscient de l'immense bonheur que j'ai d'être marié avec elle. Non seulement elle est ma critique la plus utile, mais elle est aussi mon soutien le plus sûr et le plus durable. Merci pour sa patience et son soutien dans les moments difficiles. Car il y a bien eu des moments difficiles mais ce n'est que pour mieux apprécier tous les autres.

... mes enfants, qui m'émerveillent chaque jour. Ils sont le rayon de soleil de chacun de mes jours. Dans leurs yeux, je me sens grandi et tout devient possible.

... tous ceux qui ont partagé un bout de chemin avec moi.

Table des matières

Première partie

Contexte de la thèse

1 Introduction

« En essayant continuellement on finit par réussir. Donc : plus ça rate, plus on a de chance que ça marche. »

Jacques Rouxel

1.1 Motivations

La production de documents vidéo connaît un essor considérable depuis quelques années, tant chez les professionnels de l'image que chez les amateurs : les caméras et les appareils photos numériques permettent une capture aisée de séquences d'images. Grâce à l'augmentation des capacités de stockage (disques durs, cartes à mémoire flash, disques optiques...), les limites de quantités des informations enregistrées sont sans cesse repoussées. Enfin, les connexions haut débit se développant partout dans le monde et la mise en service d'outils de partage de vidéos (YouTube, Dailymotion, Vimeo...) entraîne une croissance permanente de la quantité de données disponibles. Devant un tel contexte d'explosion du contenu numérique, se pose inévitablement le problème de l'organisation de ces données et celui de la recherche de documents.

L'indexation est une pratique indispensable pour retrouver rapidement les documents voulus. La classement par catégorie dans les bibliothèques en est un des exemples les plus simples. Jusqu'à une époque récente, l'indexation semblait réservée à l'intelligence humaine : elle consiste à affecter aux documents des indices, des marques significatives de leur contenu, à la suite d'une série d'opérations mentales complexes. Les dernières recherches en traitement informatique des langues (traduction automatique) et en sémantique (analyse conceptuelle, réseaux sémantiques, analyseur automatique de texte) ont mis à disposition des concepteurs des outils efficaces pour les documents textuels. Mais concernant les documents vidéo, le problème demeure entier. A l'heure actuelle, les bases de documents vidéo sont majoritairement indexées manuellement, quand elles le sont, via l'association de méta-

données, sous la forme de mots clés (étiquettes) et autres informations circonstancielles (prise de vue, type de montage/programmation télévisée). Par exemple, l'Institut National de l'Audiovisuel (INA) emploie 80 personnes qui sont chargées à plein temps d'annoter l'ensemble des contenus télévisés diffusés en France. Ces annotations manuelles, quoique généralement fiables, ne sont pas sans défaut, le principal étant précisément le volume de données, qui rend la tâche particulièrement longue et fastidieuse. Ce premier problème décuple l'impact de tous les autres : la méthodologie d'annotation doit être normalisée afin d'être utilisable par le plus grand nombre, ainsi toute modification sur celle-ci entraîne une nouvelle tâche d'annotation. L'annotation est sujette au problème de langue de l'annotateur, l'annotation ne pourra être disponible que dans un nombre de langues limité. Enfin, il y a la notion de subjectivité qui rend la tâche d'annotation difficilement reproductible car deux annotateurs ne produiront pas systématiquement la même annotation pour un document donné.

C'est ainsi que tout naturellement a émergé la recherche d'une solution informatique pour tenter d'apporter des réponses à ce problème d'annotation.

1.2 Indexation et fossé sémantique

A partir du document vidéo et des informations que l'on peut extraire de façon automatique, il s'agit de savoir si un concept est présent ou non. On peut donc résumer en disant qu'il s'agit d'étiqueter les séquences d'images par des concepts que l'on perçoit dans celles-ci. Le problème est donc le suivant : on peut extraire des éléments qui caractérisent les images et les objets, comme par exemple, la couleur, la texture, la forme générale, la vitesse, la distance de déplacement, mais il ne s'agit que d'informations souvent numériques de contenu sémantique faible, c'est à dire que l'on peut difficilement interpréter. A partir de celles-ci, il apparaît particulièrement difficile d'en déduire des index qui aient un sens et surtout un intérêt. On cherche donc, dans une séquence d'images, à pouvoir donner une définition comme "personne qui court" ou une plus générale comme "déplacement d'un personnage".

La difficulté réside alors dans la création de modèles permettant de faire la liaison entre, d'un côté les informations extraites des séquences d'images qui sont plutôt de bas-niveau et de l'autre côté les concepts signifiants pour l'homme qui sont de haut niveau sémantique. Il s'agit du fossé qui sépare la "sémantique" des concepts que l'on souhaite définir, de l'information issue de l'acquisition de données à partir des pixels des images composant le document vidéo.

La résolution de ce problème peut se réaliser de deux manière différentes : soit on définit des modèles par expertise où un expert détermine les règles liées à l'apparition d'un concept mais cela nécessite que l'opérateur soit expert en traitement d'images et dans l'application ; soit on utilise des modèles définis par apprentissage qui nécessite la création d'une base d'apprentissage annotée proche de l'application. La création de celle-ci est une tâche particulièrement longue et fastidieuse. Dans tous les cas, il est nécessaire d'utiliser l'expertise de l'utilisateur.

1.3 Problématique de la thèse

Pour que l'indexation soit satisfaisante au niveau applicatif, il est nécessaire que les concepts qui sont utilisés soient signifiants pour l'utilisateur, c'est à dire qu'ils correspondent à des informations exploitables. Seul l'opérateur est capable de choisir ce type de concepts. Pour illustrer ce postulat, on peut prendre l'exemple des diagnostics de médecin. On ne peut les produire de manière automatique car le médecin n'est pas en mesure d'exprimer les critères qui lui ont permis de réaliser ce diagnostic. Il se base sur son expérience, sur son ressenti et l'analyse du cerveau humain n'est pas ainsi modélisable. De ce fait, il est impossible des transcrire par un algorithme informatique. L'idée défendue dans cette thèse est que cette participation de l'utilisateur est indispensable, mais qu'il faut essayer de la limiter et faire en sorte que sa tâche soit riche en apport d'information. On fait un mélange des deux approches : on utilise l'expertise de l'utilisateur sur l'application mais en évitant la nécessité d'expertise du traitement d'images par une technique de Questions/Réponses (Q/R). Ceci permet d'extraire des prototypes qui sont des exemples susceptibles de contenir le concept. Puis on demande à l'utilisateur de valider ou non ces prototypes sur une base de données réduite. L'avantage de cette démarche est double : l'extraction d'expertise est simplifiée tout comme la tâche d'annotation.

Dans le cadre de cette thèse, nous nous intéressons tout particulièrement aux documents vidéo. Le contexte applicatif fixé se résume en quelques points :

- l'aspect image : le son et le texte ne sont pas exploités ;
- le plan : élément unitaire classique de découpe des documents vidéo. Il s'agit donc de séquences d'images ne présentant pas de transitions. La détection des transitions est présentée en 3.3.2 ;
- des bases de données très hétérogènes. On ne souhaite pas se limiter à un seul domaine d'application mais produire une méthodologie utilisable sur tout type de documents.

1.4 Structure du mémoire

A la suite de cette introduction, le document est découpé en cinq chapitres et une conclusion qui se présentent comme suit :

Dans le second chapitre, nous débutons par la définition les tâches de caractérisation et d'indexation de la structure et du contenu de séquences d'images. Ensuite, il se poursuit par la présentation du support sur lequel nous allons travailler, c'est à dire, les séquences d'images, puis se termine par un état de l'art sur les différentes méthodes existantes pour annoter une base d'indexation.

Dans le troisième chapitre, nous introduisons les sources d'informations extraites des images, les attributs images. Nous nous attardons particulièrement sur un attribut : les points d'intérêt spatio-temporels sur lesquels nous avons effectué plusieurs analyses pour l'utilisation de cet attribut en analyse du mouvement [Simac-Lejeune *et al.*, 2010a] mais également sur leur pertinence vis-à-vis de la saillance visuelle [Simac-Lejeune *et al.*, 2009].

Le quatrième chapitre aborde le sujet délicat de la modélisation de concepts où nous

présenterons le modèle original qui a été défini : les briques[Simac-Lejeune *et al.*, 2010b].

Dans le cinquième chapitre, nous proposons une nouvelle méthode d'assistance à la définition en utilisant un système de questions/réponses [Simac-Lejeune *et al.*, 2010b] qui a pour but de limiter l'effort de l'annotation à un nombre limité de prototypes.

Le sixième chapitre regroupe l'analyse des performances des différentes étapes du système sur les bases des séquences d'images test ainsi qu'une comparaison avec une méthode existante.

Nous concluons en proposant des pistes de recherche pour faire évoluer le système et les modèles proposés.

2 Positionnement des travaux

« Un problème sans solution est un problème mal posé. »

Albert Einstein

Ces dernières années ont vu l'explosion de la production et de l'utilisation de documents multimédia. Cette situation est due à plusieurs phénomènes récents. Le premier concerne les avancées techniques fantastiques en terme de numérisation, de capacité de mémorisation et de traitement des données numériques. Le deuxième est dû à la diffusion de ces techniques dans le grand public (téléphone mobile, caméra numérique, tablette tactile, etc.). Enfin, le troisième concerne la large diffusion de ces documents (YouTube, Dailymotion, Vimeo, etc.).

L'explosion de la production de ces documents pose le problème de leur gestion. Aussi, si au temps de l'argentique, le rangement des photos familiales pouvait prendre un temps raisonnable, bien qu'important, ce même rangement devient pratiquement impossible du fait de la très grande quantité de photos qui peuvent être facilement prises avec les moyens numériques. Le même problème apparait pour tous les types de documents multimédia, sonores, images ou vidéos.

Le gestion de ces documents à des fins d'exploitation (retrouver un document particulier dans une grande base de documents) ne peut se faire efficacement que si ceux-ci sont convenablement répertoriés. Cela signifie qu'il faut associer à chacun des documents des informations qui permettent de les classer. La technique classiquement utilisée consiste à indexer les documents, c'est à dire de leur associer des index qui aient un sens pour l'utilisateur de la base. Par exemple, pour une base de photos familiales, l'utilisateur pourra définir l'index "famille" et l'associer à toutes les photos sur lesquelles apparait un membre de sa famille. Actuellement, ce type d'indexation à ce niveau d'abstraction se fait manuellement. Les capacités d'indexation automatique sont pour le moment assez limitées : la principale indexation automatique est faite lors de la prise de l'image et permet de mémoriser la date d'acquisition, et éventuellement le lieu (coordonnées GPS sur certains appareils récents par exemple).

C'est dans ce cadre de l'indexation que se situent les travaux de cette thèse. Avant de présenter la problématique de nos travaux, nous rappelons quelques principes concernant les documents multimédia et leur manipulation, ainsi que la construction de modèle de représentation de concepts avant de terminer sur les problématiques de l'indexation, manuelle ou automatique.

2.1 Les documents multimédia

De nos jours, une partie très importante des documents est de nature numérique. Depuis les années 80 où sont apparus les CDRom, la numérisation de tous les documents s'est développée de manière très importante tant au point de vue de la qualité (résolution, définition, échantillonnage) que de la quantité (taille, mémorisation, nombre). D'abord mono-média (seulement de l'image, seulement du texte, seulement du son, etc.), ils sont rapidement devenus des documents associant images, sons, images animées ou vidéos, c'est-à-dire des documents *multimédias*.

2.1.1 Le document vidéo

Dans le cadre de nos travaux, les documents qui nous intéressent tout particulièrement sont les vidéos numériques composées principalement de deux sources d'informations : l'image et le son, qui sont synchronisés. Le flux visuel comporte une séquence d'images fixes qui selon l'axe temporel apparaissent animées à une fréquence allant de 24 à 30 images par seconde. Le flux sonore est composé d'un ou plusieurs canaux et il est typiquement échantillonné entre 16000 et 48000 Hertz. Un troisième flux d'informations généralement associé aux documents multimédias est le texte. Il provient soit d'un flux séparé, soit il est dérivé des sources audio et visuelle. De plus, comme pour tout type de documents, une méta-description peut contribuer à enrichir la connaissance liée au document. Dans un document multimédia, les méta-données peuvent expliquer le contexte de la prise de vue, comme par exemple, la date ou l'auteur. Pour classer des séquences de journaux télévisés, [Amir *et al.*, 2005; Snoek *et al.*, 2006] exploitent ainsi des méta-informations telles que la chaîne, la date et l'heure de diffusion. Plus récemment, [Païs, 2009; Païs *et al.*, 2009] exploite les synopsis des films d'animations en les fusionnant aux caractéristiques image afin de caractériser l'action et l'atmosphère de ces films. Dans nos travaux, nous avons seulement utilisé les vidéos comme *une séquence d'images* sans nous intéresser au son ou au texte.

2.1.2 Les caractéristiques du document vidéo

La taille des documents vidéo numériques non compressés est immense. En qualité professionnelle (NTSC HD, [Benoît, 2002]), elle peut atteindre 1 Go (1 073 Milliards d'octets) par seconde, d'où la nécessité de compression, surtout pour pouvoir l'enregistrer sur des médias communs (CD - 650 Mo, DVD 4,5 Go ou sur un BR-DVD 20/40 Go) ou pour diffuser sur un réseau. Comme d'autres types de documents numériques, les vidéos numériques peuvent

être compressées selon plusieurs algorithmes (MPEG 1,2,4 [1]),plus récemment H.264 [2], et des dérivés propriétaires de ces formats (WMV, QuickTime, RealVideo, etc.). Avec une compression en MPEG 2 équivalente au NTSC HD cité avant, le débit de la vidéo est de l'ordre de 80 Mo par seconde soit un facteur de compression de plus de 12 à 1.
La taille de ces données pose nécessairement le problème du temps de traitement. L'analyse de séquences d'images de manière exhaustive est particulièrement fastidieuse en temps de calcul et il est nécessaire de faire des choix si on souhaite un minimum de performances sur ce critère. Ces choix sont multiples : compression des données, sélection de zones ou de parties de l'image en ignorant le reste, choix des descripteurs permettant la sélection de zones ou de parties de l'image.
Un document numérique vidéo a un certain nombre de caractéristiques physiques :

- le *format* qui indique le mode de représentation des données vidéo dans un fichier ;
- le *type de compression*, souvent liée au format et qui indique l'algorithme de compression utilisé (MPEG1, MPEG2, DivX, MPEG4, H.264, XViD,...) ;
- la *taille* en nombre d'octets ;
- la *cadence* en images par secondes ; le standard NTSC est de 30 images par seconde (États-Unis, cinéma) et le standard PAL est de 24 images par seconde (Europe) ;
- le *débit* en mégabits/seconde ;
- la *définition de l'image* ou *résolution* représente le nombre de pixels affichés pour une image ; la définition classique est de 625 par 400 pixels, celle du DVD est de 720 par 576 pixels, celle de la HD 720p (Haute Définition) est de 1280 par 720 et plus récemment de la Full HD (1080i/1080p) est de 1920 par 1080 pixels ;
- la *durée*, qui représente le temps nécessaire à un humain pour visionner en entier le document vidéo numérique.

2.1.3 Manipulation de documents

Il faut distinguer la phase de création de celle d'exploitation de documents. La phase de création était réservée il y a encore peu de temps à des professionnels, aussi bien dans les domaines du loisir (films, documentaires, jeux télévisuels, ...) que de l'information (journaux télévisés, émissions politiques,...). Depuis les avancées récentes de la technologie (appareils photos, téléphones mobiles et iPhones, caméras numériques et webcam, etc.), la réalisation de films et de vidéos s'est démocratisée. Par exemple sur YouTube [3], 48 heures de vidéo sont chargées par heure et 2000 milliards de vidéos sont vues par jour.
Les bases de vidéos ont atteint de telles tailles qu'il devient extrêmement difficile de les exploiter, même dans un cadre beaucoup plus restreint que le cadre familial. L'exploitation passe généralement par une recherche de documents dans une base de vidéos. On ne peut imaginer que cette recherche ne soit qu'exclusivement visuelle, la visualisation des documents prenant un temps important. Il faut donc que cette recherche passe par une ou des requêtes, souvent textuelles permettant d'atteindre directement les documents correspondant à ces requêtes comme dans le cas des moteurs de recherche vidéo tels que YouTube ou Dailymotion. A noter que ceux-ci utilisent uniquement des mots-clés comme annotation.

1. MPEG2003 - The MPEG Home Page - http ://mpeg.telecomitalialab.com
2. H.264 - Reference documentation - http ://www.itu.int/rec/T-REC-H.264/f
3. source Google YouTube Mai 2011

2.2 L'indexation de séquences d'images

L'indexation contribue à la création d'une représentation virtuelle du document vidéo sous forme de modèles ou de signatures, servant d'intermédiaire entre le document et les besoins d'informations exprimés sous forme de requêtes. Il s'agit donc de générer une information qui sera le point de repère pour accéder ensuite aux séquences d'images et qui permet d'identifier les zones d'intérêt de ces documents. Il existe plusieurs stratégies pour effectuer l'indexation de documents.

- l'indexation manuelle (figure 2.1-gauche) : un opérateur annote manuellement chaque séquence d'images. La sémantique est très bien renseignée mais l'augmentation constante du nombre de documents vidéo rend cette tâche de plus en plus difficile à mettre en place ;
- l'indexation automatique (figure 2.1-droite) : elle consiste à appliquer des algorithmes d'extraction, de modélisation et de classification permettant d'indexer les séquences d'images de manière automatique. L'avantage de cette stratégie est la possibilité d'indexer de grandes collections de documents vidéo. Néanmoins, pour un nombre important d'applications, la qualité de l'indexation n'est pas suffisante pour obtenir des recherches précises et efficaces. La sémantique est peu représentée. Il faut noter que ce type d'indexation demande la plupart du temps une étape manuelle d'indexation d'une base dite d'apprentissage afin de construire les modèles (vérité terrain) ;
- l'indexation 'mixte' ou 'assistée' : un utilisateur intervient plusieurs fois durant le processus d'indexation soit pour annoter soit pour affiner les résultats d'indexation automatique.

En théorie, le niveau d'indexation (signal ou sémantique) n'est pas nécessairement lié à la manière de l'effectuer (manuelle, automatique ou mixte). En pratique, ce qui est produit automatiquement est le plus souvent de bas niveau et ce qui doit être indexé à haut niveau (concept signifiant) ne peut l'être que manuellement (pour des raisons de qualité et de faisabilité). Cependant, dans des applications très spécifiques, l'analyse automatique peut être utilisée pour la détection et la reconnaissance de concepts spécifiques. Dans un cadre générique, c'est actuellement inenvisageable.

2.2.1 Indexation manuelle : l'annotation

L'annotation permet de classer, de résumer, de commenter ou d'enrichir des documents. Dans notre problématique, elle représente l'intermédiaire créé entre le document et les requêtes et peut être vue comme une indexation effectuée manuellement.

De manière générale, cette tâche est considérée comme longue et fastidieuse car elle nécessite l'intervention d'un opérateur ou expert humain, et dépend d'un processus totalement manuel. Toutefois, celle-ci reste très intéressante car elle permet d'obtenir une description du contenu de type sémantique et permet d'analyser le contenu d'un point de vue utilisateur.

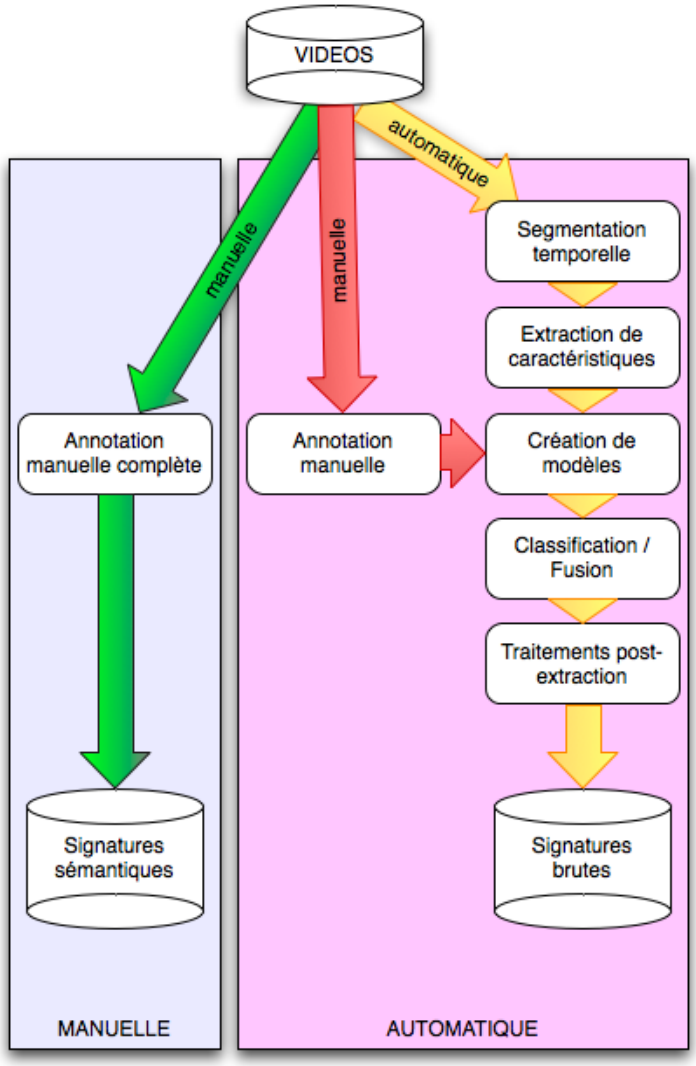


Figure 2.1 — Processus d'annotation. L'annotation peut être complètement manuelle (voie de gauche). Elle peut être complètement automatique (voie de droite) mais nécessite une annotation manuelle pour l'apprentissage.

– **L'annotation libre** qui consiste à laisser l'utilisateur annoter un document avec les descripteurs de son choix pose rapidement le problème de la pertinence et de la subjectivité de l'annotation. Deux opérateurs différents n'annoteraient pas un document de la même façon. Ce type d'annotation est désormais très peu utilisé sauf dans les cas d'un unique opérateur d'annotation et de bases très spécifiques.
On peut citer par exemple **Video-Annex** [Smith et Lugeon, 2000] qui est un outil IBM [4] permettant d'effectuer l'annotation conceptuelle de vidéo sur tout ou partie d'un document (segment vidéo ou image clé d'un plan). L'annotation qui porte sur le document en entier est effectuée via une liste mais peut également être saisie manuellement (voir figure 2.2). Cette annotation peut être collaborative ou bien indépendante.

4. http ://www.research.ibm.com/VideoAnnEx/

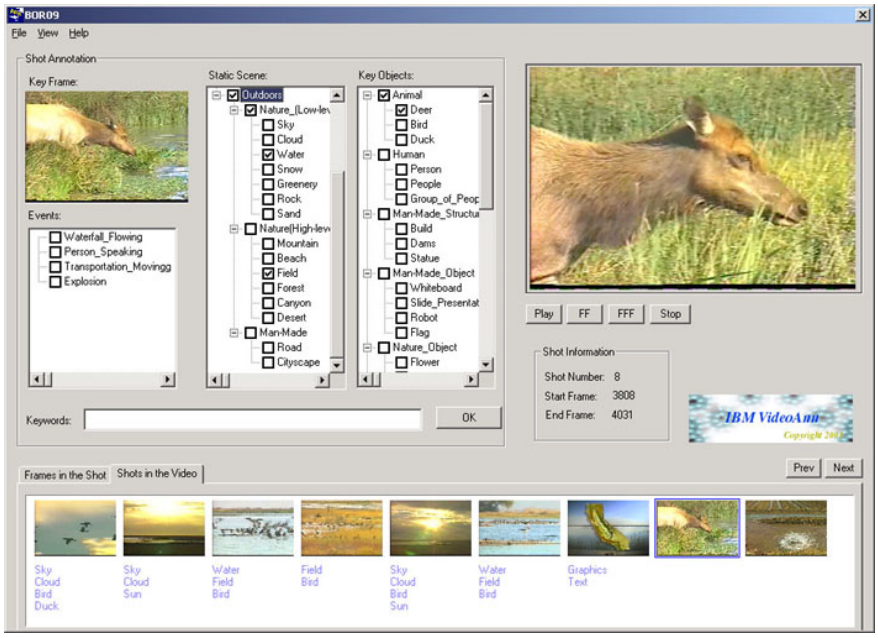


Figure 2.2 — Interface de l'outil Video-Annex (IBM).

L'annotation collaborative a pour objectif de permettre le partage des interprétations visuelles. L'annotation conceptuelle libre est subjective et permet d'élargir le vocabulaire d'annotation.

– **L'annotation conceptuelle** consiste à mettre en place une ou plusieurs ontologie(s) pour faciliter l'interprétation du contenu des séquences d'images. L'opérateur effectue l'annotation en ayant uniquement recours aux concepts de l'ontologie. Celle-ci est généralement représentée de manière graphique, sous forme d'arbre, ce qui permet à l'utilisateur de parcourir rapidement l'arbre et de sélectionner au niveau hiérarchique de son choix, le concept qui lui semble le plus pertinent. Cette annotation est intéressante dans le cas où l'opérateur ne connait pas le concept correspondant mais dispose cependant d'une idée approximative de celui-ci.
Smart VideoText est un système d'annotation vidéo basé sur le formalisme des graphes conceptuels [Sowa, 1984; Chein et Mugnier, 1992] proposé par [Kokkoras *et al.*, 2002]. Dans ce système, les portions vidéo représentent les nœuds du graphe. Ces portions sont identifiées par des références liées à la structure physique du document (identifiant du plan, numéro de l'image dans le plan, etc.) et aussi par les annotations libres. L'idée de base du modèle d'annotation Smart VideoText est de relier les descriptions du contenu de document, décrites par des annotations, au flux vidéo. Chaque annotation sera représentée par un segment logique qui est en général une partie du flux vidéo.

– **L'annotation lexicale** consiste à utiliser un lexique déterminé à l'avance pour permettre d'unifier les descriptions.

Le projet **COALA** [5][Fatemi et Khaled, 2001] (Content Oriented Audiovisuel Library) conduit par l'EPFL en Suisse a permis la réalisation d'une plate-forme d'indexation et d'annotation de journaux télévisées de la TSR (Télévision Suisse Romande). Il s'agit donc d'une application spécialisée dans l'annotation d'un genre particulier de documents vidéo qui est utilisable comme une application web. Video-Annex [Smith et Lugeon, 2000], cité dans l'annotation libre, est également un exemple d'annotation lexicale lorsqu'on utilise l'annotation par liste.

2.2.2 Indexation automatique

De manière générale, un système d'indexation automatique vise à associer à un élément (document ou partie du document) une classe ou une catégorie donnée selon des caractéristiques extraites du document. En général, les algorithmes utilisés s'appuient sur une phase d'apprentissage qui consiste à apprendre un ensemble de relations entre les les caractéristiques et l'index ou le concept. Ces algorithmes ont recours à un ensemble d'exemples afin d'apprendre ces relations. On trouve ainsi deux classes d'approches : l'apprentissage supervisé qui, à partir d'exemples et une phase de classification, peut associer un document à une classe définie préalablement et l'apprentissage non-supervisé où les exemples ne sont pas étiquetés et où la classification consiste à regrouper les éléments en classe distinctes non nommées. On parle alors de clustering.

On commence par extraire des caractéristiques telles que :
- la texture (filtre de Gabor, transformée en ondelette discrète, etc.),
- la couleur (histogramme de couleurs, histogrammes dans l'espace RGB, TSV, etc.),
- segmentation d'objets (contours ou zones homogènes),
- le mouvement (dominant, déplacement local, etc.),
- points particuliers (SIP, STIP, SURF, etc.),
- une combinaison de plusieurs de ces caractéristiques.

Ces caractéristiques sont dites de bas-niveau, car elles sont très proches du signal, et ne véhiculent pas de sémantique particulière sur l'image.

Il s'agit ensuite de définir le modèle qui lie les caractéristiques extraites de bas niveau aux concepts qui correspondent aux différents index qui seront attachés aux documents vidéo, et qui ont un contenu sémantique important. La définition de tels modèles est complexe, liée à l'application aussi bien qu'à l'utilisateur potentiel du système d'indexation. On parle alors de fossé sémantique ('*semantic gap*'). La comparaison entre le modèle du concept et les caractéristiques du document à analyser peut être réalisée par de nombreuses méthodes (mesure de distance, de similarité, de proximité) et permet de décider s'il y a correspondance.

Une technique simpliste consiste à choisir un seul représentant du concept à définir et d'en extraire une signature à partir des caractéristiques extraites, signature qui sert de mo-

5. http ://coala.epfl.ch

dèle du concept. Puis lors de la phase d'indexation, la même signature est évaluée pour tous les documents vidéo, et le concept est validé si la distance entre la signature de référence et la signature calculée est suffisamment petite. Cette approche n'est généralement pas assez performante pour réaliser une indexation robuste de la base d'apprentissage. Il est préférable de définir des modèles plus riches.

2.3 Construction du modèle

Un modèle est une représentation d'un concept permettant d'établir le lien entre ce concept et les informations extraites des documents. Il est donc particulièrement important mais difficile à établir. Plusieurs approches peuvent être exploitées : l'approche manuelle (par expertise), l'approche automatique (par apprentissage) et l'approche mixte (assistée).

2.3.1 Construction par expertise

La construction par expertise consiste à définir un modèle comme respectant un certain nombre de propriétés et de règles. Cette définition est difficile à effectuer pour prendre en compte toutes les variantes d'un concept et pose le problème de la subjectivité (deux experts donnent deux définitions au moins partiellement différentes). De plus, le lien entre la définition et les informations que l'on est capable d'extraire est parfois difficile à effectuer ou reste très spécifique à un domaine particulier. Dans cette catégorie, on peut citer les travaux de Lionel Valet [Valet, 2001] qui transcrit sous forme de symboles et de règles floues la connaissance des experts.

2.3.2 Construction par apprentissage

La construction par apprentissage consiste à utiliser les classes issues de l'apprentissage ainsi que les attributs extraits de chaque document et utilisés pendant l'apprentissage pour déterminer le modèle de la classe. Il s'agit des informations communes et partagées par l'ensemble des documents de la classe considérée.
On peut par exemple citer les travaux de [Ayache, 2007] proposant le modèle des *numcepts* qui est un modèle ayant "pour vocation de clarifier, de généraliser et d'unifier un certain nombre de notions intervenant dans les différents traitements de l'information entre le niveau numérique -num (ou signal) et le niveau conceptuel -cept (ou sémantique)" (source [Ayache, 2007]). Ce modèle vise à décrire les informations en généralisant et en unifiant les deux notions qualitativement différentes. Il définit également des opérateurs permettant de transformer un certain nombre de numcepts en d'autres numcepts plus abstraits : opérateurs d'extraction, de classification, de fusion et de contexte. La construction des numcepts est effectuée par apprentissage supervisé.
On peut également citer les travaux de [Larlus, 2008; Ullah *et al.*, 2010] proposant d'utiliser un modèle sous forme de sac de mots ("bags-of-features" ou "bags-of-words") et d'effecteur une construction par apprentissage.

2.3.3 Construction mixte (assistée)

La construction assistée consiste à partir d'une approche automatique (par apprentissage généralement) à introduire l'expertise dans le système afin de gagner en sémantique.
On peut citer les travaux de Stéphane Ayache [Ayache et Quénot, 2007, 2008a,b] avec le concept de l'annotation collaborative permettant la construction d'un modèle de concepts par annotations successives et collaboratives. Pour cela, l'annotation n'est plus effectuée par un seul 'opérateur' mais par un groupe d'utilisateurs (un grand nombre d'opérateurs au final). L'annotation proposée est simple et ne demande aucune connaissance particulière : pour chaque séquence d'images, on la présente et on l'associe à un concept. On demande à l'utilisateur d'annoter celle-ci par "positif" pour le cas où l'association concept-séquence est correcte, "négatif" pour le cas où l'association est incorrecte, "ignorer" pour le cas où on est incertain de la réponse et qu'on ne souhaite pas trancher entre "positif" et "négatif". La sélection suivante va se servir de l'annotation produite. Ainsi, plus l'utilisateur va faire d'annotations et plus l'ensemble des annotations effectuées va grandir permettant au système de se renforcer et d'être plus performant.
Cette approche est particulièrement intéressante car elle se base sur les connaissances des utilisateurs tout en minimisant leur implication. Cependant, elle est très dépendante des séquences d'images initialement dans la base et demande un certain nombre d'utilisations avant d'obtenir des modèles utilisables. De plus, la phase d'initialisation du système peut être assez longue.
On peut citer les travaux de Goëau[Goëau, 2009] qui s'inspire de l'apprentissage actif pour aider un utilisateur à structurer une collection d'images. Il propose diverses stratégies pour identifier des catégories pertinentes du point de vue de l'utilisateur.

2.3.4 Type de modèles

Il peut également être nécessaire de lier les concepts entre eux, que ce soit pour assister l'indexation ou pour assister la recherche. Par exemple, les concepts 'marcher' et 'courir' sont plutôt proches alors que 'marcher' et 'nager' ne le sont pas. On peut avoir des relations de similarité (marcher et courir) mais aussi des relations d'inclusion (courir est inclus dans le saut en longueur). Toutes ces informations peuvent être incluses dans un modèle de représentation global de l'ensemble des concepts.

2.3.4.1 Graphes conceptuels (GCS)

Les graphes conceptuels sont des modèles permettant de représenter des connaissances sous forme graphique. Ce sont des graphes bipartis étiquetés. Les deux classes (partie) de sommets étant étiquetés respectivement par des noms de "concepts" et des noms de "relations conceptuelles" entre ces concepts. Cette représentation permet de faciliter la compréhension des utilisateurs tout en lui permettant aisément de créer ou modifier des objets. Cette facilité est amplifiée par une séparation explicite de différents types de connaissances. En effet, ceux-ci sont représentés par des objets distincts. Le vocabulaire permettant de représenter des connaissances est structuré dans un objet du modèle appelé "support" qui permet de représenter de façon simple les liens "sorte de" et "est un". Ce modèle permet

d'effectuer des raisonnements qui sont vus comme des opérations de graphes et peuvent ainsi être effectués par des algorithmes de graphes classiques. Enfin, le modèle est muni d'une sémantique en logique du premier ordre permettant d'effectuer des déductions.

Dans sa thèse [Charhad, 2005], l'auteur utilise le formalisme des graphes conceptuels pour créer des modèles de documents vidéo dans le but d'effectuer l'indexation et la recherche par le contenu sémantique.

2.3.4.2 Ontologies

Une ontologie [6] est un ensemble structuré de concepts. Les concepts sont organisés dans un graphe dont les relations peuvent être soient des relations sémantiques soient des relations de composition et d'héritage (au sens objet). Le but est de modéliser un ensemble de connaissances dans un domaine donné. Généralement, celles-ci décrivent :

- individus : les objets de base ;
- classes : ensembles, collections, ou types d'objets ;
- attributs : propriétés, fonctionnalités, caractéristiques ou paramètres que les objets possèdent et/ou partagent ;
- évènements : modifications des attributs ou des relations ;
- relations : liaisons entre les objets.

Il existe quelques applications dans la littérature ayant utilisé les ontologies. On peut citer [Ouyang *et al.*, 2004] qui propose une ontologie pour effectuer l'annotation et le résumé de vidéos sportives. Cette ontologie est représentée en OWL [7]. On peut également citer [Carbonaro, 2008] qui propose une méthodologie pour la recherche de contenu sémantique en utilisant un framework basé sur les ontologies. Celles-ci lui ont permis de mettre en place un système de recherche permettant l'annotation. Dans le domaine de l'image, [Clouard, 2010] propose un modèle d'ontologie pour représenter les objectifs du traitement d'images. Cette approche pourrait être utilisée pour guider l'extraction des caractéristiques menant à l'indexation.

Dans cette thèse, qui constitue des travaux préliminaires, nous nous sommes limités à la recherche de quelques concepts et n'avons pas eu recours à l'utilisation de modèle dit 'élaboré'.

2.4 Méthodes de classification pour l'indexation

L'indexation de documents vidéo peut être vue comme une classification binaire pour chaque concept : celui-ci apparaît ou n'apparaît pas dans le document. Les méthodes classiques de classification ont donc déjà été utilisées pour faire de l'indexation de vidéos. Il s'agit en général de classification supervisée. En effet, la classification non-supervisée,

6. Ressources - http ://boita.info.unicaen.fr/plone/ressources/ontologies/
7. OWL - http ://www.w3.org/TR/owl-guide/

appelée aussi "clustering" consiste à regrouper les éléments qui ont même apparence en termes de caractéristiques. Mais à ces regroupement d'objets similaires, il est souvent difficile d'associer un concept avec un contenu sémantique précis. Ce type de classification est très utilisé lorsque la requête consiste à retrouver des documents similaires à un document exemple. La méthode d'apprentissage non-supervisé K-Means [Diday *et al.*, 1982] partitionne l'ensemble des données K groupes dit 'clusters' ou 'classes'. Cette technique est utilisée par [Zhong *et al.*, 2004] pour effectuer la classification des résultats de détection de mouvements quelconques dans une séquence vidéo quelconque. [Murthy *et al.*, 2010; Murthy et al., 2010] propose de créer des "bag-of-features" extraits de séquences d'images puis d'utiliser K-Means pour classifier l'ensemble des mots visuels obtenus.

Dans le cadre de nos travaux, nous cherchons à associer à un document des concepts signifiants correspondant à des index, les requêtes de recherche se faisant sur ces index. Ces travaux peuvent donc être vus comme une étape préliminaire à l'utilisation de classifieurs supervisés.

La classification supervisée se décompose en deux étapes :
- une première étape permet à partir d'une collection de documents annotés, d'apprendre un modèle par concept,
- une seconde étape permet à l'algorithme de classification d'utiliser ce modèle pour classer de nouveaux documents. Cette étape permet l'indexation des documents.

Les algorithmes de classification supervisée les plus couramment utilisés pour les documents multimédia sont les SVM (Support Vector Machine - Machines à Vecteurs Supports), les Modèles de Markov Caché, et les k plus proches voisins.

2.4.1 Machines à Vecteurs Supports

Les Machines à Vecteurs Supports (SVM) constituent un algorithme de classification discriminatif qui a été introduit en 1995 par Vapnik pour la classification de texte et il est maintenant largement utilisé pour les applications vidéos. [Zhou *et al.*, 2005] propose par exemple d'utiliser SVM pour détecter les vidéos présentant du football.
L'algorithme est basé sur la séparation de deux types de données par un hyperplan séparateur de marge maximum dans un espace de dimension supérieure à l'espace initiale. Le fait de maximiser les marges autour de l'hyperplan séparateur assure de bonnes capacités de généralisation et la représentation des données par un noyau permet de résoudre des problèmes non linéairement séparables (figure 2.3).

L'idée principale des méthodes à noyaux est que la similarité entre les exemples donne plus d'informations sur une classe donnée que les informations caractérisant les données.
On peut noter la proximité conceptuelle des SVM avec les réseaux de neurones, algorithme de classification plus ancien dont le but est aussi la séparation des données par un ou plu-

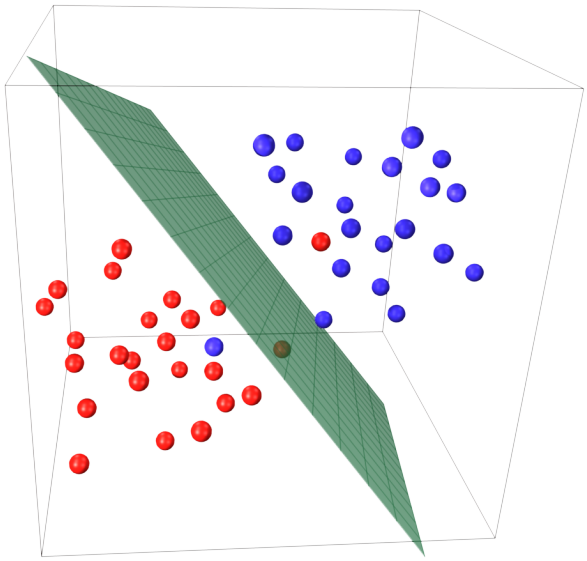


Figure 2.3 — Exemple de séparation en deux groupes de données par un hyperplan.

sieurs hyperplans [8].

2.4.2 Modèle de Markov Caché

On trouve de nombreux algorithmes pour l'apprentissage utilisant des modèles de Markov. Les deux principaux sont l'algorithme de Baum-Welch (dit forward-backward) et l'algorithme de Viterbi. Le premier vise à maximiser les probabilités de génération d'une séquence d'état par ré-estimation itérative. Cet algorithme est relativement coûteux en temps de calcul car il évalue la totalité des séquences possibles. Le second maximise la probabilité de génération des séquences d'apprentissage suivant les chemins les plus probables.
Il est intéressant de noter que [Bae *et al.*, 2005] a proposé une approche de classification hybride utilisant les modèles HMM avec les SVM pour la classification de documents audio-visuels.
On peut par exemple citer les récents travaux d'Emmanuel Ramasso [Ramasso, 2007] qui a défini des modèles pour la reconnaissance de saut d'athlétisme en utilisant des Modèles de Markov Caché et le modèle des croyances transférables. Il utilise cependant plusieurs connaissances expertes : le modèle du personnage (tête par rapport au corps, bras/jambes par rapport au corps, position dans l'espace) mais aussi dans la définition des sauts (phase de course suivie d'une phase ascendante...).

8. http ://www.dtreg.com/svm.htm

2.4.3 Les K plus proches voisins

Le modèle des K plus proches voisins (KNN) est l'une des méthodes de classification les plus naturelles et les plus anciennes. C'est une approche discriminative car elle évalue directement la classe d'un document à partir de ses caractéristiques [Cover et Hart, 1967]. L'algorithme est basé sur la mémorisation des exemples d'apprentissage et l'utilisation d'une fonction de similarité pour comparer deux documents. Pendant la phase de classification, un nouveau document est comparée à l'ensemble des exemples pour évaluer la similarité entre eux selon une fonction de distance. La classe est ensuite décidée par combinaison linéaire ou par vote sur les classes des K plus proches voisins exemples, pondérés par leur similarité avec le nouveau document. On peut utiliser différentes fonctions de distance pour la mesure de similarité en fonction de la représentation des caractéristiques (distance euclidienne par exemple).
Cette approche demande néanmoins beaucoup de ressources notamment pour stocker les exemples et est très coûteuse en temps de classification puisqu'il est nécessaire de calculer pour chaque document sa distance avec tous les exemples (complexité en $\theta(n^2)$) c'est à dire pour rechercher les K plus proches voisins. De plus, les performances se dégradent en présence de données bruitées rendant difficile la généralisation.
Bien que très simple, cette méthode est également l'une des plus efficaces. En moyenne, les KNN sont tout à fait au niveau des autres techniques plus complexes. Le cas particulier K=1, qui se contente d'attribuer à une nouvelle observation la classe de l'exemple le plus proche se révèle être une méthode assez efficace à tel point que [Jain *et al.*, 2000] recommande l'utilisation de 1-NN (KNN avec K=1) comme algorithme de comparaison pour tout nouvel algorithme d'apprentissage. L'approche semble applicable au contexte de la vidéo.

2.4.4 Classification active ou interactive

Les méthodes de recherche interactive d'images les plus rencontrées font souvent l'usage d'un bouclage de pertinence (*relevance feedback*) visant à combler le fossé sémantique à l'aide des annotations de l'utilisateur. [Gosselin, 2005] propose d'utiliser ce principe pour créer un système d'apprentissage interactif dans le cadre de recherche d'images et qui pourrait être étendu aux séquences d'images.

Le but de la recherche est de retrouver les documents appartenant à la catégorie recherchée par l'utilisateur. Le processus démarre par une requête permettant l'initialisation du système. Cette requête permet la sélection des premiers documents en fonction de leur pertinence c'est à dire leur appartenance à la catégorie recherchée. Le résultat peut être annoté par l'utilisateur qui fournit ainsi des précisions à sa requête initiale. A partir de ces précisions, le système peut calculer une nouvelle sélection et présenter les résultats. A chaque nouvelle sélection, l'utilisateur peut fournir de nouvelles annotations et à chaque mise à jour du système, il recalcule la pertinence des documents. La figure 2.4 illustre ce processus. Les travaux de [Goëau, 2009] fonctionne ainsi. Le système commence par effectuer un tri des images à partir d'informations extraites puis c'est l'utilisateur qui effectue des changements de classe pour certains exemples. A chaque changement, le

système recalcule la nouvelle sélection.

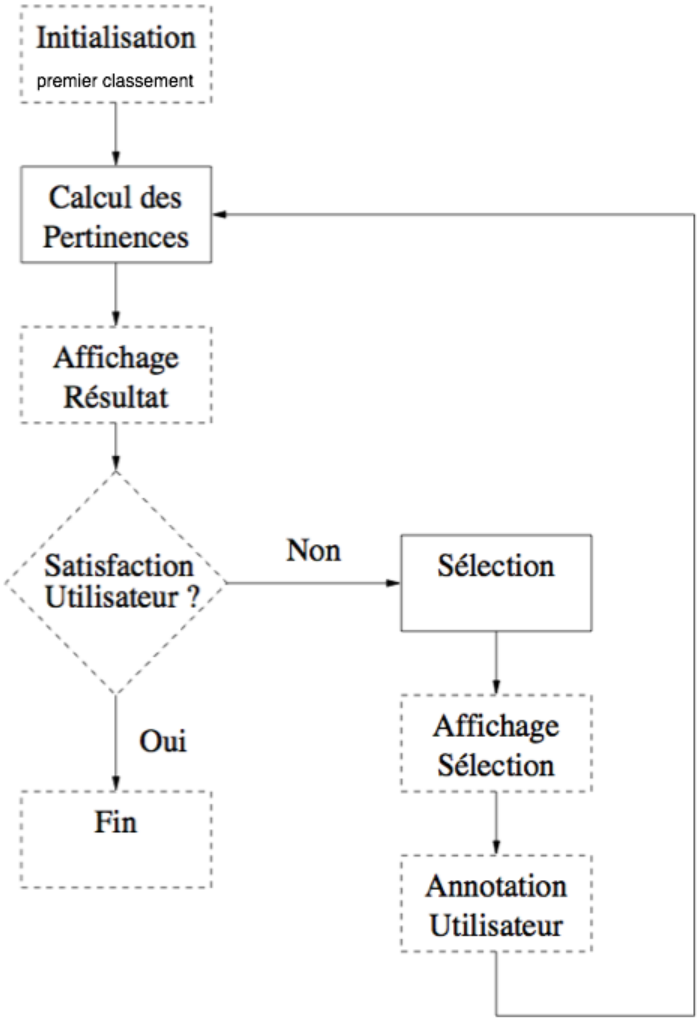


Figure 2.4 — Architecture du bouclage de pertinence (d'après [Gosselin, 2005]).

On note que ce processus est composé de deux étapes particulièrement importantes : la phase de *calcul de pertinence* estimant la pertinence de chaque document comme la probabilité d'appartenir à la catégorie recherche et la phase de *sélection* des images pouvant être annotées.
Au final, cette recherche interactive a montré des performances significativement meilleures qu'avec une approche non active. Le temps de calcul s'en trouve néanmoins allongé bien que des méthodes pour optimiser le processus aient été proposées. L'idée de mettre à profit l'utilisateur pour surmonter la difficulté du fossé sémantique semble permettre d'améliorer les performances des systèmes entièrement automatiques.

2.5 Principaux projets de recherche

Le domaine de l'indexation et de la recherche par le contenu des documents vidéo a donné lieu à plusieurs travaux dans le cadre de projets de recherche dont voici quelques exemples.

2.5.1 Grande base de données, normalisation

On note d'abord les projets dont le but est la gestion de grande base de données.

Projet VITALAS [9] : VITALAS a pour objectif de réaliser un prototype pré-industriel permettant un accès par le contenu aux archives professionnelles multimédias. Dans cette perspective, VITALAS traite des sujets suivants : indexation et recherche multimodale, techniques de recherche dans de grandes bases de descripteurs, visualisation et adaptation contextuelle. Les techniques développées dans VITALAS se basent sur des méthodes avancées de description automatique de contenu et d'apprentissage interactif.

Projet AGIR [10] : est un projet récent et plutôt ambitieux réunissant plusieurs établissement français tels que l'INA et l'INRIA. L'objectif de ce projet est le développement de technologies et d'outils pour la mise en œuvre d'une architecture d'indexation et de recherche par le contenu de données multimédias répondant aux exigences exprimées dans le contexte de la normalisation internationale. Il est composé de plusieurs éléments assurant le traitement des données multimédias : l'extraction de caractéristiques, le langage de description multimédia et les applicatifs.

2.5.2 Apprentissage, évolution, annotation

On trouve ensuite les projets basés sur l'apprentissage et sur l'annotation.

Projet SESAME [11] : SESAME a pour objectif d'étudier les nouvelles solutions à la problématique de l'indexation et de la recherche par le contenu de séquences audiovisuelles. Le but est de mettre au point un système de recherche d'informations multimédia évolutif avec des capacités d'apprentissage par acquisition incrémentale de plusieurs connaissances (stratégiques, épisodiques, etc.).

Projet MUMIS [12] : MUMIS a pour but principal le développement et l'intégration des techniques de bases qui supportent l'indexation conceptuelle automatique de données vidéo et permettant la recherche de contenu. L'idée est d'examiner de manière précise le rôle des annotations résultant d'analyses linguistiques poussées, combinées à des

9. [http ://www.ina-sup.com/recherche/vitalas]
10. Architecture Globale pour l'Indexation et la Recherche [http ://www.ina.fr/recherche /projets/finis/agir.fr.html]
11. Système d'Exploitation de Séquences Audiovisuelles et Multimédias enrichies par l'Expérience [http ://lisisun1.insa-lyon.fr/projets/descrippr21.htm]
12. Multimedia Indexing and Searching Environment [http :// parlevink.cs.utwente.nl/ projects/mumis/index.html]

informations spécifiques au domaine d'application. Le projet est décomposé en deux composantes : l'une 'hors ligne' qui permet la génération automatique d'annotations formelles pour l'indexation conceptuelle et l'une 'en ligne' permettant l'accès en temps réel à la base de données annotée par la première composante.

Projet INFORMEDIA [13] : INFORMEDIA avait pour objectif la mise en oeuvre de nouvelles approches pour l'indexation automatique, la navigation, la visualisation et la recherche vidéo. Le système en résultant était destiné à l'environnement éducatif. Il est basé sur la combinaison de caractéristiques visuelles et audio et analyse le contenu du document pour repérer les *événements signifiants*.

2.5.3 Processus dynamique, détection, reconnaissance, suivi d'objets

Enfin, on trouve les projets cherchant à effectuer en temps réel, de la détection, de la reconnaissance et du suivi d'objets (pouvant être une personne).

Projet AVITRACK [14] : AVITRACK [Francois *et al.*, 2006; Lan *et al.*, 2008] avait pour objectif de développer un framework pour l'indexation et la reconnaissance d'activité dans le cadre la surveillance (notamment d'aéroport).

Projet ADVISOR [15] : ADVISOR avait pour objectif de créer un système temps-réel de détection et de suivi de personnes en utilisant des techniques de détection de mouvements, de calibration de caméra, de modélisation de scènes et de suivi d'objets.

2.6 Conclusions

Dans les faits, un utilisateur souhaitant rechercher des séquences d'images ou des plans utilise l'information sémantique (un concept signifiant, un événement) pour obtenir les réponses les plus pertinentes. Or la plupart des systèmes actuels ne satisfont pas ce besoin car ils privilégient un type spécifique d'informations. Par exemple, ils utilisent des modèles exploitant uniquement leur aspect visuel. Il est difficile de gérer efficacement l'information sémantique par une telle approche. De plus, la plupart des approches proposées dans la littérature font le choix d'un processus totalement automatique incapable de prendre en compte le point de vue d'un utilisateur. Enfin, à partir d'une analyse bas niveau du contenu vidéo, il est difficile d'atteindre un niveau sémantique tel que celui d'un concept signifiant. C'est le problème du fossé sémantique. La figure 2.5 illustre ce problème et montre comment il est envisageable de réduire le fossé sémantique en adoptant une démarche qui fait évoluer le niveau de caractéristiques (augmentation du niveau sémantique) et celui des concepts signifiants (diminution du niveau sémantique).

13. Digital Video Library [http ://www.informedia.cs.cmu.edu/]
14. AVITRACK project [http ://avitrack.net/]
15. ADVISOR project [http ://www-sop.inria.fr/orion/ADVISOR/]

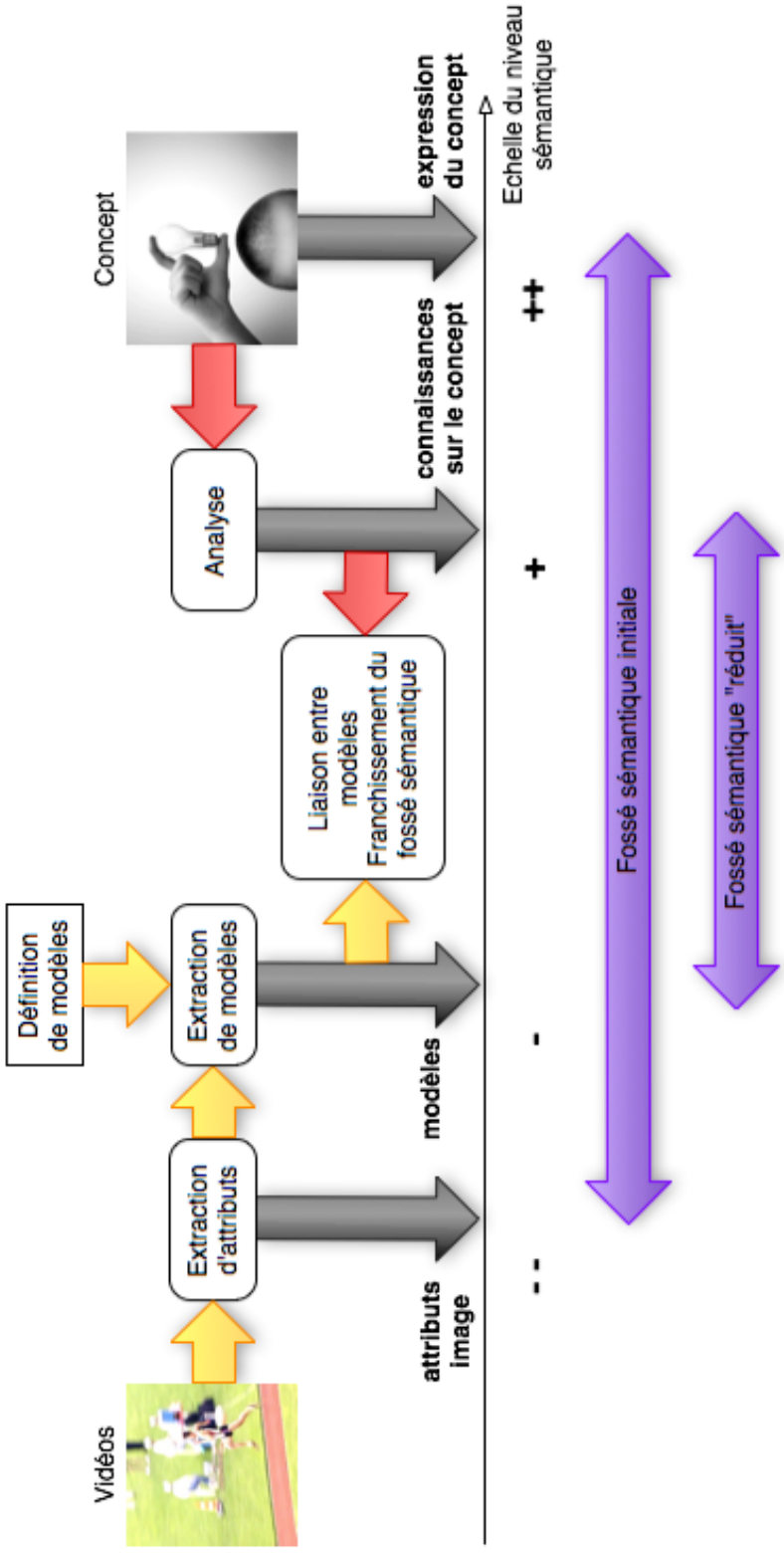


Figure 2.5 — Expression du fossé sémantique existant entre les informations extraites de manière automatique des séquences d'images (à gauche) et l'expression ou la formulation d'un concept recherché par un utilisateur (à droite) et la méthode proposée pour diminuer celui-ci : utilisation de modèles pour augmenter le niveau sémantique des informations extraites (à gauche) et d'un système d'assistance à la formulation d'un concept pour diminuer le niveau sémantique des concepts signifiants (à droite).

L'approche d'indexation proposant les performances les plus intéressantes est l'indexation automatique supervisée mais qui nécessite une base de données annotées de qualité. L'apprentissage automatique supervisé permet de limiter le temps de calcul et obtient des performances intéressantes (bien que dépendantes de la qualité de la base d'apprentissage) mais nécessite l'utilisation d'une base d'apprentissage de documents annotés.
L'apprentissage interactif est une approche avec implication de l'utilisateur. Elle permet d'améliorer les performances mais demande une implication plutôt importante de l'utilisateur. Généralement, l'augmentation de performances va de pair avec l'implication de l'utilisateur.

L'objectif des travaux de cette thèse est de proposer une méthode permettant la construction de la base annotée utilisée pour l'apprentissage automatique supervisé. Pour cela, l'idée est la définition d'un système permettant la création assistée de modèles en se basant sur les connaissances d'un utilisateur (généralement expert de l'application finale) tout en limitant son travail. Ces modèles permettent ensuite l'obtention automatique de l'annotation de la base de séquences d'images.

Année	Auteur	Méthode	Caractéristiques
1999	[Rehatschak et Müller, 1999]	Annotation générique	Découpage automatique, sélection des fragments à annoter et annotation générique
2000	[Correira et Chambel, 2000]	Visionnage actif	Considère la tâche d'annotation comme secondaire à une tâche de visionnage
2002	[Kokkoras *et al.*, 2002]	Annotation assistée	Graphes conceptuels
2008	[Ayache et Quénot, 2008b]	Annotation collaborative	Apprentissage grâce aux annotations successives
2010	[Lin *et al.*]	Annotation assistée (VideoAnnEx)	Annotation MPEG7 en XML
	Notre approche	**Annotation assistée**	**Un système d'aide à l'utilisateur pour définir son modèle**

Tableau 2.1 **—** Les différentes approches existantes pour l'annotation de vidéo et notre approche

Deuxième partie

Modélisation, extraction et gestion d'informations

3 Extraction d'informations des séquences d'images

« Aucune règle n'existe, les exemples ne viennent qu'au secours des règles en peine d'exister. »

André Breton

L'indexation de séquences d'images est une tâche difficile et l'utilisation des informations sur le mouvement permet de remplacer la description globale du contenu par une description locale des objets en mouvement. En cela, cette approche s'inspire du système perceptif de l'être humain dont le regard, dans une séquence d'images, va se porter plus naturellement sur les zones de mouvement [Marat *et al.*, 2009].
Ce chapitre débute par la présentation des attributs que nous avons retenus, parmi les nombreux attributs existants, pour opérer dans notre système d'assistance à l'annotation. Ce choix a été guidé par un des objectifs : focaliser le système sur les concepts orientés mouvement. Les attributs sont donc essentiellement liés au mouvement ou à l'environnement dans lequel il se déroule.
Dans un second temps, nous nous intéresserons plus particulièrement à l'un de ces attributs : les points d'intérêt spatio-temporels. Cet attribut, moins classique, sera notre principale source d'informations sur le mouvement. Nous nous attacherons donc à décrire ses performances pour la détection de transitions, la détection de changement dans le mouvement des objets et la détection d'objets en mouvement. Enfin, pour justifier l'intérêt particulier porté à cet attribut, nous analyserons le lien avec les zones saillantes du système visuel humain.

3.1 Extraction de primitives

La première étape dans l'analyse de contenu consiste à extraire un certain nombre d'informations basiques des images, appelées primitives car elles ne comportent pas d'informations sémantiques. Le but est de diminuer la quantité d'informations à analyser en ne sélectionnant que les informations les plus intéressantes. Cela permet de résumer une image

formée de plusieurs milliers de pixels à quelques primitives. Le problème de la sélection des primitives à utiliser et à conserver laisse apparaître deux difficultés majeures :

- comment ne pas perdre (trop) d'informations ?
  Il s'agit du problème de la sélection de l'information.
- comment ne pas garder (trop) d'informations inutile ?
  Il s'agit du problème de la représentativité de l'information.

Dans le cas des vidéos, les primitives images peuvent être considérées selon deux grandes classes :

- les primitives liées aux informations statiques : elles correspondent à des informations contenues dans une image, comme par exemple un point particulier, une droite, un contour, une couleur ;
- les primitives liées aux informations dynamiques : elles correspondent à des informations contenues dans plusieurs images successives (au moins deux), comme par exemple un vecteur de déplacement.

Dans nos travaux basés sur le mouvement, c'est bien évidemment cette seconde classe qui va nous intéresser tout particulièrement. Cependant la première ne peut pas être complètement ignorée car le mouvement, à lui seul, ne suffit pas toujours à définir de manière précise la caractérisation recherchée. Ainsi, pour une activité sportive comme une course d'athlétisme, l'environnement dans lequel elle se déroule aide beaucoup à son identification. Dans l'objectif d'obtenir des informations sur le mouvement ainsi que sur l'environnement dans lequel celui-ci se déroule. Nous présentons les extracteurs choisis et utilisés dans le dispositif.

3.1.1 Droites caractéristiques

Certaines formes caractéristiques présentes dans une image peuvent faciliter son interprétation et donc son analyse. Les formes les plus recherchées sont les droites, les cercles et les ellipses. Dans notre cas, nous nous intéressons surtout aux droites. Omniprésentes dans les contextes urbains, elles nous apporteront des informations sur l'environnement de la scène. Il existe de nombreuses méthodes pour la recherche de droites ou de segments de droite. Nous avons choisi d'utiliser le très classique détecteur de Hough [Duda et Hart, 1972].

La figure 3.1 présente l'extraction de droites obtenues par transformation de Hough sur un exemple de saut en longueur avec les paramètres suivants : la résolution de la distance d'une unité de pixel $\rho = 1$, la résolution des angles $\theta = \pi/180$ et et le seuil de l'accumulateur $\Gamma = 100$. Il est important de noter que le détecteur de Hough est très sensible au paramétrage.

La caractéristique stockée pour la suite sera le nombre de lignes noté *'ligne Hough'*.

Figure 3.1 — Exemple de droites extraites par transformée de Hough ($\rho = 1, \theta = \pi/180$ et $\Gamma = 100$) sur un exemple de saut en longueur.

3.1.2 Couleurs dominantes

La couleur est une information essentielle dans la compréhension des images. Une caractérisation couleur repose sur deux choix principaux : l'espace de couleur et la représentation. Pour la représentation, les histogrammes de couleurs sont le plus souvent utilisés du fait de leur invariance aux rotations et translations. L'efficacité d'un histogramme de couleurs dépend du choix de l'espace de couleur et de la méthode de quantification. [Wan et Kuo, 1998] ont étudié l'impact de différentes méthodes de quantification sur plusieurs espaces couleurs (RGB, YUV, HSV et CIE LAB). Par ailleurs, seules les couleurs dominantes peuvent être conservées pour former un histogramme, par exemple en classifiant préalablement les couleurs d'un ensemble d'images [Ravishankar *et al.*, 1999]. L'histogramme obtenu globalement sur l'ensemble d'une image ne contient plus l'information sur la répartition spatiale des couleurs. Aussi, certains travaux se sont intéressés à une caractérisation basée sur des régions de l'image. On distingue les approches qui indexent les régions issues d'une étape de segmentation spatiale automatique [Carson *et al.*, 1999; Guérin-Dugué *et al.*, 2001], les approches qui indexent des zones fixes [Moghaddam *et al.*, 2000; Fournier *et al.*, 2001] , les approches qui travaillent par appartenances floues [Mamlouk *et al.*, 2007] et les approches par points d'intérêt [Schmid, 2001] autour desquels le système analyse un bloc.
A partir de la méthode proposée dans [Ravishankar *et al.*, 1999] qui propose de classifier les couleurs et de travailler sur les couleurs dominantes, nous avons établi une méthode rapide d'extraction des couleurs dominantes qui se décompose en trois étapes :

1. passage de l'espace RGB à l'espace Teinte/Luminance/Saturation
2. construction d'un histogramme 2D Teinte/Saturation (dans la soucis de réduire la quantification, nous avons délibérément choisi de ne pas prendre en compte la luminance, moins informative)
3. recherche des modes de l'histogramme construit avec contrôle de la distance minimale entre deux modes (seuil égal à 3 cellules de l'histogramme)

Il est bien sûr nécessaire de réduire le nombre de couleurs. Nous avons fait le choix d'un découpage uniforme de chacune des deux composantes en 32 cellules de même taille ce qui nous permet de représenter 1024 couleurs. Ce choix permet de garder une discrimination entre couleurs suffisante pour déterminer les couleurs et les comparer (rouge, bleu, vert, blanc...) sans avoir à faire la distinction entre couleurs proches (vert clair, vert foncé...). En

pratique, une faible saturation est difficile à traiter car alors les informations de teinte sont très bruitées. Afin d'éviter d'utiliser des informations peu fiables, les 3 premières cellules de la Saturation sont donc ignorées et on utilise finalement 32x29 = 928 couleurs. La figure 3.2 illustre ce principe. Dans le système final, nous ne conservons que les 2 premières couleurs dominantes détectées par recherche de mode dans cet histogramme. Elles sont stockées pour la suite sous le nom de *'couleur dominante1'* et *'couleur dominante2'*.

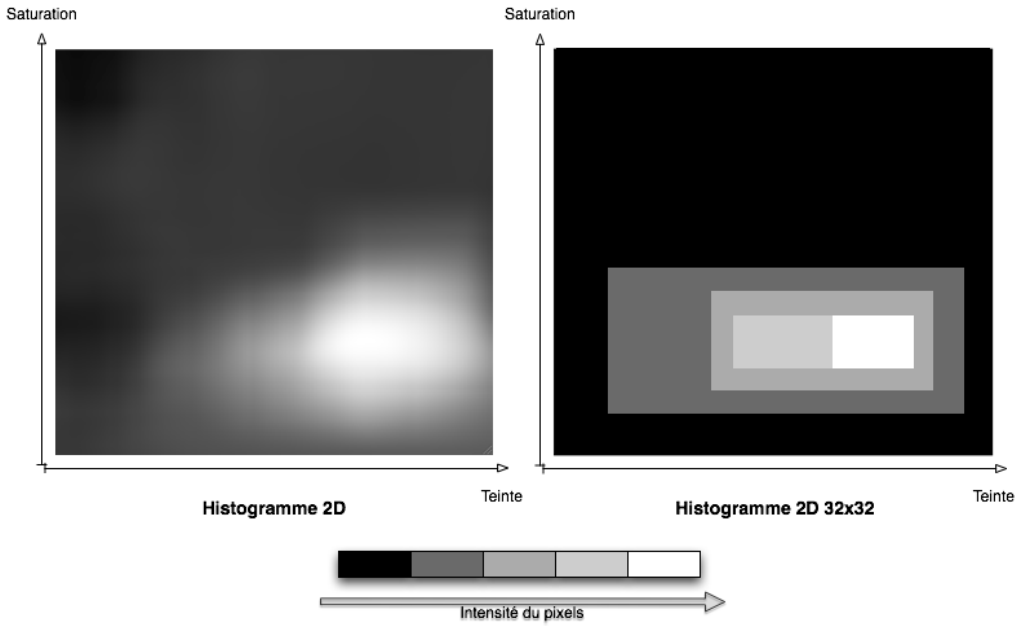


Figure 3.2 — Exemple de carte et de carte filtrée finale (32x32) de densité de couleurs Teinte/Saturation obtenue - Teinte en abscisse et Saturation en ordonnée.

3.1.3 Mouvement de caméra

Dans l'analyse du mouvement des objets, une information intéressante est le mouvement de caméra. En effet, s'il n'y a pas de mouvement de caméra, le mouvement des pixels correspond directement au mouvement des objets en déplacement, mais dans le cas d'une caméra mobile, le mouvement des pixels est l'addition du mouvement de la caméra et d'un éventuel mouvement d'objets. Il est donc nécessaire de compenser le mouvement de caméra afin d'obtenir les informations sur les objets en mouvement.

État de l'art

Une étude bibliographique a permis de distinguer quatre types de méthodes : celles **basées sur la différence d'images consécutives** - on peut notamment citer les travaux de Galmar [Galmar et Huet, 2007] et ceux portant sur le Block Matching dont différentes méthodes sont évaluées dans [Kwan, 1998], celles **basées sur la différence entre l'image courante et une image de référence** - on peut notamment citer les travaux sur la modé-

lisation du fond de Carminati [Carminati et Benois-Pineau, 2005] et de modélisation de contexte de Thonnat [Brémond et Thonnat, 1998], celles **basées sur le flot optique** [Ranchin et Dibos, 2005], celles **bio-inspirées** [Benoit *et al.*, 2010] et enfin les méthodes **par mise en correspondance** [Suvonvorn, 1999]. Les méthodes basées sur la différence entre l'image courante et une image de référence sont les plus utilisées car offrant le meilleur compromis entre efficacité et temps de calcul.

Méthode utilisée

Une des méthodes les plus utilisées est l'estimation du mouvement dominant en faisant l'hypothèse que les objets statiques occupent la majorité de l'image. Dans notre système, nous avons utilisé le logiciel Motion2D qui, en s'appuyant sur le flot optique, implémente la méthode hiérarchique d'estimation du mouvement par régression linéaire multirésolution robuste proposée par Odobez et Bouthemy [Odobez et Bouthemy, 1995] n'exploitant que les gradients spatio-temporels de l'intensité et où le mouvement de chaque région est décrit par un modèle affine 2D complet (2 paramètres de translation et 4 paramètres affines permettant de décrire les mouvements rigides classiquement rencontrés : translation, rotation, changement et toute combinaison linéaire de ces mouvements - équation 3.1).
En utilisant Motion2D, on obtient pour chaque image, une estimation du mouvement dominant :

$$\vec{w_A}(x,y) = \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} \tag{3.1}$$

avec x, y représentant la position du pixel dans l'image, $\vec{w_A}(x,y)$ le flot optique au pixel x, y (avec ses 2 composantes $u(x,y)$ et $v(x,y)$) et a_1, a_2, a_3, a_4, c_1, c_2 correspondant aux paramètres de mouvement déterminés par l'estimation.

Les caractéristiques propres à la caméra sont évaluées et mémorisées. Ce sont le type de caméra, l'orientation et le zoom.

Le type de caméra, fixe ou en mouvement, n'est pas une donnée calculée mais un paramètre indiqué par l'expert en traitement d'images lors du lancement d'un traitement. Il permet notamment de savoir si on doit effectuer la compensation de mouvement ou l'extraction de fond. Nous avons essayé de le faire de manière automatique mais la précision est de l'ordre de 80%. Lorsque le mouvement n'est pas détecté correctement, la qualité des informations est altérée car les différents extracteurs ne fonctionnent pas correctement. Le type de caméra est stocké sous le nom de '*type caméra*'.

Dans le cas d'une caméra en mouvement, l'orientation caméra est calculée à partir de la matrice de transformation fournie par Motion2D, plus précisement à partir des paramètres (a_1,c_1,c_2). Elle est stockée sous le nom de '*orientation caméra*'.

Dans le cas d'une caméra en mouvement, le zoom caméra est calculé à partir de la matrice de transformation fournie par Motion2D. Dans le cas d'une caméra fixe, le zoom caméra

n'est pas calculé. Il est stocké sous le nom de '*zoom caméra*'.

3.1.4 Segmentation et caractérisation d'objets en mouvement

La détection d'objets en mouvement est une tâche importante pour les systèmes d'analyse et de reconnaissance d'activité et pour les systèmes d'interactions monde réel/virtuel. Généralement, pour la majorité des applications, le critère le plus important est la nécessité d'être temps-réel ou de s'en approcher le plus possible. Une des exceptions est justement l'indexation par le contenu qui peut être effectuée sans nécessité de performances temporelles (souvent 'offline').

A l'heure de la démocratisation de la vidéo numérique, des jeux vidéo interactifs et des media numériques, la détection d'objets en mouvement est devenue indispensable dans de nombreuses applications différentes. On peut citer la vidéo surveillance [Brémond et Thonnat, 1998] et la surveillance du trafic routier (Projet AVITRACK - aéroport), l'analyse de contenu pour l'indexation (voir chapitre 2), la robotique, l'assistance médicale, la compression vidéo et l'imagerie satellitaire ou aérienne. Dans les jeux vidéo, de nouvelles applications ont émergé avec la notion d'interface interactive pour réaliser des interfaces avec l'utilisateur dites de nouvelles générations (comme Sony EyeToy) où les gestes (EyeSight [1]) sont détectés puis utilisés pour effectuer les interactions possibles. Ces interfaces permettent même d'effectuer de la capture de mouvement sans marqueur (les systèmes de Motion Capture utilisent traditionnellement des marqueurs sous forme de boules rouges fixées à des positions particulières sur le corps) pour reproduire les mouvements d'un personnage (MotionCapture) à retranscrire dans une production vidéo ou pour effectuer des commandes ou des simulations (Microsoft Kinect développé par PrimeSense [2]).

L'estimation du mouvement dominant obtenue permet d'effectuer la compensation dans la plupart des séquences composant notre base de données tout en limitant au minimum le temps de calcul.
Les figures 3.3 donnent un exemple de la compensation dynamique de mouvement obtenue sur une image de saut en longueur en utilisant le modèle 3.1, présenté précédemment dans l'explication de la méthode.

3.1.4.1 Masque des objets en mouvement

Les objets en mouvement sont obtenus par deux méthodes distinctes en fonction du type de prise de vue. Pour les prises de vue en caméra fixe, c'est à dire dont le fond est statique, nous avons utilisé les méthodes à base d'extraction de fond [McIvor, 2000] qui permettent l'obtention d'un masque des éléments différents du fond c'est à dire des objets en mouvement (exemple en Figure 3.4). Pour les les prises de vue en caméra mobile, nous avons utilisé les méthodes à base de compensation de mouvement [Giai-Checa *et al.*, 1993]

1. http ://www.eyesight-tech.com/
2. http ://www.primesense.com/

Figure 3.3 — Exemple de compensation de mouvement (modèle affine) effectuée en utilisant Motion2D sur un exemple de saut en longueur.

qui permettent la séparation des pixels appartenant au mouvement dominant et ceux n'y appartenant pas : les objets en mouvement (exemple en Figure 3.18).

Caméra fixe :

Dans le cas d'une caméra fixe, la méthode utilisée pour extraire le masque des objets en mouvement se décompose en plusieurs étapes (figure 3.4) qui consistent à utiliser une différence d'images, une détection d'objets basée sur la modélisation des ombres proposées par Martel-Brisson et Zaccarin dans [Martel-Brisson et Zaccarin, 2005] permettant ensuite de les supprimer pour ne pas les prendre en compte dans l'objet, une préparation des données (nettoyage) en utilisant la méthode de Boykov [Boykov *et al.*, 2001] et un filtrage morphologique (une ouverture) permettant d'éliminer les faux positifs plus petits que quelques pixels (taille de l'élément structurant de 3).

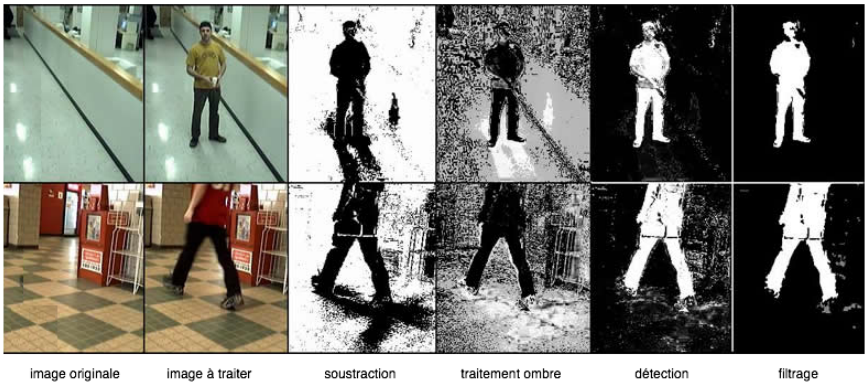


Figure 3.4 — Les différentes étapes permettant l'obtention du masque des objets en mouvement par extraction du fond.

Caméra mobile :

Dans le cas d'une caméra mobile, afin d'obtenir le masque des objets en mouvement, nous avons utilisé le logiciel "Motion2D" qui permet l'estimation du mouvement dominant comme présenté dans 3.1.4. Cela nous a permis de calculer, pour chaque image, les déplacements de la caméra. Enfin, la comparaison entre les vecteurs mouvements et le mouvement dominant permet d'obtenir le masque des objets en mouvement et d'obtenir les vitesses latérales et verticales (courbes 3.6 comme sur l'exemple 3.5).

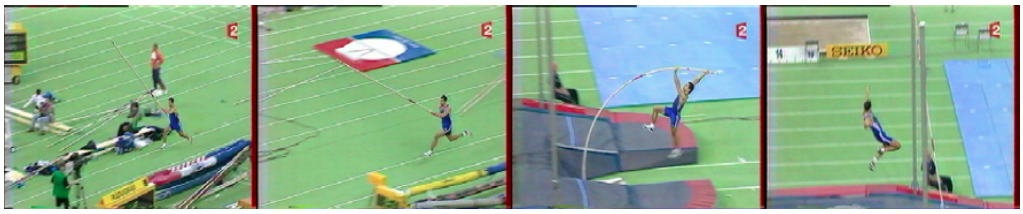


Figure 3.5 — Exemple de saut à la perche.

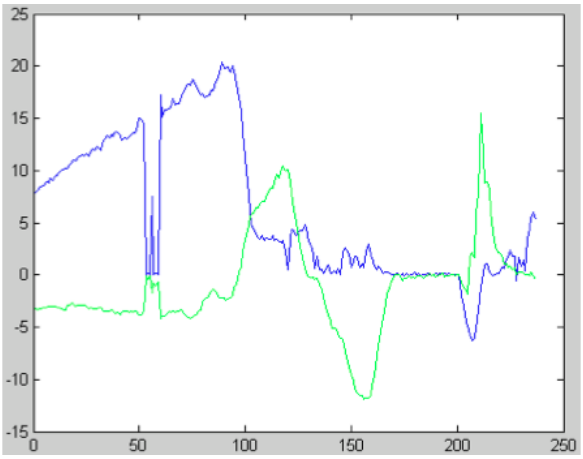


Figure 3.6 — Estimation du déplacement de la caméra en fonction du temps - courbe bleue (sombre) : vitesse latérale - courbe verte (claire) : vitesse verticale.

3.1.4.2 Caractérisation des objets en mouvement

En utilisant la soustraction de fond (figure 3.4) pour les séquences en caméra fixe et la compensation de mouvement par Motion2D (figure 3.18) pour les séquences en caméra mobile, on obtient un masque de l'objet. On considère que c'est une zone homogène et on crée une boite englobante autour de chacun de ces objets (figure 3.7). On détermine le nombre d'objets comme étant le nombre de zones segmentées (en ne comptant pas le fond).

A partir de la boite englobante d'un objet en mouvement, on définit la **position verticale** respectivement **horizontale**, (figure 3.8) comme le milieu des deux bornes horizontales, respectivement verticales, de la boite englobante :

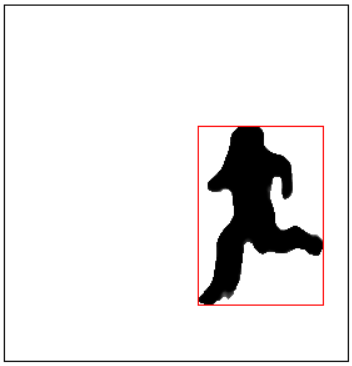


Figure 3.7 — Exemple de boite englobante obtenue sur le masque des objets en mouvement (cas de la caméra mobile).

$$PV(objet) = \frac{1}{2}(\text{max(hauteur(objet))-min(hauteur(objet))})$$

$$PH(objet) = \frac{1}{2}(\text{max(largeur(objet))-min(largeur(objet))})$$

La **compacité** C d'un objet (figure 3.8) est définie comme le rapport minimum entre sa hauteur et sa largeur (valeurs comprises entre 0 et 1) :

$$C = \text{min(largeur/hauteur,hauteur/largeur)}$$

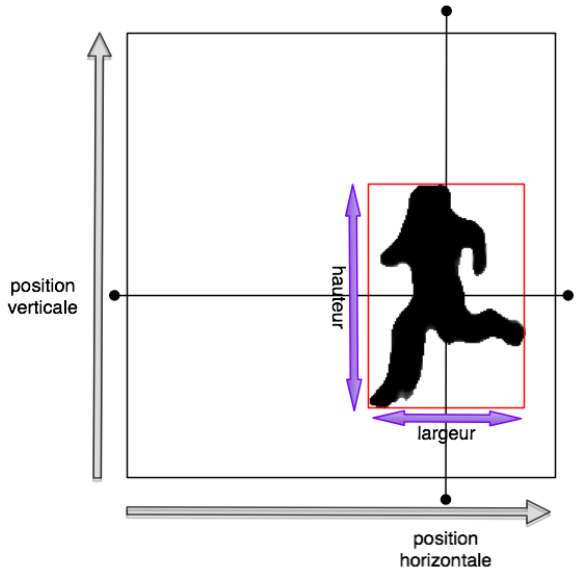


Figure 3.8 — Exemple de position verticale, horizontale et de la compacité d'un objet

A partir de la boite englobante de l'objet en mouvement et la taille de l'image, la **taille relative** est le rapport entre la taille de l'objet (nombre de pixels) et la taille de l'image

(nombre de pixels). Le rapport est une valeur comprise entre 0 et 1.

$$TR(objet) = \frac{\text{taille objet}}{\text{taille image}}$$

3.1.5 Flot optique

Le flot optique (FO dans la suite) représente un champ vectoriel dense où à chaque pixel de l'image est associé un vecteur de déplacement en faisant l'hypothèse que l'intensité (ou la couleur) d'un pixel est conservée au cours du déplacement. De nombreuses techniques de calcul de FO ont été développées à partir des années 80 [Barron *et al.*, 1994]. Généralement, le FO est ensuite représenté par un histogramme ou par le mouvement moyen [Huang *et al.*, 1999] ce qui permet alors d'interpréter les déplacements de pixels.
Dans notre système d'analyse basé sur l'analyse du mouvement, le FO calculé en chaque pixel n'est pas forcément très intéressant car le mouvement est localisé sur certains objets c'est à dire sur de petites zones de l'image. Ainsi, nous avons choisi de ne le calculer que sur un ensemble de points spécifiques, les points d'intérêt (qui seront présentés au paragraphe 3.2) représentant en moyenne moins de 5% des pixels de l'image. D'un point de vue pratique, nous avons utilisé la méthode de calcul présentée dans [Lucas et Kanade, 1981b] où la solution proposée est de faire l'hypothèse que le flot est localement constant sur un voisinage du point considéré et de chercher le flot qui vérifie au mieux les équations de contraintes dans ce voisinage.
Un exemple de résultat est donné dans la figure 3.9 en utilisant les paramètres suivants : $\xi = 0.01$ pour la qualité minimale des fonctions et $\delta = 0.01$ pour la distance euclidienne minimale entre les éléments.

Figure 3.9 — Exemple de flot optique calculé par la méthode de Lucas-Kanade (qualité : 0.01 et distance : 0.01) sur les points d'intérêt - séquence de saut en longueur.

A partir de la liste des vecteurs calculés du flot optique (qui est calculé sur les points d'intérêt spatiaux extraits et non sur une grille régulière), l'intensité est calculée en effectuant la moyenne de l'intensité de l'ensemble des vecteurs.

$$I(FO/image) = \frac{1}{\text{nombre de vecteurs}} * \sum^{\text{vecteurs}} (\text{intensite du vecteur})$$

L'intensité du flot optique par objet est calculée ce la même manière, mais en effectuant la moyenne de l'intensité de l'ensemble des vecteurs dont le point de départ est contenu dans la boite englobante.

$$I(FO/objet) = \frac{1}{n} * \sum_{\text{vecteurs ayant leur origine dans l'objet}} (\text{intensite vecteur}) \quad \text{avec } n = \textit{nombre de vecteurs dans l'objet}$$

A partir de la liste des vecteurs du flot optique, on calcule l'angle formé entre le vecteur et l'axe des abscisses. L'orientation est la moyenne des orientations de chaque vecteur pondérées par son amplitude. L'orientation obtenue est organisée en 8 cadrans : NO, O, SO, S, SE, E, et NE.

$$O(FO) = \frac{1}{\sum_{\text{vecteurs}} (\text{intensite vecteur})} * \sum_{\text{vecteurs}} (\text{angle vecteur}) * (\text{intensite vecteur})$$

Parmi toutes les classes d'extracteurs de primitives qui existent, nous allons maintenant nous intéresser à une classe particulière, celle des points d'intérêt. Dans la suite de nos travaux, les points d'intérêt constituent une information particulièrement intéressante dont nous allons étudier les performances de manière détaillée.

3.2 Les points d'intérêt

Dans une image, les zones porteuses d'informations sont souvent les zones où il y a une forte variation d'intensité ou de couleur. C'est ainsi que beaucoup de travaux commencent par rechercher les contours des objets de la scène analysée. Les contours sont généralement associés aux fortes valeurs de la dérivée première de l'intensité ou de la couleur. Dans une image, le nombre de points de contour est, la plupart du temps, très important. Aussi, toujours dans le soucis de limiter la quantité d'information à analyser, une stratégie consiste à rechercher les **points d'intérêt**. Les points d'intérêt sont par exemple des coins, des jonctions en T, des terminaisons, des points isolés ou des points spécifiques de textures. De nombreuses méthodes ont été proposées dans la littérature pour l'extraction de tels points. Malgré leur diversité, elles peuvent être classées en deux grandes catégories :

- La première extrait les points d'intérêt à partir de l'analyse des variations de la courbure des contours. C'est la classe des "Contour-based method" [Kim et Bovik, 1988] et [Byun et Nagata, 1996] [Mokhtarian et Mackworth, 1986].
- La seconde catégorie de méthodes se propose d'extraire les points directement à partir du signal d'intensité selon une approche différentielle. Les points d'intérêt sont alors assimilés à des pixels présentant de fortes valeurs de la dérivée seconde de l'intensité ou de la couleur. C'est la classe des "Signal-based method" [Harris et Stephens, 1988; Kitchen et Rosenfeld, 1982; Lucas et Kanade, 1981a; Moravec, 1980; Robbins et Owens, 1997; Schmid *et al.*, 2000; Tomasi et Kanade, 1991].

3.2.1 Les points d'intérêt spatiaux

En 1988, Harris [Harris et Stephens, 1988] présente une extension du gradient 2D permettant de mettre en évidence les points d'intérêt spatiaux (noté SIP pour "Spatial Interest Points"). Cette approche correspond à la catégorie "Signal-based method" évoquée précédemment. Les SIP sont définis à partir de la matrice Hessienne H définie par :

$$H(x,y) = \begin{pmatrix} \frac{\partial^2 I(x,y)}{\partial x^2} & \frac{\partial^2 I(x,y)}{\partial x \partial y} \\ \frac{\partial^2 I(x,y)}{\partial x \partial y} & \frac{\partial^2 I(x,y)}{\partial y^2} \end{pmatrix} \tag{3.2}$$

avec $I(x,y)$ représentant l'intensité d'un pixel de coordonnées (x,y) dans l'image I.

Pour atténuer l'importance du bruit généré par les opérations de dérivation, et aussi pour introduire une notion de facteur d'échelle, l'image est généralement spatialement lissée par un filtre gaussien de réponse impulsionnelle :

$$G_s(x,y) = \left(\frac{1}{\sqrt{2\pi}\sigma_s}\right)^2 exp(-\frac{x^2+y^2}{2\sigma_s^2}) \tag{3.3}$$

avec σ_s représentant l'écart-type de la gaussienne, définissant l'importance du lissage.

La matrice Hessienne H devient alors la matrice M définie par :

$$M(x,y) = \begin{pmatrix} \frac{\partial^2 (G_s(x,y) \otimes I(x,y))}{\partial x^2} & \frac{\partial^2 (G_s(x,y) \otimes I(x,y)}{\partial x \partial y} \\ \frac{\partial^2 (G_s(x,y) \otimes I(x,y)}{\partial x \partial y} & \frac{\partial^2 (G_s(x,y) \otimes I(x,y))}{\partial y^2} \end{pmatrix} \tag{3.4}$$

qui peut s'exprimer plus simplement sous la forme :

$$M(x,y) = I(x,y) \otimes \begin{pmatrix} \frac{\partial^2 G_s(x,y)}{\partial x^2} & \frac{\partial^2 G_s(x,y)}{\partial x \partial y} \\ \frac{\partial^2 G_s(x,y)}{\partial x \partial y} & \frac{\partial^2 G_s(x,y)}{\partial y^2} \end{pmatrix} \tag{3.5}$$

La matrice $M(x,y)$, symétrique et positive, caractérise le comportement local de l'intensité du point considéré. Les valeurs propres λ_1 et λ_2 de cette matrice correspondent aux courbures principales associées. Dans le plan (λ_1,λ_2), on distingue trois cas de figure (figure 3.10) :

- Si les valeurs propres λ_1 et λ_2 sont relativement faibles, quelle que soit la direction, le changement d'intensité est négligeable. Le pixel est dans une région homogène. (figure 3.10 - intérieur) ;
- Si la différence entre les deux valeurs propres est importante, seuls des déplacements dans une certaine direction causent un changement significatif de l'intensité. Le pixel est sur un point de contour (figure 3.10 - contour) ;
- Dans les autres configurations, le déplacement dans n'importe quelle direction cause un changement significatif de l'intensité. Ceci indique que le pixel considéré est un point d'intérêt (figure 3.10 - coin).

Un certain nombre de critères ont été proposés pour mettre en évidence ces points d'intérêt. Le critère le plus couramment utilisé est celui proposé dans [Harris et Stephens, 1988].

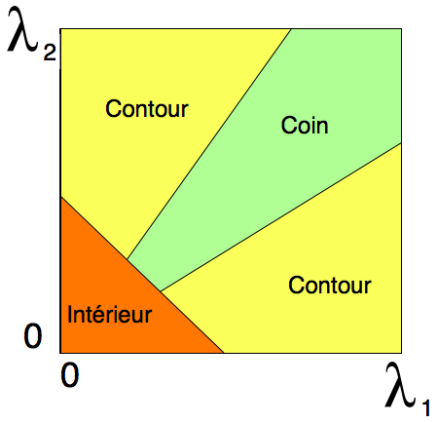


Figure 3.10 — Intérêt du point considéré dans le plan lambda.

Il favorise les configurations où les deux valeurs propres sont grandes et d'égale importance. Noté critère de saillance $R(x,y)$ dans la suite (on trouve également le terme "valeur d'intérêt") est défini par :

$$R(x,y) = det(M(x,y)) - k \times trace(M(x,y))^2 = \lambda_1 \times \lambda_2 - k \times (\lambda_1 + \lambda_2)^2 \qquad (3.6)$$

où k est le paramètre permettant de gérer la sensibilité de détection des points d'intérêt. Les valeurs typiques de k sont généralement choisies dans l'intervalle [0.04 , 0.15]. Selon plusieurs auteurs, l'expérimentation montre qu'une bonne valeur est obtenue pour k de l'ordre de 0.04. Un point d'intérêt est alors un maximum local du critère de saillance. Dans la pratique, on ajoute souvent une contrainte de seuillage sur les maxima locaux de la saillance.

En 1994, Shi et Tomasi proposent dans [Shi et Tomasi, 1994] un autre critère de saillance :

$$R(x,y) = minimum(\lambda_1; \lambda_2) \qquad (3.7)$$

permettant de maximiser l'entropie de l'image. Nos tests préliminaires ont montré que ce critère de saillance ne changeait pas globalement les points détectés en nombre mais position. Ainsi, le critère de saillance de Harris a été conservé dans la suite de l'analyse comme c'est le cas de la très grande majorité des approches de la littérature.

La valeur $R(x,y)$ du critère de saillance pour chaque point de l'image permet de constituer la carte d'intérêt (figure 3.11.b). Les points d'intérêt (figure 3.11.c) correspondent aux pixels pour lesquels la saillance dépasse un certain seuil. Le choix de ce seuil est délicat. L'expérience montre que, pour un même seuil, le nombre de points obtenu est très variable selon l'image analysée. Il peut passer de quelques points à plusieurs centaines de points.

Etievent dans [Etievent, 2004] propose de contrôler automatiquement le seuillage final afin d'obtenir un nombre de points fixé à l'avance. Pour cela, il détecte tout d'abord les points en utilisant un seuil relativement faible en dessous duquel la mesure n'est plus significative afin d'éviter d'obtenir des points instables dans les images contenant peu d'information. Il sélectionne ensuite le nombre voulu de points en choisissant les plus intéressants (en utilisant l'estimateur de la dérivée de Deriche), et la valeur d'intérêt minimale obtenue

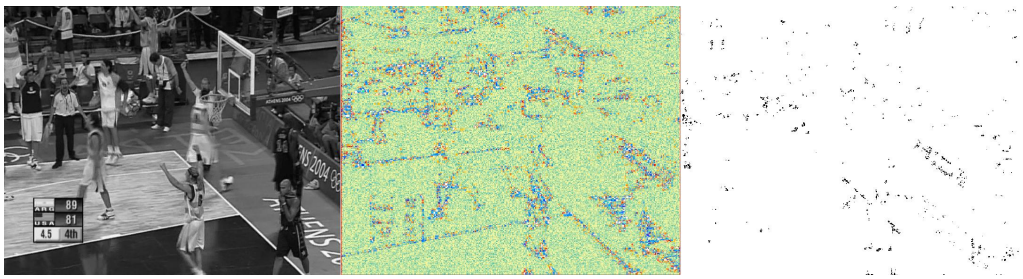


Figure 3.11 — Visualisation sur une image (a) de la carte d'intérêt colorisée (b) et des points d'intérêt obtenus. (c)

correspond alors au seuil définitif. Cette méthode permet un réglage rapide du seuil puisqu'une seule extraction des points est nécessaire. Elle demande cependant la définition de ce que l'on appelle les points "les plus intéressants".
Dans la suite le nombre de points d'intérêt spatiaux sera noté SIP (pour "Spatial Interest Points").

3.2.2 Les points d'intérêt spatio-temporels

Laptev [Laptev et Lindeberg, 2003] propose une extension spatio-temporelle du détecteur de SIP de Harris [Harris et Stephens, 1988] pour détecter les points d'intérêt spatio-temporels, notés STIP ("Space-Time Interest Points") dans la suite. La détection des STIP est réalisée en utilisant une matrice Hessienne H [Laptev, 2005] définie, pour un pixel (x, y) au temps t d'intensité $I(x, y, t)$, par :

$$M(x,y,t) = I(x,y,t) \otimes \begin{pmatrix} \frac{\partial^2 I(x,y,t)}{\partial x^2} & \frac{\partial^2 I(x,y,t)}{\partial x \partial y} & \frac{\partial^2 I(x,y,t)}{\partial x \partial t} \\ \frac{\partial^2 I(x,y,t)}{\partial x \partial y} & \frac{\partial^2 I(x,y,t)}{\partial y^2} & \frac{\partial^2 I(x,y,t)}{\partial y \partial t} \\ \frac{\partial^2 I(x,y,t)}{\partial x \partial t} & \frac{\partial^2 I(x,y,t)}{\partial y \partial t} & \frac{\partial^2 I(x,y,t)}{\partial t^2} \end{pmatrix} \tag{3.8}$$

De manière analogue au détecteur de Harris, un filtre gaussien est appliqué dans le domaine spatial (filtre 2D) et aussi dans le domaine temporel (filtre 1D). Les deux écarts-types, σ_s et σ_t, contrôlent respectivement les échelles spatiale et temporelle. Il est essentiel de séparer le filtrage spatial du filtrage temporel de manière à pouvoir imposer des effets de lissage différents [Dollar *et al.*, 2005]. L'écart type σ_s permet de "contrôler" l'étendue spatiale du détecteur de STIP alors que l'écart type σ_t permet de "contrôler" l'étendue temporelle du détecteur (c'est-à-dire en pratique le nombre d'images de la séquence). Il faut noter que la mise en œuvre de ce filtrage temporel se fait en utilisant un buffer d'images dont la largeur est fonction de σ_t. Dans une perspective de traitement en temps réel, ce buffer introduit un retard entre la dernière image lue et la dernière image traitée (ce retard est de 5 images pour $\sigma_t = 1.5$).

Comme dans le cas spatial, on montre facilement que la matrice Hessienne lissée devient la matrice $M(x, y, t)$ qui s'exprime sous la forme :

$$H(x,y,t) = I(x,y,t) \otimes \begin{pmatrix} \frac{\partial^2 G_s(x,y)}{\partial x^2} & \frac{\partial^2 G_s(x,y)}{\partial x \partial y} & \frac{\partial G_s(x,y)}{\partial x} \otimes \frac{\partial G_t(t)}{\partial t} \\ \frac{\partial^2 G_s(x,y)}{\partial x \partial y} & \frac{\partial^2 G_s(x,y)}{\partial y^2} & \frac{\partial G_s(x,y)}{\partial y} \otimes \frac{\partial G_t(t)}{\partial t} \\ \frac{\partial G_s(x,y)}{\partial x} \otimes \frac{\partial G_t(t)}{\partial t} & \frac{\partial G_s(x,y)}{\partial y} \otimes \frac{\partial G_t(t)}{\partial t} & \frac{\partial^2 G_t(t)}{\partial t^2} \end{pmatrix} \quad (3.9)$$

Pour extraire les STIP, différents critères ont été proposés. Dans [Laptev, 2005; Laptev et Lindeberg, 2003], Laptev a choisi d'utiliser l'extension spatio-temporelle de la fonction de saillance R définie par :

$$R(x,y,t) = \det(M(x,y,t)) - k \times trace(M(x,y,t))^3 \quad (3.10)$$

où le paramètre k est ajusté de manière empirique à 0.04 comme pour la détection des SIP. Dans la suite, la détection des STIP sera faite avec critère associé à un seuil choisi empiriquement. La grande variété de vidéos envisagée ne permettait pas d'utiliser le seuillage adaptatif proposé par Etievent dans [Etievent, 2004]. Par contre, nous avons constaté qu'un même seuil, réglé empiriquement, pouvait convenir pour une même catégorie de vidéos. Dans la suite le nombre de STIP sera noté STIP.

3.3 Analyses expérimentales des points d'intérêt spatio-temporels

3.3.1 Sensibilité de la détection des points d'intérêt spatio-temporels

Les STIP possèdent des propriétés qui sont bien connues notamment leur stabilité relative par rapport aux transformations géométriques [Comer et Draper, 2009] et aux changements d'illuminations [Faille, 2004]. Nous avons étudié d'autres propriétés qui sont aussi importantes dans le cadre de l'analyse de vidéos, telles que la robustesse des STIP au bruit impulsionnel, aux modifications de contraste, à l'orientation et au mouvement de caméra.

3.3.1.1 Sensibilité à l'orientation et au mouvement de caméra

Une première expérience a été effectuée en utilisant les séquences issues de la base de vidéos sportives, en particulier les vidéos de sauts d'athlétisme. Ces vidéos ont la particularité d'avoir des angles de prises de vue qui peuvent être différents et qu'on peut classer en 2 groupes principaux : les prises de vue de face et celles de profil. Parmi les séquences disponibles, nous avons également des séquences prises en caméra fixe et et en caméra mobile. Dans ce test, nous cherchons à vérifier la capacité des STIP à détecter un objet dans les différentes conditions de prise de vue. Le critère de performance retenu est la précision dans la détection de l'athlète en mouvement. Cette précision est mesurée comme le pourcentage de STIP se trouvant positionnés sur l'athlète. Le contrôle a été effectué manuellement. Le nombre moyen de STIP est également indiqué. Le tableau 3.1 présente les taux de détection obtenus pour différentes prises de vue.

Les résultats présentés dans le tableau 3.1 montrent deux choses. D'abord, globalement les STIP se trouvent bien localisés sur l'élément en mouvement de la scène. Ensuite, les conditions de prise de vue ont une influence sur le nombre de STIP détectés.

	nb moyen de STIP/image	Précision
Caméra de face	76	91%
Caméra de profil	91	99%
Caméra fixe	82	93%
Caméra mobile	83	98%

Test - 60 séquences de sauts d'athlétisme
$k = 0,04$, $\sigma_s = \sigma_t = 1.5$ et $Seuil\ saillance = 150$

Tableau 3.1 — Influence de la prise de vue sur la détection d'un objet en mouvement.

Comme on pouvait s'y attendre, la meilleure orientation est la prise latérale, dans laquelle le mouvement de l'athlète est plus visible. De manière plus surprenante, les tests effectués sur les données en caméra fixe obtiennent des résultats légèrement moins bons que pour celles en caméra mobile. Ce résultat peut être expliqué par le bruit lié à la compression. Dans le cas statique, ce bruit peut engendrer dans certaines zones de l'image de fausses détections. Dans le cas d'une caméra en mouvement, le bruit de compression a une plus forte probabilité de se produire sur des zones différentes de l'image et sera donc filtré par le lissage temporel.

3.3.1.2 Influence du contraste et du bruit

Une analyse de l'influence des modifications de qualité d'image sur la détection STIP a également été réalisée. Avec les mêmes séquences que celles utilisées dans les paragraphe précédent, en gardant les mêmes critères, deux situations ont été examinées : des modifications de contraste et l'ajout de bruit de type impulsionnel, celui-ci perturbant le plus la détection des STIP. La figure 3.12 donne des illustrations de ces perturbations. Les tableaux indiquent le nombre moyen de STIP par image et la précision pour différentes valeurs de contraste 3.12.a et pour différents niveaux de bruit (3.12.b/c).

Contraste	**nb moyen de STIP/image**	**Précision**
25%	0	/
50%	< 1	100%
75%	11	100%
100% reference	68	96,2%
125%	155	74,8%
150%	203	65,0%
175%	231	64,0%

Test - 60 séquences d'athlétisme
$k = 0,04$, $\sigma_s = \sigma_t = 1.5$ et $S = 150$

Tableau 3.2 — Influence du contraste sur la qualité de détection des STIP.

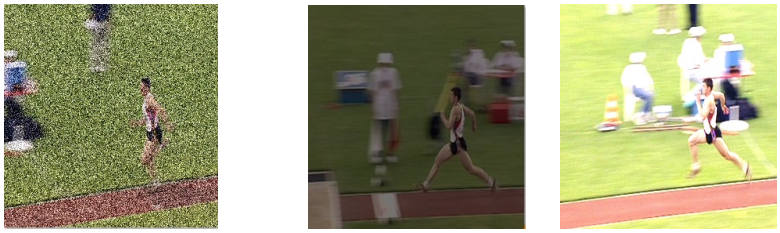
En prenant comme référence l'image non perturbée, on remarque sur le tableau 3.2 que le nombre de STIP détectés est très sensible à la modification du contraste. En revanche,

Bruit		nb moyen de STIP/image	Précision
Puissance	Intensité		
0	0	68	96,2%
20	20	69	95,6%
20	50	70	94,3%
20	70	72	93%
20	100	74	93,2%
50	20	83	86,7%
50	50	119	63%
50	70	153	58,1%
50	100	224	42,9%

Test - 60 séquences d'athlétisme
$k = 0,04$, $\sigma_s = \sigma_t = 1.5$ et $S = 150$

Tableau 3.3 — Influence du bruit sur la qualité de détection des STIP.

comme on le voit sur le tableau 3.3, ce nombre est peu affecté par l'ajout d'un bruit impulsionnel (Figure 3.12) lorsque celui-ci n'est pas trop important. Quant à la précision, sans



a) Bruit additionnel puissance=50 intensité=50 — b) Contraste faible 25% — c) Contraste fort 175%

Figure 3.12 — Exemple de vidéos.

surprise, elle diminue au fur à mesure de l'élévation du contraste ou de la quantité de bruit. On remarque néanmoins que, à faible puissance, le bruit n'a pas un impact important sur la précision de détection des points d'intérêt.

3.3.1.3 Effet de la compression

La dernière influence analysée est le facteur de compression de la vidéo. En effet, la compression peut créer des effets d'aliasing, de crénelage des lignes, qui peuvent être perçus comme des angles [Clarke, 1995]. Le tableau 3.4 présente l'influence du facteur de compression MPEG2 sur le nombre de STIP générés et sur la précision de détection. Ces tests ont été effectués sur des séquences de synthèse (formes géométriques simples -carré/cercle/triangle- se déplaçant en translation) et les séquences de sauts d'athlétisme déjà utilisées. Dans ces tests, il est intéressant de noter que, dans les séquences de synthèse, le carré ne génère pas de faux positif. On peut ainsi estimer que l'aliasing aura peu d'effet perturbateur sur le nombre de

STIP détectés. Ces résultats montrent que le facteur de compression a un effet important dès que l'on dépasse le seuil de 30% de compression. Afin de ne pas perturber les résultats, il sera donc nécessaire de s'assurer que les séquences utilisées ne sont pas compressées avec un facteur plus important que 30%.

Facteur de compression	nb moyen de STIP/image	Précision
0%	29	96,6%
10%	29	96,6%
20%	29	96,6%
30%	30	96,6%
40%	38	93,3%
50%	44	84,2%
60%	51	79,5%
70%	62	76,5%
80%	77	69,3%
90%	90	58,4%
100%	118	45,8%

80 séquences de synthèse et de sauts d'athlétisme
$k = 0,04$, $\sigma_s = \sigma_t = 1.5$ et $S = 150$

Tableau 3.4 — Influence du facteur de compression MPEG2.

3.3.2 Détection de transitions

3.3.2.1 État de l'art

Dans une séquence vidéo, les plans sont assemblés les uns aux autres en utilisant des *transitions vidéo* (voir la Figure 3.13). Les transitions vidéo sont des effets visuels qui font le

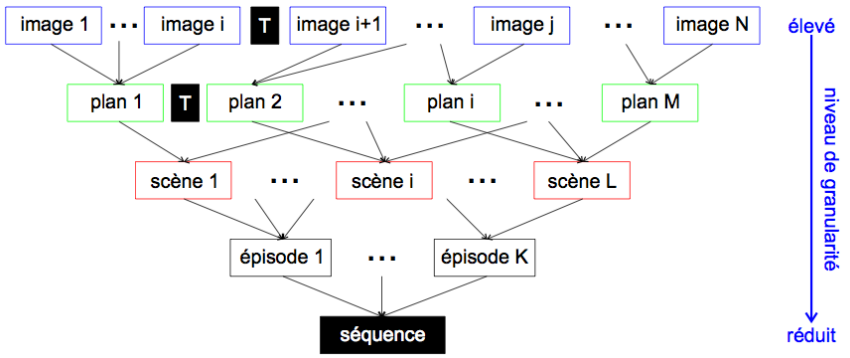


Figure 3.13 — La structure hiérachique d'une séquence d'images (T est une transition vidéo) - source [Ionescu, 2007].

lien entre des plans différents. En fonction des transformations 2D de l'image utilisées, les transitions peuvent se diviser en 5 classes [Lienhart, 2001] :

- **la classe d'identité** : transitions n'apportant aucune modification des plans et n'ajoutant pas d'image. Dans cette catégorie, on trouve les "cuts" permettant de créer la discontinuité entre deux scènes(voir la Figure 3.14) ;
- **la classe spatiale** : transitions présentant des transformations spatiales comme les "wipes", "mattes", "pages turns", "slides", etc.
- **la classe chromatique** : transitions présentant des transformations de couleurs, comme les "fades" et les "dissolves" (voir la Figure 3.14). Un "fade" transition faisant passer d'une image de la séquence vers une image de couleur uniforme ("fade-out") ou l'inverse ("fade-un"). L'enchainement d'un "fade-out" et d'un "fade-in" est appelé un "dissolve".
- **la classe spatio-chromatique** : transitions combinant les caractéristiques de la classe spatiale et de la classe chromatiques comme par exemple les effets de morphing ou un "dissolve" présentant un mouvement de caméra.
- **la classe temporelle** : transitions composées de mouvements de translation de la caméra, comme les "tilt" ou les "pan".

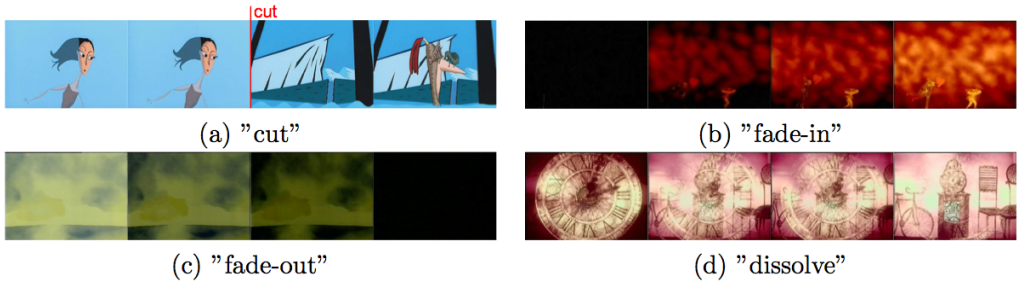


Figure 3.14 — Quelques exemples de transitions vidéos sur des films d'animation : (a) Film "François le Vaillant" [6], (b) et (d) Film "Coeur de Secours" [7]et (c) Film "Le moine et le poisson" [8]

Pour chacune des ces transitions, un certain nombre d'approches ont été proposées pour mesurer la discontinuité visuelle. Elles peuvent être classées de la manière suivante :

- les méthodes utilisant l'intensité des pixels [Otsuji *et al.*, 1991] ;
- les méthodes basées sur l'analyse des contours [Heng et Ngan, 1999] ;
- les méthodes basées sur l'analyse du mouvement [Pawar *et al.*, 2007] ;
- les méthodes développées dans le domaine compressé [Bouthemy *et al.*, 1997; Yu et Wolf, 1999; Fernando et Loo, 2004; Varandas, 2008] ;
- les méthodes utilisant l'analyse de similarité entre images [Guironnet, 2006; Ionescu, 2007] ;
- les méthodes travaillant la détection et le suivi d'objets [Heng et Ngan, 2003].

En analysant l'évolution temporelle du nombre de STIP détectés, nous avons pu constater des configurations particulières lors des transitions entre plans (une hausse très importante du nombre de STIP due au chevauchement de deux plans différents dans le filtre temporel). Aussi avons nous envisagé l'utilisation des STIP pour détecter des transitions. Dans

8. Studio Folimage. http ://www.folimage.com
8. Centre International du Cinéma d'Animation (CICA) - http :// www.annecy.org
8. Studio Folimage. http ://www.folimage.com

la suite de ce paragraphe, nous allons analyser cette capacité des STIP. L'approche envisagée appartient à la catégorie des méthodes utilisant "l'intensité des pixels".

3.3.2.2 La méthode proposée

Fonction d'activité

Dans [Laganière *et al.*, 2008], Laganière définit une fonction "activité" à partir du nombre de STIP détectés par image avec l'objectif de créer automatiquement un résumé vidéo. L'évolution temporelle de cette activité permet de détecter des informations intéressantes d'un point de vue sémantique. En particulier, les maxima temporels locaux de cette fonction d'activité sont généralement liés à des événements importants dans la séquence tels que les transitions (exemple des "dissolves" - figure 3.15) ou un changement entre deux phases d'un même mouvement.

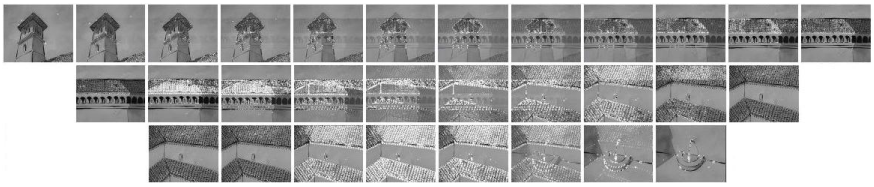


Figure 3.15 **—** Exemples de transitions "dissolve" pour "Le moine et le poisson".

Nous avons repris cette mesure d'activité, et l'avons exploitée pour la détection de différents types de transition. L'exploitation de la fonction "activité" requiert deux étapes :

1. Mesure du nombre $a(t)$ de STIP par image (où t représente le temps).
2. Lissage de $a(t)$ par un filtre moyenneur. Le but est de lisser la fonction d'activité généralement bruitée. La taille du filtre moyenneur est choisie en adéquation avec l'écart-type σ_t de la gaussienne utilisée dans la détermination des STIP. A titre d'exemple, pour $\sigma_t = 1,5$, ce qui correspond à une fenêtre temporelle de recherche des STIP de 11 images, on choisit un filtre moyenneur de taille 11. On note $a_{filt}(t)$ la fonction d'activité filtrée.

Sur la figure 3.16, on peut voir l'évolution de $a_{filt}(t)$ (courbe rouge) et les transitions réellement présentes (lignes vertes). Un certain nombre de transitions brutales ("cuts") présentes dans la séquence semblent correspondre à un pic de l'activité enregistrée. Les transitions plus lentes telles que les "fades" ou les "dissolves" correspondent à des configurations typiques de croissance ou de décroissance de $a_{filt}(t)$.

Selon le type de transitions, l'exploitation de la fonction "activité" va donc être différente.

Détection des "cuts"

Un "cut" est modélisé par un pic d'activité très court et très important. Il se traduira donc

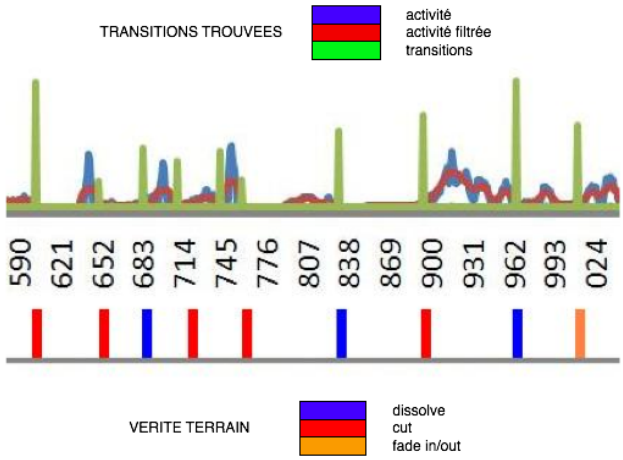


Figure 3.16 — Transitions trouvées sur un passage du "Le moine et du poisson".

par un maximum local de $a_{filt}(t)$ satisfaisant à la condition suivante :

$$0.8 \times a_{filt}(t-\alpha) \leq a_{filt}(t) \geq 0.8 \times a_{filt}(t+\alpha) \tag{3.11}$$

où α est la demi-taille du filtre moyenneur utilisé pour lisser la fonction $a(t)$. Le coefficient 0.8 a été choisi de manière empirique après une première évaluation effectuée sur 250 plans (voir [Laganière *et al.*, 2008]).

La méthodologie utilisée consiste à chercher sur l'ensemble des images déjà traitées le maximum d'activité $a(t)$ et de fixer le seuil de détection de "cut" comme étant la moitié du maximum d'activité atteint. Cette recherche est réalisée dynamiquement, c'est à dire qu'à chaque image, si son activité est supérieure à la valeur maximale mémorisée au préalable, alors celle-ci est modifiée ainsi que le seuil de détection associé. La première apparition d'un "cut" permet d'établir une valeur satisfaisante de seuil et toutes détections précédant ce premier "cut" génèrent un faux positif. L'application d'un seuil adaptatif permet d'utiliser la méthode quelle que soit la résolution de l'image, le nombre de points et donc la valeur d'activité étant très dépendante de ce paramètre.

Détection des "fades"

Les "fade-out" et "fade-in" sont modélisés par une plage où l'activité est très faible. Pour détecter cette plage nous avons utilisé la méthodologie suivante :

On parcourt les images et on calcule la fonction d'activité. On détecte la valeur nulle si on ne trouve aucun point d'intérêt sur α images consécutives. Le critère de détection est donc très restrictif et convient bien pour les vidéos dont la résolution est plutôt petite et les vidéos les moins compressées (moins sujettes aux divers artefacts de compression). Dans le cadre général, on peut considérer avoir détecté la valeur nulle si on trouve moins d'une

dizaine de points par l'image sur α images consécutives.

Un "fade-out" est la zone précédant la plage de faible activité alors que le "fade-in" la suit.

Dans les deux cas, il est important de noter qu'une absence de mouvement dans une image peut être considéré comme le passage à un "fade" par la méthode proposée. Dans le but de corriger ce problème, nous avons ajouté un critère de détection de "fade" : le passage à la couleur uniforme (en l'occurrence le noir, mais la méthode est généralisable à la notion de couleur uniforme dans l'image). On considère le passage à un "fade" lorsqu'on passe par une valeur d'activité nulle et une image présentant une couleur uniforme (écart-type de la couleur inférieur à 5).

Détection des "dissolves"

Les dissolves sont les transitions les plus difficiles à détecter. Comme les "cuts", ils peuvent être modélisés par un maximum local de $a_{filt}(t)$, mais ce maximum est plus large et moins important que pour un "cut". La méthodologie utilisée est la suivante :

On effectue le calcul de l'activité en utilisant plusieurs α différents. Les valeurs de α utilisées sont supérieures à celle utilisée pour la détection des "cuts" et selon l'augmentation, permettent de détecter des "dissolves" plus ou moins longs. Nous avons fixé deux seuils : $\alpha_{dissolve-court}$ et le $\alpha_{dissolve-long}$ définis respectivement comme $arrondi_{impair}(2 * \alpha)$ et $arrondi_{impair}(4 * \alpha)$ soit $arrondi_{impair}(2 * \alpha_{dissolve-court})$. De plus, afin d'éliminer les faux-positifs pouvant provenir de "cuts", on considère une détection valide si et seulement si, on n'a pas détecté une transition "cuts" à $\pm\alpha$ images de la position courante.
On parcourt donc les images et on calcule $a_{filt}(t)$ pour α, $\alpha_{dissolve-court}$ et $\alpha_{dissolve-long}$. Le seuil de détection est la moitié du maximum d'activité atteint par une transition "dissolve".

Critères de performances

Pour évaluer les performances obtenues dans la détection des transitions grâce aux STIP, nous avons classiquement utilisé la *Precision* et le *Rappel* :

$$Precision = \frac{BD}{BD + FD}, Rappel = \frac{BD}{NT} \tag{3.12}$$

où BD est le nombre de bonnes détections, FD le nombre de fausses détections et NT est le nombre total de transitions dans la séquence. La vérité terrain a été obtenue manuellement. Dans le calcul de ces critères, nous avons introduit une tolérance vis-à-vis de la localisation temporelle des transitions. Pour ce qui concerne les "cuts", une transition est considérée détectée si le pic considéré est situé à $\pm 3 images$ de la vérité terrain. En ce qui concerne les "fades" et les "dissolves", la tolérance est un peu plus large. La transition est considérée comme détectée si elle est située à $\pm 5 images$ de la vérité terrain.

3.3.2.3 Résultats expérimentaux

Pour ces tests, les séquences utilisées sont des films d'animation et les vidéos sportives. Dans les films d'animation, les transitions présentes sont les "cuts", les "fades" et les "dissolves". Les vidéos sportives ne contiennent qu'un type de transition, des "cuts". Les résultats obtenus sont résumés dans le tableau ci-dessous (3.5).

	Précision	Rappel
Sports - "cuts"	0.92	1.00
Animation - "cuts"	0.93	1.00
Animation - "fade in"	0.82	0.88
Animation - "fade out"	0.80	0.92
Animation - "dissolve"	0.86	0.86

Test - 20 séquences (divers sports séparés par des "cuts")
ainsi que 60 séquences d'animations
$k = 0,04$, $\sigma_s, \sigma_t = 1.5$ et $S = 150$

Tableau 3.5 — Rappel/Précision obtenus sur la détection des transitions.

Pour les vidéos sportives, on observe un très bon rappel et une très bonne précision. En effet, les images avant et après un "cut" sont en général très différentes ce qui entraîne une variation importante du nombre de STIP. Les seuls faux positifs (dans notre test, 2 faux positifs) que l'on trouve se situent lors d'un changement brusque très important dans le mouvement qui élève le niveau d'activité (le nombre de points d'intérêt) de manière très importante le rapprochant d'une valeur d'activité que l'on retrouve au moment d'une transition de type "cut". Ces faux positifs peuvent être supprimés en élevant le niveau du seuil de détection mais au risque de ne plus détecter certaines transitions. La phase d'initialisation de notre algorithme génère d'autres faux positifs correspondant aux changements dans les mouvements précédant la première transition, qui sont détectés comme des transitions tant que le seuil n'a pas été ajusté par la première transition. Un travail en 2 passes permettrait la suppression de ces faux positifs (ce travail en 2 passes n'a pas été réalisé puisqu'il double le temps de calcul pour chaque plan considéré et que notre objectif est d'extraire le maximum d'informations tout en minimisant le temps de calcul).

Dans les films d'animation, pour les transitions de type "cut", les résultats sont très bons et assez proches de ceux observés pour les séquences sportives et ce, pour les mêmes raisons. En ce qui concerne, les fades et les dissolves, les performances sont un peu moins bonnes (précision et rappel de l'ordre de 0.8/0.9 pour les "fades", de l'ordre de 0.86 pour les "dissolves"). Ces résultats restent néanmoins tout à fait corrects pour ce type de transition plus délicates à détecter.

On constate que l'augmentation de la valeur de σ_t améliore sensiblement les performances comme le tableau 3.6 nous l'indique. Néanmoins l'augmentation du lissage temporel a pour effet de diminuer le nombre de STIP et finit par ne plus en générer aucun.

Méthode	Précision	Rappel
Méthode Ionescu - fade in	0.82	0.94
Méthode Ionescu - fade out	0.94	1.00
$\sigma_t = 1.5$ - fade in	0.82	0.88
$\sigma_t = 1.5$ - fade out	0.80	0.92
$\sigma_t = 1.7$ - fade in	0.90	0.97
$\sigma_t = 1.7$ - fade out	0.95	1.00
$\sigma_t = 2.0$ - fade in	0.88	0.97
$\sigma_t = 2.0$ - fade out	0.91	1.00

Test - 60 séquences d'animations
$k = 0,04$, σ_s et $S = 150$

Tableau 3.6 — Influence du paramètre σ_t sur la détection des "fades".

3.3.2.4 Comparaison

Les travaux de Ionescu [Ionescu, 2007] ont notamment porté sur la détection des transitions en utilisant une méthode basée sur l'analyse d'histogrammes couleurs. Son approche ayant été évaluée sur la base de films d'animation dont nous disposons également, il est intéressant de comparer ses travaux avec notre approche à base de suivi d'activité.

La détection des "cuts"

Ionescu a développé des algorithmes de détection de "cuts" adaptés aux particularités des films d'animation. Les méthodes qu'il propose sont basées sur l'analyse des histogrammes couleurs par cadrans, ces méthodes étant plus efficaces que les approches basées sur l'analyse du mouvement ou l'analyse des contours. La dissimilarité visuelle introduite par les "cuts" est transformée en une distance entre histogrammes couleurs. Avant que la détection soit effectuée, un certain nombre de pré-traitements doivent être réalisés : les sous-échantillonnages (un temporel et un spatial) et une réduction des couleurs. Parmi les différentes méthodes proposées, nous retiendrons la plus performante notée *Ionescu* dans la suite.
La comparaison avec notre approche a été testée sur deux films d'animation de longue durée : "A Bug's Life" 84min46s et 1597 "cuts", et "Toy Story" 73min18s et 1569 "cuts", ce qui fait une durée totale de 158min et comportant 3166 "cuts".

Méthode	Précision	Rappel
Ionescu	0.96	0.92
Niveau d'activité	0.96	0.95

Test - 60 séquences d'animation
$k = 0,04$, $\sigma_s, \sigma_t = 1.5$ et $S = 150$

Tableau 3.7 — Comparaison dans la détection des "cuts".

En ce qui concerne la détection des transitions de type "cuts", le tableau 3.7 montre que la méthode basée sur les points d'intérêt permet d'obtenir des performances équivalentes pour les fausses détections (même valeur de précision) et légèrement meilleures pour les non-détections (valeur de rappel supérieure).

La détection des "fades"

La méthode de détection de "fades" proposée par Ionescu est inspirée des travaux présentés dans [Fernando *et al.*, 1999] et utilise l'hypothèse que lors d'un "fade in" l'intensité lumineuse augmente progressivement jusqu'à la fin de la transition où elle devient approximativement constante.
Cette méthode a été testée sur la base composée de 14 films d'animation de longue durée, ce qui représente 37 "fade in" et 56 "fade out". De notre côté, nous avons évalué notre méthode sur la même base.

Méthode	Précision	Rappel
Méthode Ionescu - fade in	0.82	0.94
Méthode Ionescu - fade out	0.94	1.00
Niveau d'activité - fade in	0.82	0.88
Niveau d'activité - fade out	0.80	0.92

Test - 60 séquences d'animation
$k = 0,04$, $\sigma_s, \sigma_t = 1.5$ et $S = 150$

Tableau 3.8 — Comparaison dans la détection des "fades".

Le tableau 3.8 montre que la méthode basée sur les points d'intérêt ne permet pas d'atteindre les résultats de la méthode proposée par Ionescu mais s'en rapproche toutefois. Avec notre approche, un certain nombre de "fades" ne sont pas détectés et il subsiste des fausses détections. Néanmoins, notre approche présente deux avantages par rapport à celle de Ionescu :

- La méthode de Ionescu a besoin de valeurs de seuils qui sont ajustées automatiquement une fois la séquence entièrement analysée. Elle n'a donc pas le caractère temps réel que présente notre approche.
- Notre approche utilise la fonction d'activité quel que soit le type de transition, alors que Ionescu utilise des approches totalement différentes pour détecter chaque type de transition.

3.3.2.5 Conclusion

La méthode proposée, basée sur l'analyse de l'activité des points d'intérêt, permet d'obtenir en temps réel la détection des transitions de type "cuts" avec des performances très intéressantes. Elle fait même légèrement mieux qu'une méthode dédiée. Pour les transitions de type "fade", en utilisant les mêmes paramétrages que pour les "cuts", la méthode n'est pas satisfaisante mais dès qu'on modifie certains paramètres (notamment σ_t), les résultats

approchent ceux d'une méthode dédiée. En conclusion, la méthode basée sur l'analyse de l'activité des points d'intérêt est une méthode utilisable pour la détection des transitions de type "cuts" et "fades".

3.3.3 Détection de changements significatifs dans les mouvements

Les plans segmentés peuvent contenir une ou plusieurs activités, une activité correspondant à un mouvement cohérent du contenu des images. Par exemple, ce peut-être une activité de course d'un personnage. L'étape de segmentation en plans est souvent suivie de l'étape de segmentation de chaque plan en blocs présentant la même activité.

3.3.3.1 La méthode proposée

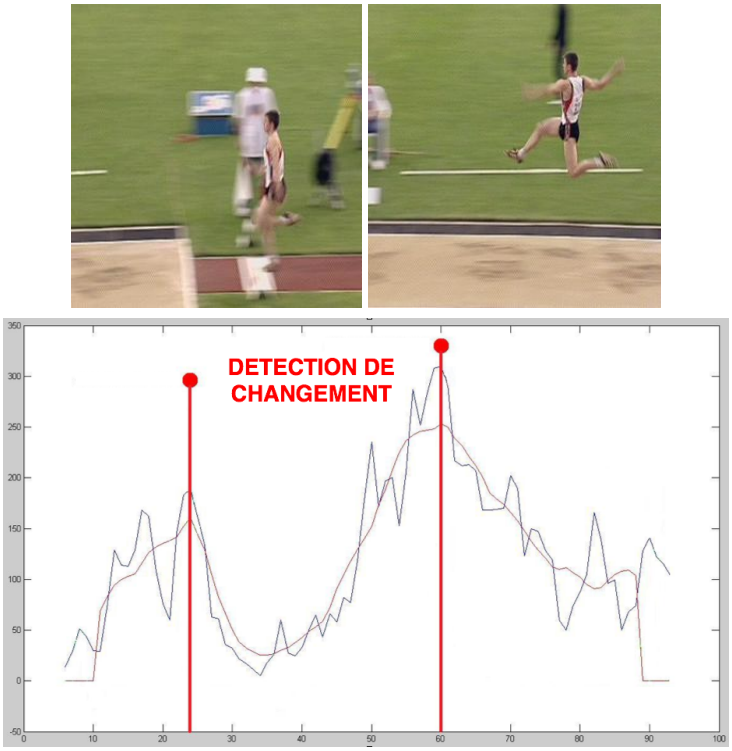
Afin de procéder à cette seconde segmentation, en posant l'hypothèse que le séparateur de deux blocs d'activité distincte est un changement significatif dans le mouvement. La méthode de suivi d'activité proposée dans 3.3.2.2 peut donc être à nouveau appliquée à l'intérieur d'un même plan. La figure 3.17 illustre la détection de deux changements au cours d'un saut en longueur : le premier se situe lors de l'appel sur la planche. C'est la transition entre la phase de course d'élan et la phase ascendante du saut. Le second changement est le moment où l'athlète est au plus haut du saut. Il se situe entre la phase ascendante suivie et la phase descendante du saut.

L'algorithme de détection consiste donc à rechercher les maxima locaux (en utilisant l'équation 3.11), mais sans imposer de seuil sur la valeur de la fonction d'activité $a(t)$. En effet, ces changements correspondent souvent à des amplitudes fortes de $a(h)$, augmentation allant de 50% à 200%.

Dans ce test, ce sont les séquences de sauts d'athlétisme et les séquences de la base 'télévision' qui sont utilisées pour analyser la détection de changement par suivi d'activité. Il est important de noter que ces séquences sont mono-plan. Un changement est défini comme étant une variation importante de trajectoire comme par exemple le passage de la course au saut, ou un virage avec une voiture. Les séquences ont été annotées manuellement. Pour chaque changement considéré, le numéro de l'image centrale du changement a été repéré. On détermine aussi l'erreur de position temporelle en nombre d'images entre la vérité terrain et l'image centrale.

3.3.3.2 Résultats expérimentaux

Pour calculer la précision et le rappel, nous avons fixé expérimentalement un seuil de tolérance sur l'erreur de position temporelle. Sur le tableau 3.9, on peut observer que les transitions sont correctement détectées avec une tolérance comprise entre 3 et 10 images. Les valeurs de précision et de rappel sont relativement élevées puisqu'on obtient une précision moyenne de 0.87 et un rappel moyen de 0.86. Il est important de noter que certains changements présents dans les séquences de saut à la perche sont imprécis. En effet, les



La courbe bleue est le niveau d'activité et la rouge, le niveau d'activité lissé

Figure 3.17 — Détection des transitions lors d'un saut en longueur.

	Précision	Rappel	Tolérance
Saut en longueur	0.93	0.92	$\pm 3 images$
Saut en hauteur	0.92	0.88	$\pm 3 images$
Triple saut	0.81	0.71	$\pm 5 images$
Saut à la perche	0.84	0.85	$\pm 10 images$
Hand-ball	0.76	0.86	$\pm 3 images$
Circulation automobile	0.97	0.93	$\pm 3 images$

20 séquences de saut, 8 séquences de hand-ball et 8 de surveillance routière
$k = 0,04$, $\sigma_s = \sigma_t = 1.5$ et $S = 150$

Tableau 3.9 — Performances de détection de changements significatifs dans les mouvements.

positions des caméras (généralement au niveau de la barre à franchir) doivent effectuer un mouvement important pour suivre l'athlète. C'est ce mouvement de caméra qui atténue la fonction d'activité et diminue les performances de détection. Celles-ci sont également moins satisfaisantes pour le triple saut, le changement de mouvement étant moins marqué. Concernant les vidéos de hand-ball, les changements correspondent aux passes entre les joueurs. Sur ces séquences, les caméras prennent l'action de manière oblique si bien que l'activité est moins importante (voir influence de la caméra - section 3.3.1.1). De plus, l'annotation

manuelle des changements a été plus difficile et un peu moins précise sur ces séquences.

3.3.3.3 Conclusion

La méthode de détection de changements significatifs dans les mouvements par suivi d'activité est une méthode qui fonctionne relativement bien (de l'ordre de 90% en moyenne en précision et en rappel) à part dans quelques conditions spécifiques comme les prises de vue obliques. Elle permet d'effectuer une segmentation satisfaisante d'un plan en blocs d'activité identique et continue. Dans notre objectif d'analyser le mouvement, ce résultat permet une segmentation utilisable pour la compréhension du contenu de la séquence. Pour atteindre une description de plus haut niveau, il sera nécessaire de compléter cette méthode avec l'analyse et le suivi d'autres attributs comme les paramètres du mouvement dominant du personnage ou les vecteurs du flot optique du personnage en mouvement. Il semble également intéressant de noter que cette méthode n'utilise que les points d'intérêt qui sont obtenus en temps réel (au retard du buffer d'images près - voir 3.2.2). Cette méthode pourrait donc être particulièrement intéressante dans des applications concernant la surveillance de trafic automobile ou de la surveillance dans le métro par exemple en permettant de détecter les changements de comportement d'un élément parmi un groupe ou même d'un groupe dans sa globalité. Les attributs de changements sont notés 'nbp' dans la suite.

3.3.4 Détection d'objets en mouvement par les STIP

Les points d'intérêt spatio-temporels sont définis comme des points présentant des discontinuités dans l'espace et dans le temps. Ainsi, un objet présentant des coins générant un point d'intérêt spatial et ayant un mouvement irrégulier va générer des points spatio-temporels (cf. 3.3.1). Dès lors, nous nous sommes interrogés sur la capacité des points d'intérêt spatio-temporels à indiquer la présence d'un objet en mouvement et à préciser sa localisation dans l'image.

Dans cette expérience, nous allons extraire les objets en mouvement et les points d'intérêt et analyser la position des points par rapport aux objets en mouvement.

"Motion2D" permet également de calculer la transformation globale pour compenser le mouvement dominant entre 2 images. A partir de là, une soustraction d'images permet la séparation des pixels appartenant au mouvement dominant de ceux n'y appartenant pas. Afin d'augmenter la précision du masque, il est tout à fait possible de prendre en compte plus de deux images. Dans notre cas, nous avons utilisé deux images en plus de l'image courante : une image avant et une image après. Nous obtenons ainsi des masques comme sur la figure 3.18 obtenus sur une vidéo de saut en longueur, dans la phase d'élan du sauteur. Afin d'obtenir le masque final, on applique un filtrage morphologique de type ouverture afin d'isoler les surfaces utiles, lisser les contours et éliminer les fausses détections de taille inférieure à l'élément structurant (dont la taille a été fixée empiriquement à 5 après une série de tests préliminaires montrant qu'une taille égale à 3 filtre peu et qu'au dessus de 5, le filtrage est trop fort). Ce paramètre est également en liaison directe avec la taille de la fenêtre de calcul et de filtrage des STIP qui prend en compte 5 images avant et 5 images après l'image courante).

Figure 3.18 — Masque des pixels en mouvement (n'appartenant pas au mouvement dominant) avant et après filtrage morphologique, obtenu en utilisant Motion2D.

Critère de performances

Le critère de performance est la précision. Dans cette évaluation, il n'est pas possible d'évaluer le rappel car les points d'intérêt ne sont que des points de détection et ne sont pas censés couvrir le masque entier. Le principe est de vérifier si les STIP détectés sont ou non dans le masque de l'objet en mouvement obtenu par l'une des méthodes décrites au paragraphe 3.1.4.1. On distingue deux cas de figure :

- un point est dans le masque (vrai positif) : il est donc sur un objet en mouvement, il est compté comme ayant détecté un objet en mouvement ;
- un point n'est pas dans le masque (faux positif) : il n'est pas sur un objet en mouvement, il n'est pas compté comme ayant détecté un objet en mouvement.

Pour chaque objet en mouvement, on dispose donc du nombre de points d'intérêt dans l'objet en mouvement (NVP) et du nombre de points d'intérêt hors l'objet en mouvement (NFP). La précision est définie par : $P = \frac{NVP}{NVP+NFP}$.
NVP (respectivement NFP) représentant le nombre de vrai (respectivement faux) positifs mesurés sur la séquence.

3.3.4.1 Résultats expérimentaux

L'expérimentation est effectuée sur 60 séquences d'athlétisme (les 4 différents sauts, 15 séquences par saut) durant approximativement 4-5 secondes et sur des séquences de films d'animation (60 séquences également).

Le tableau 3.10 montre qu'on obtient d'assez bons résultats (précision moyenne de 78%) en utilisant les STIP comme détecteurs d'objets intéressants, même dans les cas ou plusieurs objets se déplacent dans la même image (c'est le cas de près de la moitié des plans de la catégorie 'films d'animation'). La précision augmente même lorsque le nombre d'objets est

	Précision
Films d'animation	0.73
Vidéos d'athlétisme	0.81

Test - 60 séquences d'athlétisme (5 secondes)
et 60 séquences de films d'animations
$k = 0,04$, $\sigma_s = \sigma_t = 1.5$ et $S = 150$

Tableau 3.10 — Performances de détection d'objets en mouvement.

supérieur à 1 (de l'ordre de 2 ou 3). Pour conclure, nous pouvons souligner que les STIP montrent des performances intéressantes pour détecter des objets en mouvement. Ils pourraient donc être utilisés lors d'une phase de pré-traitement suivie par une méthode de localisation, et éventuellement de segmentation et de suivi.

3.3.5 Étude de la saillance visuelle des SIP/STIP

Dans le but de qualifier l'intérêt des SIP/STIP, nous nous sommes interrogés sur les liens entre saillance (au sens de l'utilisateur) et intérêt (mesuré par les SIP/STIP). Les informations de saillance sont obtenues en analysant le contenu des images ou des séquences d'images et sont supposées faire ressortir les zones susceptibles d'attirer le regard humain. Les informations utilisées pour les construire sont des informations de luminance et de mouvement.
Quelle est la différence entre une région intéressante et une région saillante ? Une région intéressante est-elle saillante ? et inversement une région saillante est-elle intéressante ? Les études sur ces questions proposent souvent leurs propres définitions de la saillance et de l'intérêt. Dans un récent article Masciocchi [Masciocchi *et al.*, 2009] propose de comparer les régions retournées comme saillantes par le modèle d'Itti [Itti *et al.*, 1998] aux régions qui ont été sélectionnées comme les plus intéressantes par des sujets. Dans cette expérience la notion d'intérêt se réfère aux sujets et peut donc être subjective. Dans une autre étude, Privitera [Privitera et Stark, 2000] fait référence à des régions d'intérêt définies en utilisant différents algorithmes dont certains proches des points d'intérêt et d'autres proches de principes biologiques du système visuel. Ces régions d'intérêt sont ensuite comparées aux régions fixées par un utilisateur.

Nous allons dans cette partie nous intéresser à la saillance visuelle et étudier si les points d'intérêt permettent de prédire les régions regardées par les sujets. Nous allons pour cela comparer les points d'intérêt et les positions oculaires des sujets. Cette étude a été menée conjointement avec Sophie Marat [Simac-Lejeune *et al.*, 2009], qui a élaboré une expérience d'oculométrie pour étudier la saillance visuelle. Nous décrivons ici les principaux résultats obtenus dans sa thèse.

3.3.5.1 Les cartes de saillance

S. Marat [Marat *et al.*, 2009] propose un modèle de saillance à deux voies permettant l'extraction d'une carte de saillance statique et d'une carte de saillance dynamique. Bien qu'en

grande partie indépendants, les traitements mis en œuvre sur chacune des deux voies utilisent des modules communs (filtres rétiniens, filtres corticaux) s'inspirant du système visuel humain. En général, les cartes statiques font ressortir "les parties texturées de l'image différentes de leur voisinage" [Marat *et al.*, 2009] et les cartes dynamiques "mettent en évidence ce qui bouge" [Marat *et al.*, 2009]. Dans la suite, ces cartes seront appelées $M_s(x, y, t)$ pour la saillance statique et $M_d(x, y, t)$ pour la saillance dynamique. S. Marat [Marat *et al.*, 2009] a également proposé une carte fusionnant les cartes statiques et dynamiques. Cette carte, dite de fusion renforcée sera notée $M_{rsd}(x, y, t)$ dans la suite.

3.3.5.2 Présentation de l'expérience

Afin d'avoir des données "réelles" ou "vérités terrain", Marat dans [Marat *et al.*, 2009] a mis en place une expérience d'oculométrie permettant d'enregistrer les positions oculaires de plusieurs sujets regardant librement des vidéos, dans le but d'évaluer son modèle de saillance visuelle et vérifier l'exactitude de ces cartes de prédiction des positions oculaires.

Le système utilisé est un oculomètre Eyelink II (SR Research [9]) composé de trois caméras miniatures montées sur un casque. Ces caméras situées devant chaque oeil et sur le front du sujet permettent de calculer la position des yeux et de la tête par rapport à l'écran que le sujet regarde. L'oculomètre enregistre les positions oculaires à une fréquence temporelle de 500 Hz.

Quinze sujets ont passé l'expérience : 3 femmes et 12 hommes. Leur âge variait de 23 à 40 ans. Tous les participants avaient une vue normale ou corrigée. Les participants étaient naïfs quant au but de l'expérience et avaient pour consigne de regarder les vidéos librement sans contrainte. Pendant l'expérience les sujets munis de l'oculomètre étaient assis, leur menton posé sur une mentonnière, en face d'un écran de 21". La résolution de l'écran était de 1024×768 pixels et sa fréquence de rafraîchissement de 75 Hz. L'écran se trouvait à une distance de 57 cm, ce qui correspond à un champ visuel de 40×30. Une calibration en 9 points était réalisée au début de l'expérience ainsi que tous les 5 stimuli (un stimulus désigne tout ce qui est de nature à déterminer une excitation chez un organisme vivant - dans l'expérience, c'est un stimuli visuel à savoir une séquence vidéo présentant du mouvement), de plus un recadrage de contrôle permettait de recentrer le regard avant chaque stimulus.

Cette expérience et plus particulièrement les stimuli sont inspirés par une expérience de Carmi et Itti [Carmi et Itti, 2006]. Cinquante-trois vidéos (25 images par seconde, 720×576 pixels par image) ont été sélectionnées en provenance de diverses sources : films et séries, émissions de télévision, journaux télévisés, films d'animation, publicités, émissions sportives, clips musicaux et concerts. Ces vidéos ont été choisies pour représenter des scènes dynamiques aussi variées que possible : 53 vidéos rassemblent des scènes d'intérieur, d'extérieur, des scènes de jour et de nuit. Ces vidéos ont été découpées en extraits de 1 à 3

9. http ://www.sr-research.com/

secondes pour former 305 extraits appelés *snippets*. Les snippets ont été formés de manière à ce qu'aucun changement de plan ne se produise. Ces extraits ont ensuite été concaténés pour former 20 *clips* d'environ 30 secondes chacun. Pour chaque clip, on trouve au plus un snippet provenant d'une source donnée. L'enchaînement des snippets ainsi que leur durée ont été choisis de manière aléatoire afin d'empêcher les sujets d'anticiper les transitions. Les transitions entre les snippets ne sons pas cachées ou atténuées. Il est important de préciser que, comme le modèle de saillance proposé par Marat, l'extracteur de points d'intérêt ne considère que les informations d'intensité, les vidéos étant transformées en niveaux de gris avant d'être présentées aux sujets.

3.3.5.3 Carte de densité de positions oculaires

Comme indiqué dans la présentation de l'expérience, nous analysons les positions des yeux plutôt que les fixations. L'oculomètre permet d'enregistrer les positions oculaires à la fréquence de 500 Hz, ce qui correspond à 20 positions oculaires (10 par œil) pour chaque image d'une vidéo ayant une vitesse de 25 images/secondes. Pour chaque image, pour chaque sujet, nous calculons la position médiane des 20 positions enregistrées. La position horizontale (respectivement verticale) finale est obtenue en calculant la position médiane des positions horizontales (respectivement verticales) des 20 points. Pour chaque image, nous avons une position médiane par sujet, nous obtenons donc 15 positions (15 points) par image. Si un sujet cligne des yeux, aucune position valide n'est enregistrée et il y a un point de moins sur l'image correspondante. Pour chaque image à l'instant t, nous construisons une carte de positions oculaires $M_p(x,y,t)$:

$$M_p(x,y,t) = \sum_{i=1}^{N} \delta(x-x_i, y-y_i) \tag{3.13}$$

$$et \qquad \delta(x-x_i, y-y_i) = \begin{cases} 1 & si \quad x = x_i \quad et \quad y = y_i \\ 0 & sinon \end{cases} \tag{3.14}$$

avec N le nombre de sujets, (x_i, y_i) la position oculaire médiane pour le sujet i, et δ le symbole de Kronecker.

3.3.5.4 Normalized Scanpath Saliency (NSS)

Le critère 'Normalized Scanpath Saliency' (*NSS*) a été défini par Peters et Itti [Peters *et al.*, 2005],[Peters et Itti, 2008] pour comparer une carte de saillance aux positions oculaires des sujets. Il a été utilisé par Marat [Marat *et al.*, 2009] pour évaluer la pertinence de son modèle de saillance visuelle.

Le *NSS* est calculé à l'aide de l'équation suivante :

$$NSS(t) = \frac{\overline{M_h(x,y,t) \times M_m(x,y,t)} - \overline{M_m(x,y,t)}}{\sigma_{M_m(x,y,t)}} \tag{3.15}$$

avec $M_h(x,y,t)$ la carte de densité de positions oculaires normalisée (de manière à avoir une moyenne égale à 1) et $M_m(x,y,t)$ la carte de saillance donnée par le modèle envisagé.

La notation $\overline{X}$ désigne la moyenne et $\sigma_{M_m(x,y,t)}$ l'écart-type de $M_m(x,y,t)$. Le modèle peut être la carte statique ou dynamique de Marat ou la carte fournie par les SIP ou les STIP. Le *NSS* peut être interprété comme la divergence des résultats expérimentaux (positions oculaires) par rapport à la moyenne du modèle, exprimée en nombre d'écarts-types du modèle. Si le *NSS* est nul, il n'y a pas de lien entre les positions regardées et la saillance. Ceci se retrouve immédiatement dans l'expression du NSS (éq. 3.15) : si les cartes sont indépendantes, la moyenne du produit est égale au produit des moyennes, et la normalisation à 1 de la moyenne de la carte $M_m(x,y,t)$ permet d'avoir alors un numérateur nul dans l'expression du NSS. Si le *NSS* est négatif, les positions oculaires se trouvent sur des régions non saillantes. Si le *NSS* est positif, les positions regardées se trouvent sur des régions saillantes. Plus le *NSS* a une valeur positive grande et plus les points fixés sont saillants, au sens du modèle utilisé.

3.3.5.5 Comparaison

La comparaison est effectuée en utilisant la base de vidéos présentées dans le paragraphe 3.3.5.2. Le critère de comparaison utilisé est le *NSS* (éq. 3.15) en utilisant pour carte modèle M_m la carte d'intérêt SIP M_{SIP} ou la carte d'intérêt STIP M_{STIP}. Pour introduire une certaine tolérance, ces cartes sont lissées par un filtrage gaussien. Les différentes cartes obtenues sont illustrées à la figure 3.19.

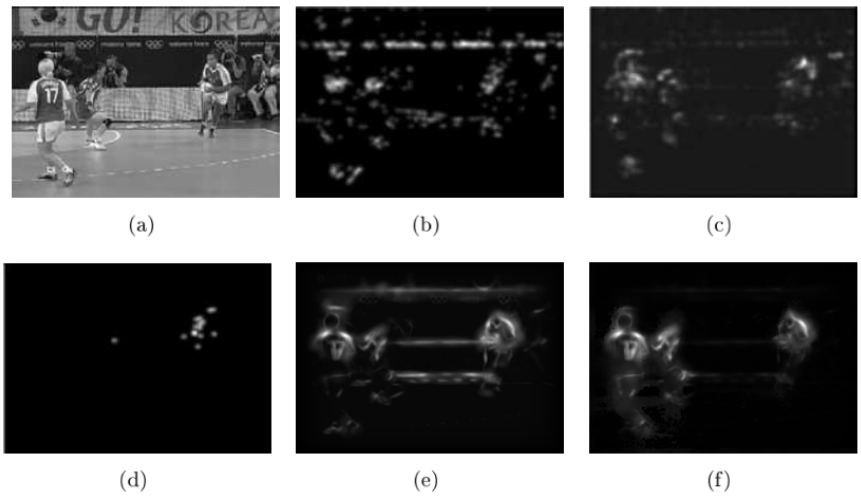


Figure 3.19 — Exemple de cartes d'intérêt et de cartes de saillance sur une image choisie de la base test de vidéos : (a) image originale, (b) carte d'intérêt SIP M_{SIP}, (c) carte d'intérêt STIP M_{STIP}, (d) carte de densité de positions oculaires M_h, (e) carte de saillance statique M_S, (f) carte de saillance fusionnée M_{Rsd}.

Les cartes données par les points d'intérêt, qui représentent plutôt des concentrations de points d'intérêt, n'ont pas tout à fait la même allure que les cartes de saillance qui elles mettent en valeur des contours ou des régions. Néanmoins, ce sont sensiblement les mêmes régions qui sont mises en évidence dans les deux cas. Nous avons calculé la moyenne du *NSS* sur toutes les vidéos pour les deux cartes d'intérêt M_{SIP} et M_{STIP}. Pour les cartes M_{SIP}

cette moyenne vaut *0.50* et pour les cartes M_{STIP} elle est de *0.49*, valeurs proches. Néanmoins, les cartes M_{SIP} sont légèrement meilleures que les cartes M_{STIP}. Ceci peut venir du fait que les cartes M_{STIP} sont une restriction des cartes M_{SIP} aux points d'intérêt spatiaux qui ont une variation temporelle irrégulière.

Critères	NSS			
Carte	M_S	M_{Rsd}	M_{SIP}	M_{STIP}
Moyenne	0.68	1.07	0.50	0.49

Test - 53 séquences télévisées
STIP $k = 0,04$, $\sigma_s = \sigma_t = 1.5$ et $S = 150$

Tableau 3.11 — *NSS* moyen des cartes de saillance statique M_S, fusion renforcée des voies statique et dynamique M_{Rsd} et des cartes d'intérêt spatiales M_{SIP} et spatio-temporelles M_{STIP} sur toute la base de vidéos courtes.

Si on souhaite comparer les résultats obtenus avec les points d'intérêt à ceux obtenus avec le modèle de saillance, il convient de comparer des cartes de même nature. Ainsi, on compare la carte M_{SIP} avec la carte M_S puisqu'elles donnent toutes deux une information spatiale alors que l'on compare la carte M_{STIP} avec la carte M_{Rsd} car elles donnent une informations spatio-temporelles. Le NSS moyen obtenu sur les différentes cartes citées est donnée dans le tableau 3.11. Le modèle de saillance donne de meilleurs résultats tant en spatial qu'en spatio-temporel. Cependant, pour obtenir un bon NSS, il est nécessaire d'obtenir une corrélation entre les positions oculaires et le modèle. Le modèle de saillance retournant des contours, un bon NSS indique que les positions oculaires de la carte de positions oculaires M_h sont situées sur les contours. Le modèle d'intérêt retournant des points, un bon NSS indique alors que les positions oculaires sont réparties sur les points. Le critère est donc peut-être un peu plus 'dur' pour les points d'intérêt.

Cependant, il est important de noter que les deux modèles ne travaillent pas avec le même type de données : le modèle de saillance renvoie des contours alors que le modèle d'intérêt des points, lissés par une fonction gaussienne.

Pour préciser cette analyse, nous avons tracé l'évolution du *NSS* au cours du temps pour les cartes M_s , M_{Rsd} , M_{SIP} et M_{STIP} en figure 3.20.

On constate que les courbes M_{SIP} et M_S ainsi que les courbes M_{STIP} et M_{Rsd} ont des allures proches. Elles ont de faibles valeurs au début puis augmentent rapidement (rebond) avant de se stabiliser et d'osciller autour d'une valeur constante. La carte M_{SIP} met en évidence les coins spatiaux et M_{STIP} retient ceux qui sont aussi temporels ainsi ces deux cartes obtiennent des résultats très proches. Les courbes obtenues par le modèle de saillance donnent de meilleurs résultats que celles du modèle d'intérêt comme on s'y attendait au vu de la valeur moyenne.
A noter qu'à partir de la cinquantième image, il ne reste que le tiers des snippets concernés par le calcul de la moyenne (snippets plus long que 50 images).

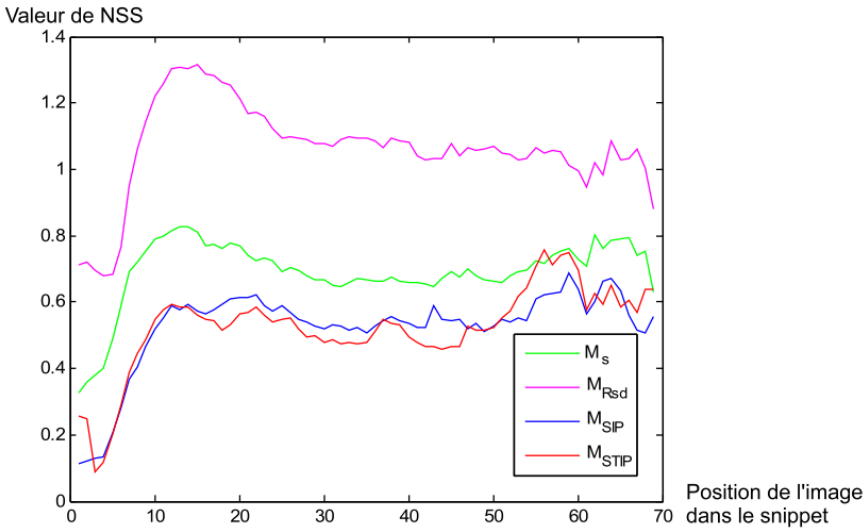


Figure 3.20 — Évolution du *NSS* en fonction de la position des images dans les snippets pour les cartes d'intérêt M_{SIP} et M_{STIP} et pour les cartes de saillance M_S et M_{Rsd}.

3.3.5.6 Analyse par catégorie sémantique

Pour compléter cette étude, nous nous sommes demandés si les points d'intérêt étaient pertinents pour les différentes catégories sémantiques présentes dans les snippets. Une analyse montre que les performances sont bien différentes en fonction du contenu du snippet. Sur les 305 snippets, nous avons extrait un certain nombre de classes ayant le même contenu. Nous avons choisi en particulier 4 classes particulièrement représentées : la circulation automobile (18 snippets soit environ 6% des snippets), les sports d'équipe (44 snippets soit environ 14% des snippets), les visages et/ou les mains (47 snippets soit environ 15% des snippets) et les groupes de personnes comme les manifestations (30 snippets soit environ 10% des snippets). La figure 3.21 présente quelques images de snippets correspondant à chacune de ces 4 classes.

a - Circulation automobile | b - Sports d'équipe | c - Visage et mains | d - Groupes

Figure 3.21 — Exemples d'image des 4 classes.

Le tableau 3.12 résume les valeurs du NSS obtenues sur les différentes classes. Les résultats montrent que la circulation automobile et les sports d'équipe obtiennent une moyenne et un maximum du NSS meilleurs avec les STIP qu'avec les SIP. En particulier, la circulation automobile obtient une valeur maximale relativement haute. Au contraire, pour

		Circulation automobile	Sports d'équipe	Visages et mains	Groupes
NSS_{SIP}	Moyenne	0.86	0.17	1.85	0.19
	Maximum	2.10	0.72	4.78	0.78
NSS_{STIP}	Moyenne	1.26	0.77	0.39	0.23
	Maximum	4.24	1.98	3.06	0.95
Nombre de snippets		18 (6%)	44 (14%)	47 (15%)	30 (10%)

Note : la valeur minimum de chaque carte d'intérêt est 0

Tableau 3.12 — NSS moyen et maximum pour les différentes classes envisagées.

la classe des visages et des mains, les SIP sont meilleurs en moyenne et sur le maximum que les STIP. Enfin, la valeur du NSS pour la classe des groupes est proche de 0, ce qui indique qu'il n'y a pas de lien relationnel entre les positions oculaires et les points d'intérêts.

Circulation automobile

Le premier exemple regroupe les snippets présentant des évènements de circulation automobile. Ce premier exemple provient de la classe de trafic (image de la figure 3.22), qui présente des séquences de la circulation. La circulation automobile est une classe particulièrement intéressante car elle est souvent caractérisée par des mouvements uniformes mais présentant occasionnellement des discontinuités relativement importantes : changement de file, virage,... Ces discontinuités sont particulièrement bien détectées et renforcées par les STIP. De plus, ces changements attirent facilement l'attention visuelle d'un observateur. Ceci explique les très bons résultats obtenus par le NSS et l'amélioration obtenue par les STIP en utilisant la composante temporelle par rapport au SIP (courbe de la figure 3.22).
Sur cette figure, l'évolution du NSS_{STIP} présente un maximum local (valeur $\simeq 4$) autour de la trentième image correspondant à un changement brutal de direction de l'un des véhicules.

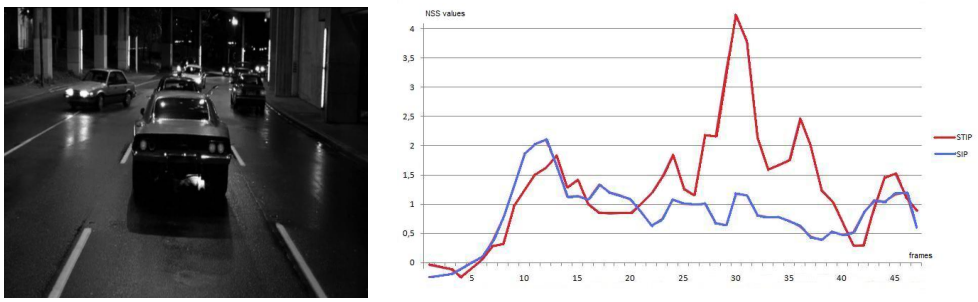


Figure 3.22 — Exemple d'une image et de l'évolution du NSS au cours du temps pour un snippet de la classe "Circulation automobile".

Sports d'équipe (utilisant une balle)

Le second exemple concerne la catégorie des sports d'équipe (image de la figure 3.23) utilisant une balle comme le basketball, le hand ball ou le rugby. Ces sports sont caractérisés par des mouvements rapides, pouvant être considérés comme erratiques et comportant des changements rapides. En outre, plus un joueur est proche de la balle, c'est à dire de l'action, plus les mouvements sont rapides. Ce contexte est favorable aux STIP qui ont tendance à détecter les mouvements irréguliers.
La courbe de la figure 3.23 présente l'évolution du NSS pour le SIP et les STIP. Clairement, la corrélation avec les positions oculaires est meilleure avec les STIP qu'avec les SIP. Le maximum local du NSS_{STIP} correspond généralement aux changements brusques du mouvement. En effet, ces changements attirent l'attention alors qu'ils permettent également la génération de STIP. Pour cette catégorie, la contribution de la composante temporelle des points d'intérêt semble être pertinente. Toutefois, ce résultat n'est pas toujours vrai et notamment pour les séquences de football. Ce contre-exemple est probablement dû au fait que les images de football donnent généralement une vision plus large de l'action, qui induit un mouvement apparent plus fluide.

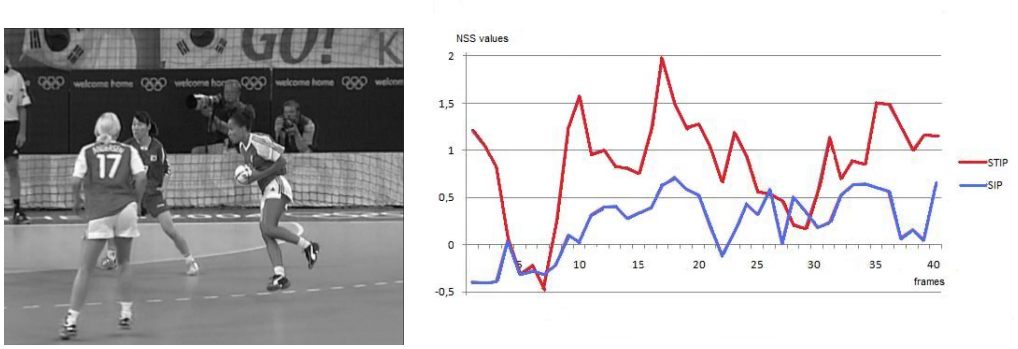


Figure 3.23 — Exemple d'une image et de l'évolution du NSS au cours du temps pour un snippet de la classe "Sports d'équipe".

Visages et mains

La troisième classe est composée de séquences en gros plan des mains et des visages (par exemple, lors d'un concert de musique - image de la figure 3.25). Cette classe représente la situation typique où le NSS_{SIP} est supérieur au NSS_{STIP}.
Dans ces séquences, les zones les plus attractives visuellement sont les visages [Cerf *et al.*, 2007]. De plus, le mouvement dans ces séquences, et particulièrement dans les zones contenant des visages, est généralement très faible. Le NSS reste donc à des valeurs basses. Cependant, ces séquences contiennent de nombreux points d'intérêt spatiaux, généralement situés sur le visage ou les mains qui sont des zones de l'attention visuelle. Ceci explique les bons résultats obtenus. La courbe de la figure 3.25 présente l'évolution du NSS. NSS_{SIP} est

a un niveau plutôt bon (autour de 1.5) et il est presque tout le temps supérieur au NSS_{STIP}.

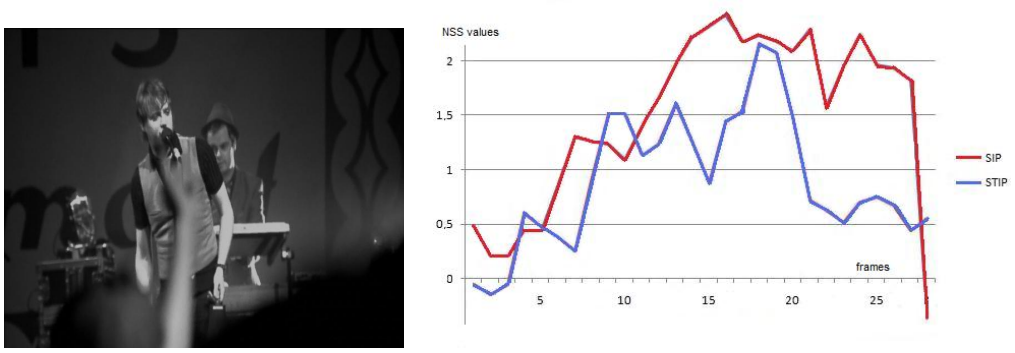


Figure 3.24 — Exemple d'une image et de l'évolution du NSS au cours du temps pour un snippet de la classe "Visages et mains".

Groupes

La dernière classe contient les snippets de manifestation ou de regroupement (image de la Figure 3.25). Cette classe est caractérisée par des mouvements effectués par plusieurs personnes (jusqu'à plusieurs centaines) et ce, avec un taux de couverture de l'image pouvant être proche de 100%.
La courbe de la figure 3.25) montre que cette classe est caractérisée par un NSS assez faible et relativement constant, autour de 0.2, tant pour les SIP que pour les STIP bien que les STIP présentent un NSS légèrement supérieur.
Ces résultats indiquent que les positions oculaires ne correspondent pas aux points d'intérêt. Dans ce type de séquence, les SIP et les STIP sont répartis de manière relativement uniforme dans les images, et les positions oculaires présentent également une distribution aléatoire uniforme car aucune zone de l'image n'est réellement saillante. Il ne peut donc pas y avoir de correspondance entre les points d'intérêt et postions oculaires.

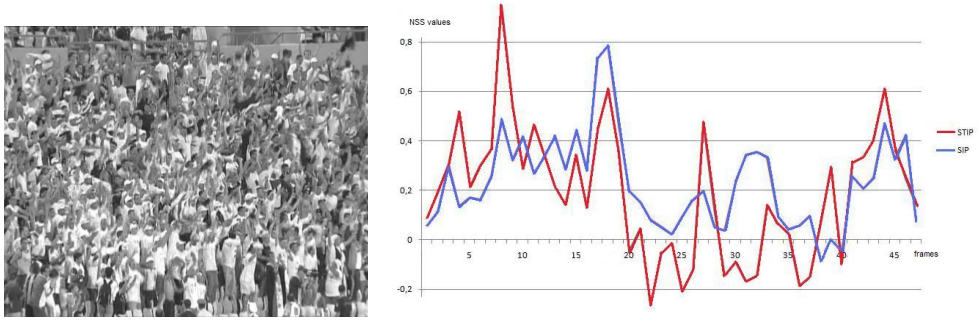


Figure 3.25 — Exemple d'une image et de l'évolution du NSS au cours du temps pour un snippet de la classe "Sports d'équipe".

3.3.5.7 Conclusion

Dans ce paragraphe, nous avons comparé la localisation des points d'intérêt spatiaux et spatio-temporels à la localisation du regard humain. Les points d'intérêt SIP ainsi que leur extension spatio-temporelle STIP fournissent une assez bonne prédiction des positions oculaires des sujets. Ils sont cependant moins performants dans cette tâche que les modèles de saillance, dont c'est le but mais sont beaucoup plus rapide à extraire (détection plus rapide, simplicité de l'algorithme). Ils présentent un compromis performance/rapidité intéressant.

Les cartes d'intérêt M_{SIP} sont en moyenne plus prédictives que les cartes M_{STIP}. Les points d'intérêt spatio-temporels sont un sous-ensemble des points d'intérêt spatiaux, ce qui explique que les cartes M_{SIP} et M_{STIP} aient des NSS moyennes proches et que les évolutions temporelles du NSS pour les deux types de cartes soient également proches.

3.4 Conclusion

Dans ce premier chapitre, nous avons présenté les méthodes fournissant les informations indispensables à l'indexation : les caractéristiques extraites des images (tableau récapitulatif 3.13). Parmi les nombreuses primitives que nous aurions pu utiliser, nous avons choisi d'en utiliser seulement certaines dans le but de répondre à nos besoins d'analyse portés sur le mouvement. Nous avons notamment présenté notre principale source d'informations : les points d'intérêt dans leur version spatio-temporelle avec des applications permettant la segmentation en plans, la segmentation en bloc d'activité, et la détection d'objet en mouvement mais aussi en vérifiant leur saillance par rapport à ce que l'homme perçoit. Cette source d'informations reste néanmoins à un faible niveau sémantique et nous sommes encore loin de pouvoir définir un concept tel que la course uniquement à partir de ces quelques caractéristiques. Il faut maintenant ajouter une étape permettant l'élévation du niveau sémantique nous rapprochant de notre but, la définition d'un concept signifiant.

NOMS	EXTRACTEURS
ligne hough	hough
couleur dominante 1	histogramme couleur TLS
couleur dominante 2	histogramme couleur TLS
type caméra	*saisie utilisateur*
orientation caméra	mouvement dominant
zoom caméra	mouvement dominant
position verticale objet	boite englobante
position horizontale objet	boite englobante
compacité	boite englobante
taille relative	boite englobante
intensité flot optique image	flot optique
intensité flot optique objet	flot optique, boite englobante
orientation flot optique	flot optique
nombre de points par image SIP	SIP
nombre de points par image STIP	STIP
nombre de points par objet cadran 1	STIP, boite englobante
nombre de points par objet cadran 2	STIP, boite englobante
nombre de points par objet cadran 3	STIP, boite englobante
nombre de points par objet cadran 4	STIP, boite englobante
nombre de changements dans le mouvement	STIP (activité)

Tableau 3.13 — Tableau récapitulatif de toutes les caractéristiques.

4 Modèle de briques temporelles

« S'il n'y a pas de solution, c'est qu'il n'y pas de problème. »

Jacques Rouxel

En utilisant comme point de départ des attributs extraits tels que ceux décrits dans le chapitre 2, il s'agit d'indexer une base de données vidéo. Indexer un document vidéo consiste à associer un segment vidéo, c'est à dire une suite d'images, à un ou plusieurs concepts signifiants pour l'utilisateur. Cela signifie que le concept apparaît dans le segment vidéo. Le concept peut concerner l'image complète, ou simplement une partie particulière de l'image, celle-ci étant souvent segmentée en objets. Donc, pour un concept donné, il s'agit de déterminer les séquences (suite d'images) pour lesquelles ce concept apparaît. Nous proposons d'appeler *briques* ces suites d'images du fait de leur représentation 3D (les deux dimensions de l'image et la dimension temporelle).

La segmentation d'objets est classiquement réalisée suivant des approches contour ou région. On peut tenter de prolonger cette distinction dans le contexte spatio-temporel. Dans l'approche contour, on cherche des variations ou des points saillants spatio-temporels. Ceux-ci indiquent une rupture dans une séquence (typiquement le changement de plan). Par exemple les travaux de [Dollar *et al.*, 2005; Laptev *et al.*, 2007a; Klaser *et al.*, 2008] utilisent une approche contour. Dans l'approche région, on cherche en revanche une certaine cohérence spatio-temporelle dans le segment, donc une propriété vérifiée tout au long du segment. Les travaux de [Gambotto, 1989] sont un exemple d'approche régions.

Dans le cadre de nos travaux, nous avons opté pour une segmentation par approche région sur laquelle on effectue un suivi temporel. Pour cela, nous utilisons les attributs extraits des séquences d'images afin d'extraire des *briques spatio-temporelles* : elles représentent une suite d'images consécutives dans le plan et correspondent à la validité de certaines propriétés en terme d'attributs.

4.1 Structure générale

La structure que nous proposons repose donc sur l'extraction de briques, définies comme une séquence d'images du document vidéo ne présentant pas de transitions, c'est à dire des plans (voir positionnement - chapitre 2). Quand le concept est signifiant pour l'utilisateur, il correspond souvent à un modèle mettant en jeu plusieurs attributs parfois combinés de manière évoluée. Nous proposons donc une structure hiérarchique à deux niveaux, un premier niveau où l'on extrait des briques dites de base, puis un deuxième niveau où l'on combine ces briques de base afin d'extraire les briques correspondant au concept (figure 4.1).

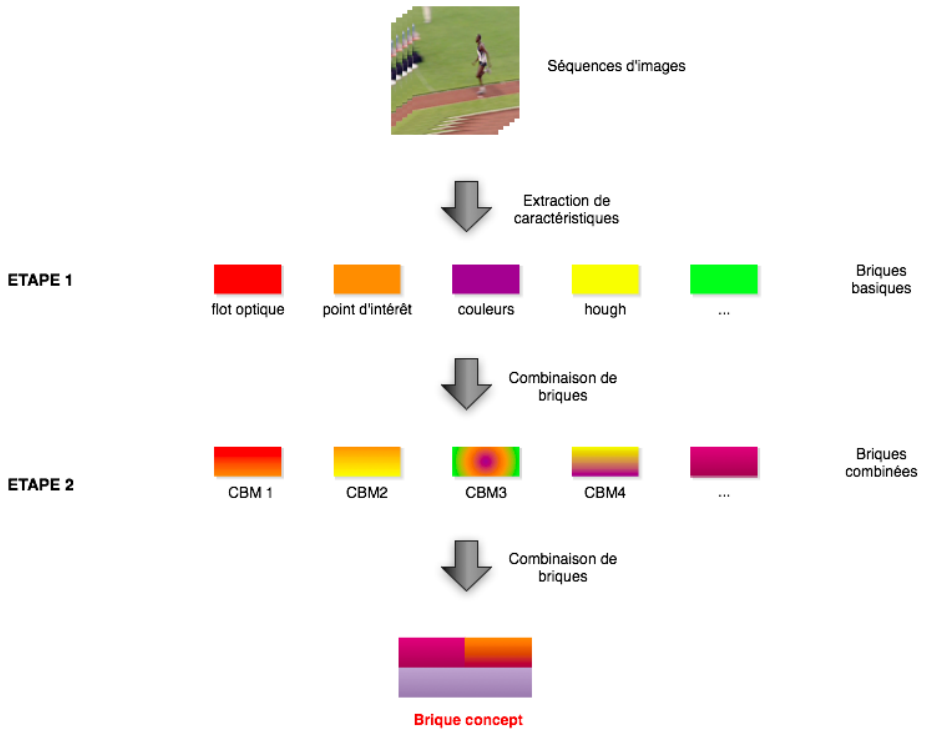


Figure 4.1 — Les deux niveaux de briques : les briques de base de l'étape 1 (Basic Block Model - BBM) correspondant directement aux caractéristiques extraites et les briques combinées de l'étape 2 (Combined Block Model - CBM) correspondant aux combinaisons de briques pour arriver à la brique concept final.

Premier niveau Les briques de base sont directement définies à partir des caractéristiques extraites du document vidéo : dans une brique de base, une caractéristique particulière vérifie une propriété particulière. Ainsi une brique de base sera caractérisée par un couple caractéristique/propriété, ce couplage pouvant prendre différentes formes.
Par exemple, la brique 'type de caméra' peut prendre deux valeurs : fixe ou mobile ; la brique 'compacité' peut prendre toutes valeurs comprises entre 0 et 1. Chaque couple définit donc une brique de base et une séquence vidéo peut être segmentée avec les briques de base ainsi définies.

Deuxième niveau Le concept est défini à partir d'une combinaison de caractéristiques véri-

fiant un certain nombre de propriétés. On définit les briques combinées correspondant à ces combinaisons de caractéristiques jusqu'à la brique concept.

La structure en briques nécessite 2 étapes :

une première étape de modélisation consistant à définir les modèles de représentation des briques de base et des briques combinées c'est à dire la manière de les définir, de les construire, de les représenter, de les assembler et enfin de les utiliser,

une seconde étape d'extraction consistant à extraire les briques de séquences d'images. Cette extraction est réalisée en utilisant les caractéristiques extraites de chaque séquence (voir chapitre 3) et des modèles de briques définis.

Dans un premier temps, nous définissons les modèles de briques de base puis les modèles de briques combinées, ainsi que les outils permettant de les définir, comme les opérateurs de liaison. Nous présentons ensuite l'implémentation sous forme de base de données qui correspond à l'architecture informatique que nous avons choisie. Enfin, nous expliquerons la méthode d'amélioration de la qualité des briques après segmentation basée sur des techniques de filtrage.

4.2 Définition du modèle de briques de base

Les briques de base constituent les éléments sémantiques constitutifs de premier niveau. Un modèle de briques de base correspond à la définition d'un type de brique. Deux éléments le composent :

- une caractéristique extraite de la séquence d'images comme le nombre d'objets en mouvement ou un attribut bas-niveau construit à partir d'une caractéristique extraite comme la vitesse et l'orientation de la caméra (construit à partir du flot optique) ;
- une propriété de validation sur la caractéristique qui peut être par exemple une valeur ou un ensemble de définition ;

4.2.1 Caractéristiques

La première information permettant l'élaboration des modèles de brique de base est la caractéristique considérée. Nous avons vu dans le chapitre précédent un certain nombre d'extracteurs de primitives. A partir de ceux-ci, nous avons construit des attributs de plus haut niveau sémantique : les caractéristiques images. Un expert développe les algorithmes qui réalise les opérations de traitement d'images qui permettent la réalisation de cette étape. Ce sont ces caractéristiques que nous allons utiliser pour définir les modèles de briques. La liste des caractéristiques complète et détaillée est donnée dans l'annexe A.

Ces caractéristiques sont stockées sous forme de base de données (figure 4.2) dans plusieurs tables. La table 'sequence' stocke les informations et les caractéristiques correspondantes relatives aux séquences. La table 'image' stocke les caractéristiques relatives aux images et la table 'objet' stocke les caractéristiques relatives aux objets en mouvement présents dans l'image. La table 'image' est reliée à la table 'sequence' (on trouve un certain

nombre d'images dans une séquence) et la table 'objet' est reliée à la table 'image' (on trouve un certain nombre d'objet dans une image - de 1 à plusieurs).

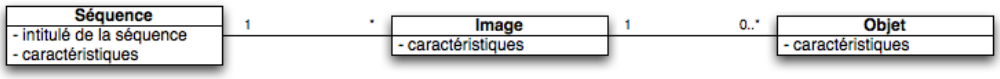


Figure 4.2 — Stockage des caractéristiques extraites (le détail des caractéristiques est donné dans l'annexe A).

4.2.2 Propriétés sur les caractéristiques

La deuxième information correspondant à un modèle de brique est une propriété sur la caractéristique considérée : elle peut se définir comme un sous-ensemble de l'ensemble de définition et elle permet d'effectuer la segmentation de l'ensemble de définition de la caractéristique en plusieurs parties. On dispose de deux types de propriétés en fonction de la nature de l'ensemble de définition initial de la caractéristique : un de type discret et un de type continu. Pour un certain nombre de caractéristiques, les valeurs pouvant être prises sont discrètes et peu nombreuses comme par exemple : 0 ou 1, vrai ou faux, présent ou absent, horizontal, vertical ou oblique, etc. Par conséquent, l'ensemble de définition associé à une brique est l'une de ces valeurs. Pour les autres caractéristiques, s'agissant d'ensemble continu, l'ensemble de définition associé à une brique est un sous-ensemble de l'ensemble de définition de la caractéristique. Afin de ne pas créer de cas *sans solution*, on fait le choix de définir les propriétés pour que l'union de toutes les propriétés de définition des briques d'une caractéristique donnée soit égale à l'ensemble de définition de la caractéristique considérée. Par exemple, la compacité est définie sur l'intervalle [0 ;1] et est divisée en 3 propriétés qui ont pour intervalles de définition : [0 ;0,45], [0,45 ;0,6] et [0,6 ;1]. L'union de ces trois intervalles recouvrent bien l'intervalle de définition de la compacité.

4.2.2.1 Discrétisation des espaces de définition continus

Si le nombre d'intervalles est insuffisant, le caractère de discrimination des briques est altéré et si le nombre est trop important, les briques ne seront pas réellement utilisables et les définitions, trop précises ou spécifiques, ne permettront pas d'obtenir des résultats satisfaisants. C'est le problème de la discrétisation des espaces de définition continues.

Dans le but de définir les différents intervalles de définition en utilisant le moins d'expertise possible, nous avons commencé par effectuer des découpages en parts égales, généralement en trois parties. Néanmoins, il est apparu que le nombre d'intervalles et leurs tailles avaient de l'importance quant à la capacité discriminatoire de chaque brique définie (certaines n'étaient jamais utilisées, d'autres l'étaient à 90%) pour une application donnée. Cette étape est réalisée par un expert en traitement d'images qui n'est pas spécialiste de l'application. Finalement, après extraction des caractéristiques, nous disposons d'une connaissance statistique des valeurs. Il est donc envisageable d'optimiser la répartition des intervalles dans l'ensemble de définition de chaque caractéristique. Nous avons utilisé une base hétérogène composée de 1000 séquences issues de l'ensemble des bases disponibles

afin de garantir la diversité des contenus. Par exemple, la figure 4.3 montre la répartition des valeurs de compacité comprises entre 0 et 1. On voit sur l'exemple que la répartition

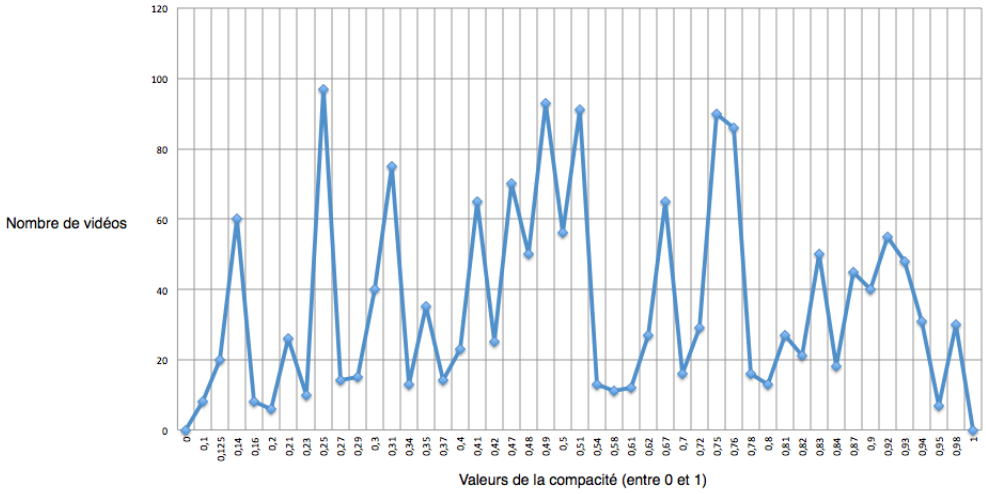


Figure 4.3 — Exemple de répartition des valeurs trouvés pour la compacité dans une base hétérogène.

n'est pas homogène et n'est pas exhaustive. Certaines valeurs n'apparaissent jamais comme les faibles valeurs de compacité, alors que d'autres apparaissent plus souvent que d'autres comme par exemple les valeurs 0.25 et 0.5 de compacité. Il est donc préférable de prendre en compte cette répartition pour définir les différents intervalles de valeurs de briques. Le tableau 4.1 montre la répartition en deux groupes égaux, en trois groupes égaux, en quatre et en cinq groupes égaux sur l'exemple de compacité dont la répartition a été présentée dans 4.3. On voit que la répartition ne se fait pas en intervalles égaux. Pour chaque attribut, nous avons fait cette étude statistique afin de déterminer les intervalles permettant la meilleure répartition.

Le nombre d'intervalles à définir est assez difficile à choisir. Il est bien évident que si le nombre n'est pas suffisant, même si on optimise la répartition, les briques ne seront pas réellement utilisables. Au regard des résultats initiaux obtenus sur les intervalles à parts égales, un minimum semble être 3. Plus on augmentera la quantité et plus on aura de briques et donc plus il sera difficile d'effectuer le choix d'une brique ou d'une autre lors de la phase de définition d'un concept. En fait, la quantité d'intervalles a un impact direct sur la finesse de la définition réalisable lors de la spécification d'un concept mais cela entraîne également l'augmentation des difficultés de détection.

Dans l'exemple de la compacité, nous avons choisi d'utiliser trois intervalles. La répartition équitable en quantités étant 40/25/35 sur un intervalle de définition allant de 0 à 1 inclus, les intervalles définies sont donc [0 ;0,45[, [0,45 ;0,6[et [0,6 ;1].

Répartition des vidéos souhaitée	50%	50%
Répartition de la valeur à appliquer (de l'intervalle de définition)	57%	43%

Répartition des vidéos souhaitée	33%	33%	33%
Répartition de la valeur à appliquer (de l'intervalle de définition)	40%	25%	35%

Répartition des vidéos souhaitée	25%	25%	25%	25%
Répartition de la valeur à appliquer (de l'intervalle de définition)	35%	22%	10%	33%

Répartition des vidéos souhaitée	20%	20%	20%	20%	20%
Répartition de la valeur à appliquer (de l'intervalle de définition)	31%	13%	18%	17%	20%

Tableau 4.1 **—** Différentes répartitions des valeurs (pour 2/3/4/5 groupes) pour la caractéristique compacité.

4.2.2.2 Le problème des intervalles ouverts

Certains espaces de définition sont des intervalles ouverts et posent le problème de leur découpage. Par exemple, le nombre de points d'intérêt varie entre 0 et l'infini (bien qu'il soit limité par le nombre de points de l'image, si l'image était de taille infinie...). Dans le cas des intervalles ouverts, on a utilisé les valeurs trouvées pour établir la répartition mais le dernier intervalle qui doit prendre en compte la notion d'infini est défini comme ouvert. Par exemple, l'attribut 'nombre de lignes' est réparti sur 3 intervalles : [0,10], [10,100] et [100+] c'est à dire un nombre allant de 100 à l'infini.

4.2.2.3 Base de données applicatives

Le résultat final de cet assemblage caractéristique/propriété permet la définition d'un certain nombre de modèles de briques. Pour chacune des 18 caractéristiques définies, un certain nombre de modèles de briques permettent de segmenter l'espace des caractéristiques. Au final, nous obtenons un ensemble de 60 modèles de briques dont la liste est dans l'annexe B.

4.2.2.4 Remarques

Cette première définition des intervalles permet de définir chaque modèle de brique. Néanmoins rien ne garantit qu'il s'agisse des meilleurs choix. Par exemple, si dans la définition d'un concept, la valeur oscille autour d'un changement d'intervalle, cela pourrait diminuer la qualité de la définition. Pour cela, deux solutions peuvent être envisagées. La première consiste à utiliser un intervalle de plus soit en passant au niveau de découpage su-

périeur toujours en garantissant la répartition, soit en découpant les deux intervalles en trois parties égales (figure 4.4.a) ou en trois parties réparties (figure 4.4.b). La seconde possibilité est de faire évoluer les bornes des intervalles définis en utilisant des méthodes d'apprentissage.

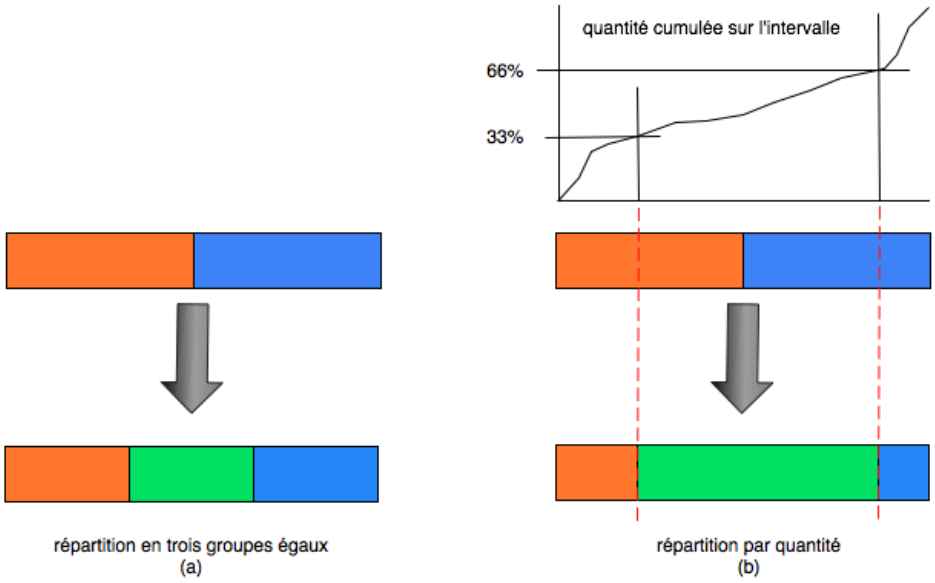


Figure 4.4 — Méthode d'adaptation de la répartition initiale : a) en parties égales et b) par répartition quantitative.

Le type de représentation est également discutable. En effet, les problèmes aux bornes des différents ensembles de définition des briques pourraient être plus ou moins dissipés par l'utilisation d'intervalles flous (figure 4.5).

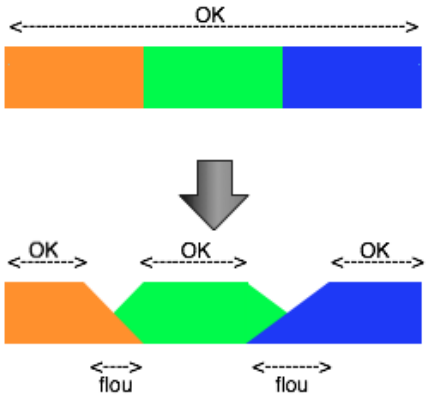


Figure 4.5 — Passage d'une définition en intervalles nets à une définition en intervalles flous.

Enfin, le nombre de caractéristiques disponibles est un facteur important quant aux capacités finales du système pour la définition de concepts précis. Alors que dans la plupart

des systèmes actuels on tend à l'augmentation du nombre de caractéristiques, nous nous sommes limités dans un premier temps à une quantité de caractéristiques suffisantes pour un certain nombre de concepts mais qui ne permettra pas d'être discriminante quand nous aurons des centaines de concepts à définir surtout si leur définition est proche. Il faudra trouver un élément permettant de les différencier et parmi ceux disponibles, il n'y en aura peut être aucun.

4.3 Définition des modèles de briques combinées

Nous disposons d'un ensemble de modèles de brique de base liées directement aux caractéristiques extraites. Mais en général, un concept ne peut être défini uniquement sur la propriété d'une seule caractéristique. C'est la combinaison de plusieurs propriétés sur différentes caractéristiques qui va permettre de définir un concept.
Pour répondre à ce besoin, nous avons défini les modèles de briques combinées. Il s'agit de combiner des modèles de briques en utilisant des opérateurs temporels complétés par des opérateurs logiques.

On peut définir un modèle de briques combinées comme étant une combinaison d'au moins deux modèles de briques (de base ou combinée) et l'utilisation d'au moins une relation temporelle permettant de lier les briques entre elles.

Définition : Un concept est défini comme un modèle de brique combinée c'est à dire par une combinaison d'un certains nombre de briques basiques et/ou combinées et l'utilisation de relations logiques et temporelles liant ces briques entre elles.

4.3.1 L'algèbre temporelle

On définit une brique combinée comme une combinaison de briques de base et/ou combinées. La combinaison de modèle de briques peut être logique ou temporelle. La combinaison logique correspond à l'existence simultanée de deux briques de base alors que la combinaison temporelle est la mise en séquentialité de deux briques de base. L'aspect temporel joue un rôle important dans la définition de brique combinée. Nous nous sommes donc particulièrement intéressés à cet aspect.
La première difficulté rencontrée est la représentation et la modélisation des trois éléments clés pour l'analyse temporelle :

- date des événements (position chronologique) ;
- durée des événements ;
- délais entre événements.

Mais les relations entre les entités temporelles sont difficiles à établir la plupart du temps. Il est nécessaire de faire la distinction entre les différents types d'entités temporelles mises en relation (points, intervalles) ainsi qu'entre les propriétés des domaines temporels sous-jacents (ordonnancement, métrique, discret/continu).

Différents modèles ont été développés pour modéliser le temps. Ils se déclinent en différents aspects : temps ponctuel ou sous forme d'intervalle, temps totalement ou partiellement ordoné (temps linéaire ou branché), temps discret ou dense, borné ou non, incluant différentes granularités ou non [LeBer *et al.*, 2007; Allen, 1983].
Parmi ces représentations du temps, on peut en citer deux principales :

- l'algèbre de points de Vilain et Kautz [Vilain et Kautz, 1986] qui propose trois relations de base : précède ($<$), identique ($=$) et suit ($>$) ;
- l'algèbre d'intervalles de Allen [Allen, 1981, 1983] qui propose 13 relations : b (before), m (meets), o (overlaps), s (starts), d (during), f (finishes), eq (equals) et leurs relations inverses bi, mi, oi, si, di, fi que l'on peut respectivement traduire par b (avant), m (fait démarrer), o (chevauche), s (démarre avec), d (est pendant), f (fini avec), eq (est égal à). La durée n'est pas représentée. La figure 4.6 présente les 13 relations en les illustrant sur deux concepts X et Y.

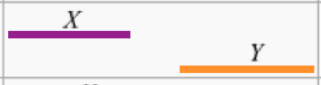
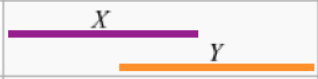
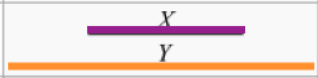
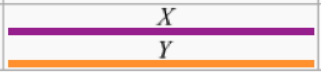
Relation	Illustration	Interpretation
X<Y Y>X		X se déroule avant Y
XmY YmiX		X fait démarrer Y (i représente l'inverse)
XoY YoiX		X chevauche Y
XsY YsiX		X démarre avec Y
XdY YdiX		X est pendant Y
XfY YfiX		X finit avec Y
X=Y		X est égal à Y

Figure 4.6 — Les 13 relations de l'algèbre des intervalles d'Allen.

Ce formalisme permet de représenter de manière simple les relations temporelles entre les actions. Voici par l'exemple donné dans [Ber *et al.*, 2009], en prenant la phrase "Pendant le repas, Peter lit le journal. Ensuite il va se coucher" se traduit par :

$$\text{journal}\{\text{d}, \text{s}, \text{f}\}\text{diner} : \textbf{journal} \text{ démarre, dure et termine } \textbf{diner} \tag{4.1}$$

$$\text{diner}\{<, \text{m}\}\text{se coucher} : \textbf{diner} \text{ précède et fait démarrer se } \textbf{coucher} \tag{4.2}$$

L'ensemble de ces relations et des tables de composition ont été beaucoup employées dans l'analyse syntaxique et le traitement automatique des documents écrits. Nous nous proposons d'utiliser cet algèbre en utilisant les briques comme des mots afin d'établir des combinaisons cohérentes de briques qui pourront être assimilées à des *phrases* syntaxiquement correctes.

4.3.2 Prise en compte de la durée

L'algèbre de Allen ne proposant aucun moyen de représenter la durée, elle a été complétée en introduisant une notion de durée des intervalles pour construire le modèle *INDU* pour "*interval and duration*" [Pujari *et al.*, 1999]. Ainsi, ce modèle est composé des 13 relations de Allen auxquelles on ajoute 3 relations permettant de comparer la durée de deux intervalles X et Y : X est de durée inférieure à Y ($X<Y$), supérieure ($X>Y$) ou égale ($X=Y$). On obtient finalement 25 relations (certaines combinaisons étant impossibles).

4.3.3 Opérateurs logiques et complémentaires

Certains cas particuliers ne sont pas définissables par l'utilisation du modèle INDU. Par exemple, il n'est pas possible de gérer l'impossibilité d'effectuer un choix entre deux briques différentes pour un même intervalle. Pour cela, nous avons complété l'algèbre des intervalles d'Allen, les règles de composition ainsi que le modèle INDU en ajoutant l'opérateur **OU exclusif**.

De plus, la séquentialité c'est à dire la répétition d'une séquence de briques ne peut être modélisée à partir de tous ces éléments. Pour terminer la liste des opérateurs, nous avons donc ajouté l'opérateur * indiquant la répétition d'une séquence.

Enfin, il est également nécessaire de définir un opérateur permettant d'exprimer un choix de manière quantitative comme ***un parmi*** qui peut également être vu comme un OU dont on connaît le nombre de choix à prendre.

Les notations pour ces opérateurs sont :

$$OU\ exclusif : \oplus \tag{4.3}$$

$$repetition : \{X\}^* \tag{4.4}$$

$$choix\ multiple(n\ parmi) : \{fo-*\}^n ou\ bien \{fo-o; fo-r; ...\}^n \tag{4.5}$$

4.3.4 Exemple de la définition de la course à pied

On peut définir la course à pied (humaine) comme un déplacement linéaire par rapport au sol effectué en utilisant la partie postérieure du corps, les jambes qui alternent rapidement au cours du temps. En utilisant les 60 modèles de briques basiques disponibles (la liste est disponible en annexe B), on peut définir la course comme une variation de compacité de l'objet en mouvement - ce qui traduit l'alternance rapide des jambes, et comme une continuité dans l'orientation et l'intensité du flot optique.
Les briques utilisées sont donc celles concernant la compacité, le nombre d'objets en mouvement, l'intensité du flot optique et l'orientation du flot optique.
La liste des briques est donc la suivante :

- compacité faible/moyen/fort (c-f, c-m, c-F)
- le nombre d'objet (no-1)
- intensité du flot optique de l'objet (fo-io-moyen)
- orientation du flot optique (fo-$*$)

Comme on l'a vu précédemment dans 4.3.1, on dispose d'opérateurs de liaison. D'après la définition effectuée, on a besoin de l'opérateur permettant d'indiquer la succession de briques, la simultanéité et l'inclusion de deux briques, la répétition et les indicateurs sur la durée des briques.
La liste des opérateurs est donc la suivante :

- succession de briques : l'opérateur m. XmY indique "X fait démarrer Y" c'est à dire X et Y s'enchaîne.
- l'inclusion de briques : l'opérateur d. XdY indique "X est pendant Y".
- la répétition : l'opérateur *.
- les indicateurs sur la durée : < et >. X<Y indique "X est plus court que Y".

En utilisant les briques et opérateurs indiqués, une définition de la course à pied est :

$$(c-f\ \mathbf{m}\ c-m\ \mathbf{m}\ c-F\ \mathbf{m}\ c-m)^*\ \mathbf{d}\ \{no-1\ \mathbf{d} fo-io-moyen\}\ \mathbf{d}\ \{fo-*\}^1 \tag{4.6}$$

avec

$$(c-f) > c-m\ et\ (c-F) > c-m \tag{4.7}$$

Au final, on obtient une définition du concept 'course à pied'. C'est définition n'est pas unique. Une autre définition, utilisant des briques différentes serait tout à fait envisageable. Le principal est que la définition choisie permette la récupération du concept recherché.

4.4 Structuration des données

L'ensemble des briques basiques (caractéristiques et propriété associée) et des briques combinées (combinaison de briques basiques et/ou combinées à l'aide de relations temporelles) constitue un ensemble de données qu'il est nécessaire de structurer. La structuration doit permettre de conserver leurs relations tout en permettant un certain nombre d'opérations sur ces données. Le choix des structures de représentation des données est stratégique car le système doit répondre à plusieurs critères :

- facilité et rapidité pour l'insertion, la récupération de données ;
- permettre la mise à jour de l'architecture et la maintenance des données et des structures ;
- permettre la mise en place d'algorithme pour le parcours et la manipulation des données ;
- garantir la pérennité dans le temps des données.

Nous avons choisi de construire le système autour des bases de données permettant de gérer facilement l'insertion et la récupération de données mais également minimiser le temps des nombreuses entrées/sorties (E/S), de faciliter la mise à jour des tables et des données tout en garantissant la conservation des données et des relations entre les données. Par contre, la souplesse de manipulation est relativement limitée et les algorithmes de manipulation sont parfois un peu difficiles à construire.

4.4.1 Organisation des données

Les données ont été organisées dans une base de données. Les connaissances de base sur les bases de données (modèle Entité-Association, modèle relationnel, représentation et ma-

nipulation) sont accessibles dans le livre [Audibert, 2009]. Les tables concernant le stockage des modèles sont au nombre de trois : une pour les modèles de brique basique (BBM - Basic Block Model), une pour les modèles de brique combinée (CBM - Concept/Combined Block Model) et une pour les opérateurs. La figure 4.7 illustre la représentation de ces trois tables.

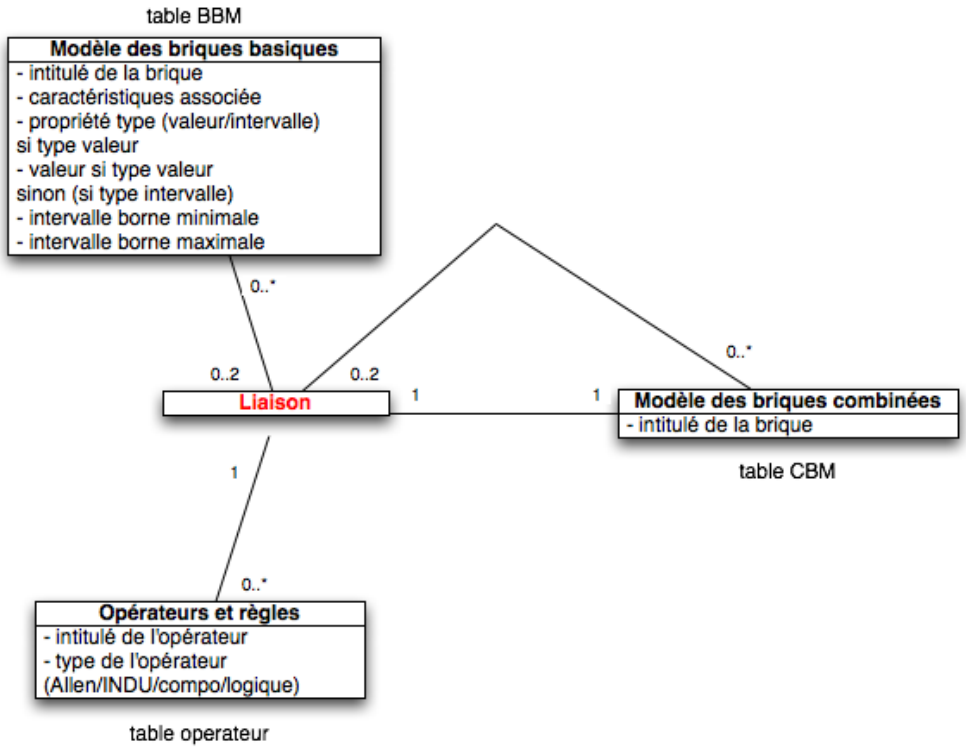


Figure 4.7 — Modèle de stockage des modèles de briques et d'opérateurs de combinaison/liaison.

En analysant le schéma relationnel 4.7 avec les règles expliquées dans [Audibert, 2009] on distingue 3 tables :

- la table 'BBM' contient les modèles de briques basiques. Chaque entité est composée plusieurs attributs : l'intitulé, la référence de la caractéristique associée, le type de propriété pour la définition ainsi que la ou les valeurs de la propriété. Le 'type de propriété' est soit 'valeur' soit 'intervalle'. Dans le cas du type valeur, on a un attribut 'valeur' qui est renseigné et qui correspond à la valeur de la propriété de la définition de la brique. Dans le cas du type intervalle, les attributs 'borne minimale' et/ou 'borne maximale' sont renseignés et correspondent aux bornes de l'intervalle de la propriété définie.
- la table 'CBM' contient les modèles de briques combinées. Chaque entité est composée de plusieurs attributs : l'intitulé et via les liaisons, les briques basiques ou combinées utilisées dans la définition ainsi que l'opérateur utilisé.
- la table 'opérateur' contient les opérateurs logiques et temporels permettant la liaison des briques. Chaque entité est composée de deux attributs : l'intitulé de l'opérateur et le type (temporelle-Allen / INDU / logique / composition).

On dispose d'une relation 'liaison' qui permet la définition de la brique combinée comme étant une combinaison de briques basiques ou combinées et d'opérateurs. On note en observant les cardinalités choisies que :

- une brique combinée est une liaison qui est constituée de deux briques (0 à 2 basiques ou 0 à 2 combinées) et d'un opérateur (tels que définis au paragraphe 4.3.1)
- une brique basique ou combinée peut ne jamais avoir été utilisée lors de définition (aucune définition n'utilise cette brique)
- un opérateur peut ne jamais avoir été utilisé lors de définition (aucune définition n'utiliser cet opérateur)

4.4.2 Évolution des données de ces tables

On peut différencier deux types de données dans le système. D'un coté, on a les données définies par l'Expert Applicatif (qui sont des données hors-lignes) comme les définitions des briques de base qui peuvent évoluer lors de l'ajout de caractéristiques extraites par exemple. De l'autre, on a les données liées au système qui évolue lors de la définition d'un concept et du fonctionnement du processus.

Évolution des données hors-ligne

La table des opérateurs n'est pas destinée à évoluer dans le système. On peut envisager d'enrichir les opérateurs proposés, de les supprimer s'ils sont inutiles voire de les remplacer par un autre système d'opérateurs. Mais il conviendra d'évaluer l'impact d'une telle modification car il sera sans doute nécessaire d'éliminer tout ou partie des briques combinées définies et de recommencer la définition des concepts qui avaient été insérés.

La table des briques basiques est sans aucun doute la plus intéressante. Dans un premier temps, elle n'évoluera pas. Ultérieurement, deux évolutions sont envisagées. La première, à l'instar de celle des opérateurs, serait d'enrichir cette table. Lors de l'insertion d'un nouvel attribut dans le système, il faudra ajouter les modèles de brique basique associés. L'augmentation ou la diminution d'un nombre de modèles de briques par attribut modifiera également cette table. Enfin, les définitions des intervalles de chaque modèle pourraient également être réévaluées. La seconde évolution envisagée, et de loin la plus intéressante, est l'insertion d'un bouclage de pertinence (relevance feedback) afin d'améliorer les modèles des briques. Nous verrons dans les perspectives les propositions que nous faisons sur ce point particulier.

Évolution des données d'usage

La définition d'un nouveau concept consiste à créer une ou plusieurs briques combinées (au moins le concept défini). Ces nouvelles briques sont insérées à chaque nouvelle définition dans la table des modèles de briques combinées.

4.5 Extraction des briques

4.5.1 Extraction des briques basiques

La première étape du processus est l'extraction des briques basiques. A ce stade, nous disposons des données issues de la phase préliminaire d'extraction de caractéristiques. Il s'agit de créer une liste de briques basiques classées par séquences vidéo, puis par images et enfin par objet en mouvement. Selon la caractéristique considérée, elles sont stockées dans la table leur correspondant. Par exemple, la compacité, une caractéristique concernant un objet, est stockée dans la table 'objet' mais on garde également un lien sur l'image dans laquelle l'objet a été trouvé et sur la séquence dans lequel l'objet apparaît. L'agencement de ces tables est donné dans la figure 4.2.

4.5.1.1 Processus général

Le processus général permettant de passer d'une base de caractéristiques à une base de briques basiques est composé de deux étapes représentées sur la figure 4.8.

La première étape est une phase de récupération de données et de mise en forme qui passe par la création automatique de requêtes. Cette phase découle directement des choix qui ont été effectués au niveau des structures choisies de représentation des données. En ayant choisi d'utiliser une base de données, cette phase consiste à construire des requêtes de récupération de données et à mettre en forme les données récupérées pour les exploiter dans les traitements.

La seconde permet de filtrer ces données. Cette phase consiste à *débruiter* les données initialement extraites afin de construire des briques plus cohérentes. Ces données sont ensuite stockées dans la base de données.

4.5.1.2 Extraction des briques

L'extraction des informations contenues dans les séquences d'images, c'est à dire des caractéristiques extraites, dans le but de construire les briques à partir des modèles, est une étape qui nécessite d'utiliser la table des modèles de briques basiques ainsi que les trois tables (4.2 - tables séquence/image/objet) contenant les données extraites. Il s'agit d'extraire pour chaque modèle de brique, sur une séquence d'images donnée, toutes les images concernées.

On dispose de trois catégories d'informations stockées : celles concernant les séquences, celles concernant les images et celles concernant les objets. De plus, on a deux types de propriétés : définies par intervalles et définies par valeurs. Il faut donc prendre en compte le type de propriété et construire la requête en utilisant les définitions présentes.

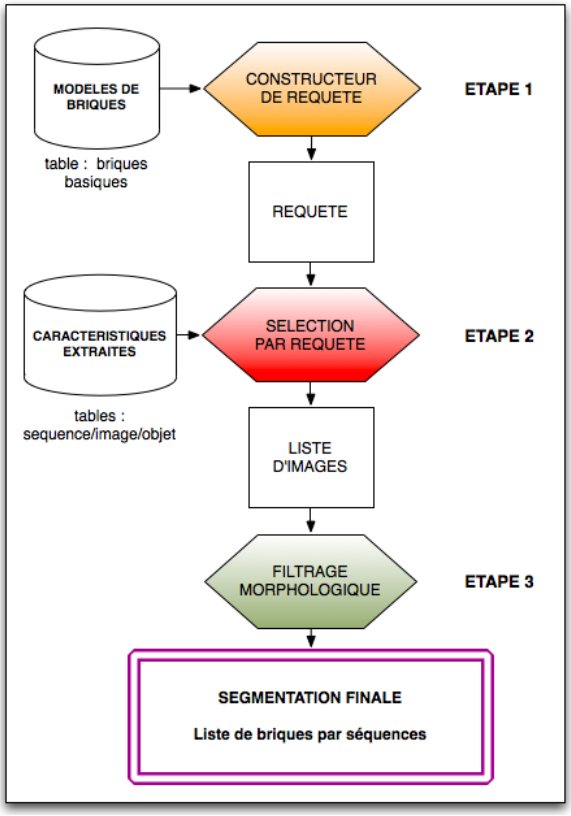


Figure 4.8 — Extraction des briques basiques : à partir d'une base de caractéristiques extraites et des modèles de brique basique d'obtenir une base de briques basiques filtrées.

La difficulté réside dans le fait que tous les modèles ne concernent pas systématiquement toutes les mêmes données. En effet, on dispose de caractéristiques basées séquence, de caractéristiques basées image et de caractéristiques basées objet. Il faut donc faire passer les caractéristiques basées séquence et les caractéristiques basées objet en caractéristiques basées image. Afin de ne pas dupliquer les données et de ne pas perdre l'avantage d'une structuration en base de données, il a donc fallu établir des algorithmes de sélection adaptatifs. L'idée est de construire un générateur de requêtes (algorithme 4.5.1.2) qui, en fonction des données à récupérer, cible les tables à utiliser et construit les liens nécessaires (jointures entre les tables notamment).

L'algorithme 4.5.1.2 permettant la génération de requêtes fonctionne en deux étapes. Il commence par déterminer la table à utiliser : soit 'sequence', soit 'image' soit 'objet'. S'il doit utiliser 'sequence', il sélectionne la table 'sequence'. S'il doit utiliser image, il sélectionne la table 'sequence', la table 'image' et effectue la jointure. S'il doit utiliser objet, il sélectionne la table 'sequence', la table 'image', la table 'objet' et effectue la jointure entre les tables 'sequence' et 'image' et entre les tables 'image' et 'objet'. La seconde étape consiste à regarder le type de propriété de la brique à utiliser. Si la propriété est de type valeur, il effectue une

sélection de type égalité ("paramètre=valeur") sinon, la propriété est de type intervalle, il effectue une sélection par encadrement (paramètre>=borne inférieure et paramètre<borne supérieure) avec la gestion du cas particulier des intervalles ouverts.

```
requete = ""
requete << "SELECT "
SI (concerne une donnee de la table 'sequence')
   SI (on a le numero de la sequence)
  requete << "numero "
   SINON
     requete << dernieresequence(numero)
   FINSI
   requete << "FROM sequence WHERE "
SINON SI (concerne une donnee de la table 'image')
 SI (on a le numero de la sequence)
  requete << dernieresequence(numero)
  SINON
     requete << "sequence.numero,image.indice,frame"
     SI (concerne une donnee de la table 'objet')
       requete << ",reference"
     FINSI
  FINSI
  requete << "WHERE sequence.numero=image.numero AND "
  SI (concerne une donnee de la table 'objet')
    requete << "image.indice=objet.indice AND "
  FINSI
FINSI
SI (la donnee est sur un intervalle)
 SI (si il a une borne inferieure)
     requete << "parametre>=borneinferieure "
     SI (si il a une borne superieure)
       requete << "AND "
     FINSI
  FINSI
  SI (si il a une borne superieure)
  requete << "parametre<bornesuperieure "
  FINSI
SINON
 requete << "parametre=valeur"
FINSI
RETOURNE requete
```

Figure 4.9 — Algorithme de génération de requêtes

La figure 4.10 présente trois exemples de requêtes générées en utilisant les trois tables (sequence, image et objet) contenant les caractéristiques extraites (figure 4.2).

La première (requête figure 4.10 listing 4.1) permet la sélection (SELECT) de tous

```
SELECT numero INTO sequence WHERE sequence.camera='fixe'
```

Listing 4.1 — requête utilisant uniquement la table 'sequence'

```
SELECT numero INTO sequence,image WHERE sequence.numero=image.sequence
    AND image.stip<100
```

Listing 4.2 — requête utilisant la table 'sequence' et la table 'image'

```
SELECT numero INTO sequence,image,objet WHERE sequence.numero=image.
    sequence AND image.numero=objet.image AND objet.compacity<0.4
```

Listing 4.3 — requête utilisant la table 'sequence' la table 'image' et la table 'objet'

Figure 4.10 — Quelques exemples de requêtes générées

les numéros des séquences (numero INTO sequence) dont la caméra est fixe (WHERE sequence.camera='fixe').

La seconde (requête figure 4.10 listing 4.2) permet la sélection (SELECT) de tous les numéros des séquences (numero INTO sequence) dont les images présentent moins de 100 STIP (WHERE image.stip<100). Pour cette requête, il est nécessaire d'effectuer une jointure entre les tables 'sequence' et 'image' (WHERE sequence.numero=imagesequence).

Enfin, la troisième requête (requête lfigure 4.10 listing 4.3) permet la sélection de tous les numéros des séquences présentant des images dont les objets en mouvement ont une compacité inférieure à 0.4 (objet.compacity<0.4). Dans cette requête, il est nécessaire de faire la jointure entre les tables 'sequence' et 'image' (sequence.numero=image.sequence) ainsi qu'entre les tables 'image' et 'objet' (image.numero=objet.image).

4.5.2 Filtrage des données et stockage des informations

Certaines des briques extraites ont été obtenues image par image. D'autres durent plusieurs dizaines d'images mais présentent quelques *trous* d'une ou deux images. Ainsi, il est nécessaire de réaliser une étape de filtrage de ces données triées (figure 4.11). On rappelle que les briques sont triées par séquence puis par image et enfin par objet. Par exemple, on a d'abord toutes les briques issues de la séquence A. Parmi celles-ci, elles sont triées par image donc les premières sont celles de l'image 1 (de la séquence A). Enfin, elles sont triées par numéro d'objet.

Le but de l'étape de filtrage est par conséquent :
– éliminer les briques trop petites ;

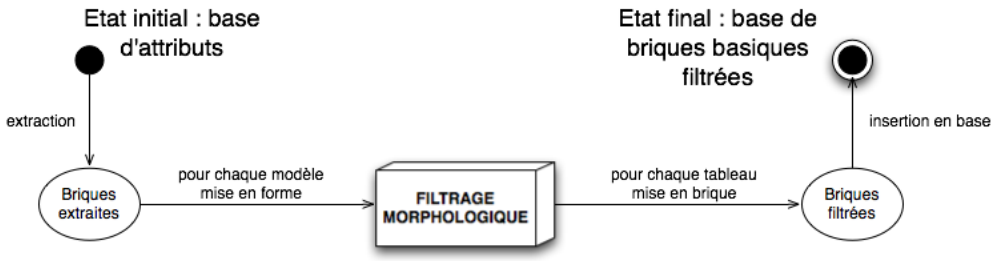


Figure 4.11 — Processus général permettant à partir d'une base d'attributs extraits d'obtenir une base de briques basiques filtrées.

- "combler" les trous des briques quand ceux-ci sont suffisamment petits ;
- ne pas altérer la taille des briques c'est à dire qu'on ne doit pas modifier les bornes minimale et maximale d'une brique. Par exemple une brique allant de l'image 100 à 200 et présentant des trous de 110 à 112 et de 127 à 129 doit être au final une brique continue de 100 à 200 (sans augmentation de la taille (de 100 vers 98 ou de 200 vers 202) - problème des effets de bord).

Pour réaliser cette étape de filtrage, nous avons utilisé les techniques issues de la morphologie mathématique [Serra, 1982, 1988]. Nous avons mis en place un filtrage à base de fermeture et d'ouverture binaire sur nos briques extraites.

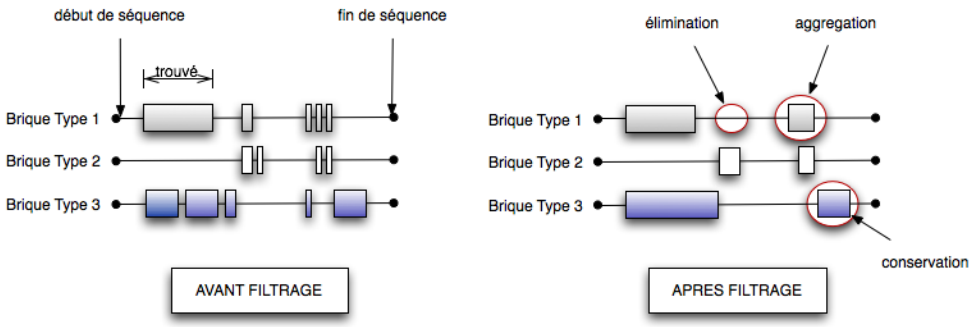


Figure 4.12 — Illustration du filtrage morphologique sur les briques

La figure 4.12 illustre le fonctionnement de notre filtre sur les briques. Celui-ci permet ainsi de garder les briques d'une taille compatible avec les définitions des modèles utilisés sans modifier leur taille, d'éliminer les briques de trop petite taille et de combler les trous de petites tailles. Les techniques de filtrage morphologique sont basées sur l'utilisation d'un élément structurant. La taille minimale des briques et celle des trous dépend de l'élément structurant utilisé dans le filtrage. L'élément structurant est à définir en fonction des données et du type des corrections à apporter.

Le choix de l'élément structurant

Le choix de cet élément est très important puisqu'il conditionne la taille admissible pour les trous comme pour les briques. Nous faisons d'abord une fermeture suivie d'une ouverture. Au cours de ces opérations, l'érosion et la dilatation doivent être réalisée avec le même élément structurant afin de garantir la conservation de la taille des briques après filtrage. Néanmoins, entre la fermeture et l'ouverture, on peut changer l'élément structurant.
Concernant la forme (convexe), pour des raisons de continuité, nous avons choisi d'utiliser un élément structurant sur des images consécutives sans trous. La seule caractéristique qui peut donc être modifiée est la taille de l'élément. Nous posons également la contrainte de symétrie de l'élément structurant ce qui conduit à ce que la taille de celui-ci soit impaire. Afin de déterminer les dimensions de ces deux éléments structurants, nous avons fait varier celles-ci et nous avons analysé le nombre et la qualité des briques extraites du processus en nombre (évaluation quantitative) et en qualité (évaluation qualitative). La quantité est représentée par le nombre moyen de briques par séquence et par modèle. La qualité est indiquée par le nombre de modèles représentés. Le tableau 4.2 regroupe les résultats de ce test (lors de ce test, le système était composé de 60 briques, ce nombre a été porté à 75 par la suite).

A partir de ces résultats, on peut établir que le meilleur compromis semble être l'utilisation d'un élément structurant de taille 3 pour la fermeture comme pour l'ouverture. Il permet d'optimiser la quantité de briques (autour de 20 par modèle et par séquence) en moyenne tout en garantissant un nombre de modèles représentés convenable.

La dernière étape consiste à partir du résultat du filtrage à stocker ces informations sous la forme de briques. Dans le but de minimiser l'occupation mémoire du stockage, il n'est pas nécessaire de stocker les briques trouvées par séquence, par image et par objet. Il suffit de stocker le type de brique (référence du modèle), la séquence dans laquelle on retrouve celle-ci, l'image de début et l'image de fin.

Par exemple, concernant la brique A dont le modèle est basé sur une caractéristique image, on avait stocké 1 entité dans la table 'sequence' (séquence X) et 23 entités dans la table 'image' (1 par image). Après filtrage, on stocke 3 entités :
- brique A - séquence X - image 12 à 16
- brique A - séquence X - image 33 à 59
- brique A - séquence X - image 67 à 104

On est donc passé de 24 entités stockées à 3. La recherche est simplifiée, le nombre de brique réduit au minimum et le positionnement temporel aisément réalisable.

4.6 Extraction des briques combinées

Il s'agit maintenant d'extraire les briques combinées correspondant aux modèles décrits dans la base de modèles à l'aide des briques de base qui ont été extraites précédemment.

4.6.1 Processus général

Le processus général permettant de passer d'une base de briques basiques à une base de briques combinées ou de concepts est composé de deux étapes à l'instar de l'extraction

Fermeture	Ouverture	Nombre	Moyenne par séquence et par modèle	Nombre de modèles représenté
1	1	56	93,3 briques	60
1	3	27	45 briques	60
1	5	20	33,3 briques	49
1	7	13	21,6 briques	34
1	9	2,7	4,5 briques	11
3	1	27	45 briques	60
3	3	12	20 briques	32
3	5	5	8,3 briques	29
3	7	2,1	3,5 briques	16
3	9	0,8	1,3 briques	8
5	1	20	33,3 briques	49
5	3	5	8,3 briques	29
5	5	1,9	3,2 briques	17
5	7	0,7	1,2 briques	6
5	9	0,05	0,8 briques	2
7	1	13	21,6 briques	34
7	3	2,1	3,5 briques	16
7	5	0,7	1,2 briques	6
7	7	0,08	0,13 briques	3
7	9	0,02	0,03 briques	2
9	1	2,7	4,5 briques	11
9	3	0,8	1,3 briques	8
9	5	0,05	0,8 briques	2
9	7	0,02	0,03 briques	2
9	9	0,004	-	1

Unité du nombre : (X∗100.000) briques

Tests effectués sur une base composée
de 1000 séquences hétérogènes
et 60 modèles de brique basique

Tableau 4.2 — Influence de la taille de l'élément structurant sur la quantité et la qualité des briques obtenues

des briques basiques. La première est une phase de d'extraction où à partir des définitions (briques basiques/combinées et opérateurs), on recherche les briques correspondantes. La seconde permet de filtrer ces données, la mise en forme des données filtrées et l'insertion en base de données. Cette dernière étape est similaire à celle correspondante lors de l'extraction des briques basiques.

Lors de l'extraction des données pour les briques basiques, il suffisait d'effectuer une requête sur notre base afin d'obtenir toutes les localisations du modèle basique considéré. Pour les briques combinées, il ne s'agit plus d'une valeur mais d'une définition. Il va donc falloir définir un certain nombre de processus nous permettant à partir d'une définition donnée, de produire la requête correspondante afin d'effectuer la récupération des localisations des briques combinées recherchées.

4.6.2 Sélection des séquences prototypes

4.6.2.1 Construction de la requête

L'extraction des briques combinées consiste à récupérer les séquences présentant le modèle des briques combinées. Le modèle est construit à partir de plusieurs briques basiques et/ou combinées (formées par des briques basiques et/ou combinées) et d'opérateurs. Il se résume donc à chercher un certain nombre de briques basiques dans les séquences et de vérifier leurs positions respectives afin de déterminer si le modèle est bien respecté.

4.6.2.2 Recherche en plusieurs phases

Rechercher simultanément les briques basiques, les briques combinées, les opérateurs tout en vérifiant les positions respectives, en effectuant des requêtes en base de données est une opération difficilement réalisable. De ce fait, la recherche est divisée en plusieurs étapes : une étape permettant de diminuer le niveau conceptuel des briques, une étape permettant la présélection des séquences et enfin la recherche des modèles temporels et logiques.

Diminution du niveau conceptuel

On commence par diminuer le niveau conceptuel des briques. Pour cela, on extrait les briques constitutives de chaque brique combinée. Si cette brique est constituée de briques combinées, on fait de même avec chacune des briques constitutives jusqu'à n'avoir plus qu'une liste de briques basiques. Cette étape est réalisée algorithmiquement :

```
1 TANT QUE (toutes les briques sont basiques)
2   POUR CHAQUE (brique constitutive)
3     Recuperer les briques constitutives de la brique combinee courante
4     Recuperer l operateur concernant ces briques (si il existe)
5   FINPOUR
6 FINTANTQUE
```

Listing 4.4 — Algorithme de réduction du niveau conceptuel

Pré-sélection des séquences

On réalise une pré-sélection des séquences en effectuant une recherche permettant de récupérer les séquences présentant toutes les briques basiques constitutives que l'on a identifiées lors de l'étape de diminution du niveau conceptuel. A ce niveau, on peut utiliser un seuil de tolérance dans la recherche. Ce paramètre permet également d'effectuer du relâchement de contraintes. Par exemple, toute séquence ayant 80% des briques basiques constitutives est admissible. Cette étape est réalisée par des outils de base de données au moyen d'une requête.

Recherche des modèles temporels et logiques

Sur les séquences pré-sélectionnées, on effectue la recherche des positions respectives des briques entre elles. Cette étape est réalisée en plusieurs phases, de manière algorithmique :

```
POUR (chaque brique)
  Recuperer la position du debut de la brique courante
  Recuperer la position de la fin de la brique courante
  Stocker les valeurs des poitions
FINPOUR
POUR (chaque sequence)
  POUR (chaque brique constitutive)
    POUR (chaque brique constitutive apres la brique courante)
       Comparer la brique courante avec la brique suivante
       Positionner les briques entre elles
    FINPOUR
  FINPOUR
FINPOUR
```

Listing 4.5 — recherche des modèles

Pour cela, on fait le chemin inverse effectué lors de l'étape de diminution du niveau conceptuel. On prend chaque relation briques/opérateurs permettant la construction d'une brique combinée, on recherche celle-ci dans les briques disponibles des séquences pré-sélectionnées. On construit ainsi les briques combinées (constituants et repère temporel (début-fin)). Cette étape se poursuit jusqu'à l'obtention de la définition finale (brique concept).

A la fin de cette étape, on dispose d'une liste de séquences potentiellement intéressantes, appelées les prototypes dans la suite (L'évaluation de cette étape sera présentée dans le dernier chapitre).

4.7 Perspectives

4.7.1 Tolérance de la recherche

La recherche d'une définition d'une manière stricte peut ne pas permettre la récupération de toutes les briques attendues. En effet, par exemple, si une brique basique utilisée dans la définition et celle qu'on retrouve en réalité dans les concepts sont différentes, les recherches ne fonctionneront pas même si la différence n'était que d'un intervalle. Pour remédier à ce problème, nous suggérons l'utilisation de mécanismes de tolérance par relâchement de contrainte, lors de la recherche, avec l'attribution d'un niveau de confiance.

Le principe est le suivant : on commence par rechercher en utilisant la définition exacte qui a été produite. Si le nombre de résultats est voisin de zéro, on commence à relâcher certaines contraintes (sur la sélection des modèles de briques, la sélection des opérateurs,...) lors de la recherche pour essayer d'augmenter le nombre de résultats produits. La libération de ces contraintes dégrade le niveau de confiance que l'on va attribuer aux résultats de la recherche.

4.7.2 Libération de contraintes

Les contraintes qui sont *imposées* par la définition concernent le choix des briques et le choix des opérateurs de combinaison. De ce fait, la libération de contraintes consiste, soit à faire évoluer le choix des briques constitutives, soit à modifier l'importance ou l'ordre des opérateurs (selon le type d'opérateur initialement utilisé). Elle s'apparente aux méthodes de relâchement de contraintes des problèmes d'optimisation combinatoire (comme les CSP - Problèmes à Satisfaction de Contraintes).

Relâchement sur les briques

Le relâchement sur les briques consiste à élargir la définition initiale. Pour cela, on remplace une brique basique liée au concept par une brique combinée construite avec la brique basique concernée et les briques basiques ayant une définition adjacente. Par exemple, la brique compacité-faible peut être remplacée par une brique combinée définie par la compacité-faible OU compacité-moyenne. La brique résultante correspond à un intervalle de définition plus important que la brique initiale. Il est important de noter toutefois que le remplacement d'une brique basique par une brique combinée peut conduire à l'élimination complète de la contrainte. Par exemple, lors du remplacement de la brique ('compacité-moyenne') par la brique ('compacité-faible' OU 'compacité-moyenne' OU 'compacité-forte') implique qu'on ne prend plus en compte le critère de compacité dans la définition. La principale difficulté réside dans le choix de la brique sur laquelle on va relâcher la contrainte.

Relâchement sur les opérateurs

Le relâchement sur les opérateurs dépend du type d'opérateur. Par exemple, sur les

opérateurs précisant la dimension temporelle des briques, le relâchement est effectué en supprimant les informations de durée, ou sur les opérateurs précisant la position des briques, on peut les remplacer par des opérateurs logiques (ET/OU). Mais il conviendra de faire une étude complète des opérateurs à remplacer pour libérer les contraintes.

Niveau de confiance

La libération de contraintes, même si elle permet d'améliorer sensiblement la bonne détection de certaines briques concepts, entraîne également une augmentation des fausses détections c'est à dire une augmentation du rappel et une dégradation de la précision. Il est donc important de donner à l'utilisateur un indicateur lui permettant d'évaluer la qualité des recherches effectuées. On peut par exemple utiliser un indicateur de confiance qui est une simple valeur. Plus celle-ci est élevée et plus l'utilisateur peut avoir confiance dans les résultats des prototypes sélectionnées. L'utilisation de plusieurs réponses lors d'une question, qui entraîne l'élargissement de la définition dégrade cet indicateur tout comme le relâchement de contraintes.

4.8 Conclusion

Nous avons présenté dans ce chapitre un modèle de représentation des données permettant de tenir compte de l'aspect spatio-temporel de données puis de représenter des concepts en utilisant des opérateurs de combinaison et de liaison. Cette modélisation, associée à des mécanismes d'opération sur les bases de données, permet d'extraire de façon automatique des briques définies comme suites d'images à des niveaux conceptuels différents. Nous avons distingué la brique de base, liée à une propriété sur une caractéristique particulière et la brique concept susceptible de correspondre au concept recherché. Néanmoins, la définition de ces modèles pose un problème de construction sur les choix des briques et des opérateurs.
Nous nous retrouvons de nouveau avec le problème du fossé sémantique où d'un côté nous disposons de données (les modèles) et d'un expert en traitement d'images (ETI) ; et de l'autre, des définitions (briques combinées) et l'expert applicatif (EA) non expert en traitement d'images ni dans le modèle de représentation proposé (briques).
Pour résoudre ce problème de liaison entre données et concept (brique combinée), nous avons fait le choix d'utiliser un système d'assistance sous forme de questions/réponses. Ce système va permettre de récupérer la connaissance de l'expert applicatif et de relier cette connaissance aux modèles de briques.

Troisième partie

Interaction avec l'utilisateur

5 Définition des concepts, vers la réduction du fossé sémantique

« La valeur d'une idée dépend de son utilisation. »

Thomas Edison

Un concept est la représentation intellectuelle d'une idée abstraite. Dans le cadre de notre application, c'est une représentation générale et abstraite d'une réalité. Par exemple, le concept 'courir' dans une vidéo est rattaché au fait qu'il y a un personnage vu en entier qui se déplace rapidement. Un concept a une signification pour l'utilisateur, à un instant donné et pour une application donnée. Par exemple, la définition du concept 'sauter' est différente selon que l'on souhaite l'appliquer à des données issues de séquences d'athlétisme, de gymnastique, de motocross ou de séquences de vidéo-surveillance. Cette définition n'est pas figée, elle peut évoluer dans le temps soit parce que l'application évolue ou est différente soit parce que l'expert a évolué ou est différent. Un concept peut donc présenter un caractère non stationnaire.

Indexer une vidéo consiste à rattacher un ou plusieurs concepts à des segments de cette vidéo. L'indexation automatique se base sur l'extraction automatique de caractéristiques fournies par un système de traitement d'images. Cependant, il est nécessaire de définir les index ou concepts. Pour cela il faut définir le lien qui existe entre ces caractéristiques et ces concepts. Ce qui sépare les caractéristiques extraites sur lesquelles se base l'indexation automatique et les concepts est appelé **fossé sémantique**. Par définition [Smeulders *et al.*, 2000], le fossé sémantique est le manque de concordance entre les informations que les machines peuvent extraire depuis les documents numériques et les interprétations que les humains en font. La définition d'un concept peut être faite automatiquement si l'on dispose d'une base d'apprentissage liée au concept. Dans ce cas, il est possible *d'apprendre* le concept de manière statistique. Mais la construction de cette base d'apprentissage nécessite de faire intervenir un utilisateur ou un Expert Applicatif. En fait, il s'agit de s'appuyer sur les connaissances de l'expert applicatif pour extraire des segments vidéo représentatifs du concept que l'on souhaite définir. On peut lui demander d'indexer manuellement la base d'apprentissage, mais cette opération est longue et fastidieuse. Dans ce chapitre, nous proposons une méthode qui permet d'extraire l'expertise pour que l'implication de l'expert soit la plus simple et la plus

limitée possible.

Le problème de l'extraction de connaissances [Hoc, 1995; Azé, 2003] est un problème récurrent qui est né dans les années 60. Les premiers développements concernaient notamment les problèmes de classification. On a vu d'abord émerger dans les années 80, les systèmes experts [Darlington, 2000; Walker et al., 1990; Jackson, 1998; Ignizio, 1991; Giarratano et G.Riley, 1998] qui ont introduit les méthodes par induction. Ensuite, les années 90 ont vu apparaître les méthodes basées sur les réseaux de neurones [Wasserman, 1989; Agre, 1997; Arbib, 1995; Muller et Insua, 1995].

Les systèmes de questions/réponses que l'on rencontre habituellement sont construits afin d'aider l'utilisateur à construire une recherche sur un ensemble de données *préparées* (étiquetées, indexées, classées, annotées, etc). L'approche proposée dans ce chapitre consiste à utiliser un système de questions/réponses qui permette à l'expert applicatif de définir le concept en limitant son travail. De tels systèmes ont déjà été développés pour d'autres domaines d'application. On peut citer le système 20Q[1] et Akinator[2]. Ces deux systèmes de questions/réponses ont pour objet de deviner le concept auquel pense l'utilisateur en quelques questions. Ce sont des systèmes collaboratifs qui sont basés sur des réseaux de neurones et qui évoluent au fil des utilisations.

Dans le processus global de construction de base d'apprentissage d'un concept (voir chapitre introduction), cette phase de questions/réponses est généralement la première car elle permet de combler, au moins partiellement, le fossé sémantique entre les caractéristiques extraites, et donc les briques telles qu'elles ont été définies dans le chapitre précédent et le concept que souhaite illustrer l'**Expert Applicatif**. Le processus représenté dans la figure 5.1 fait apparaître trois étapes principales :

- le système de questions-réponses ;
- l'extraction de briques concepts (voir chapitre précédent) ;
- la validation.

La construction du jeu de questions/réponses est réalisée dans une phase préliminaire par l'**Expert en Traitement d'Images** qui fait l'effort du passage du fossé sémantique. L'objectif est de construire une structure suffisamment générique en termes d'application composée de questions/réponses rédigées en langage commun et dont les réponses correspondent à des propriétés sur les caractéristiques extraites des vidéos. En effet, par la création de cette structure, l'expert en traitement d'images crée des liens entre les couples de questions/réponses et les caractéristiques dont il a la maîtrise de l'extraction. En revanche, il n'a a priori pas de connaissances sur l'application traitée et c'est dans le processus de construction du concept par l'expert applicatif que cette connaissance sera intégrée.

Dans ce chapitre, nous nous intéressons à la phase de construction des questions-réponses puis à leur utilisation (figure 5.2 - Etape 2). Nous présentons en premier lieu (section 5.1), les différents types de questions/réponses ainsi que leur structuration (section 5.2). Ensuite, nous expliquons la méthode suivie pour construire des questions/réponses dans le

1. 20Q - http :// www.20q.net - Jeu de questions/réponses en 20 questions permettant de deviner un concept (objet, sentiment, animal,...)
2. Akinator - fr.akinator.com - Jeu de questions/réponses permettant de deviner un personnage

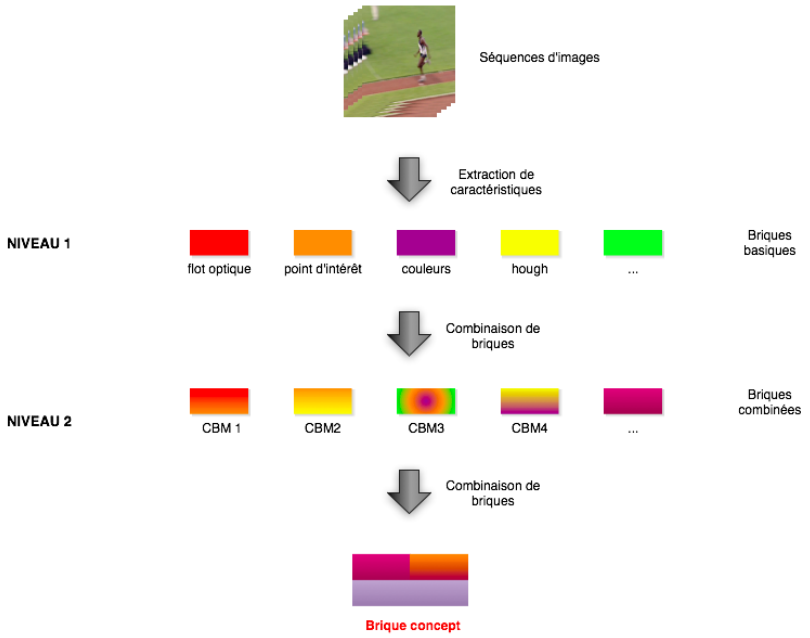


Figure 5.1 — Processus général d'extraction de briques concepts.

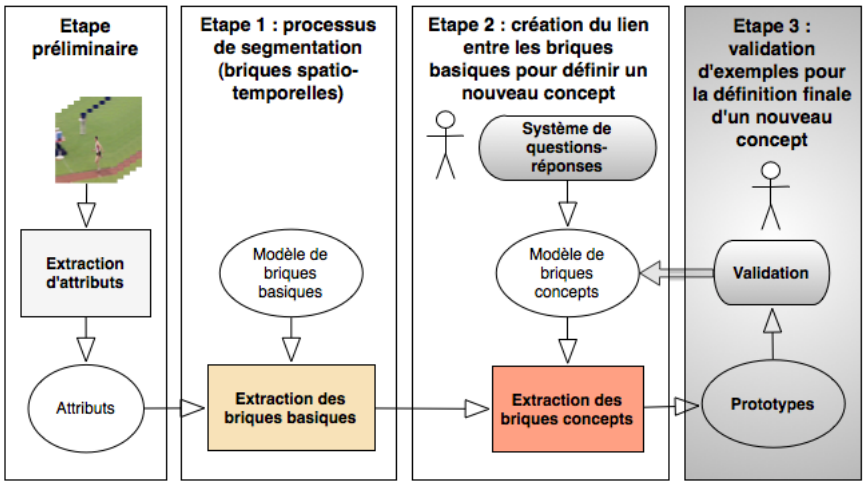


Figure 5.2 — Processus général du système proposé.

but de surmonter le fossé sémantique (section 5.3). Enfin, la section 5.4 nous permet d'expliquer le fonctionnement de la validation des prototypes sélectionnés avant de présenter la structuration informatique et les améliorations envisagées du système de questions/réponses (section 5.7).

5.1 Système de questions/réponses

Le système de questions/réponses doit répondre à un certain nombre de problématiques au regard de la structuration en briques décrite dans le chapitre précédent :

- il doit permettre de lier des réponses à des briques de base ;
- il doit permettre de définir les liaisons entre briques afin de définir des briques composées ;
- il doit assurer une certaine cohérence dans le processus de questionnement (éviter les questions sans objet, maintenir une suite "logique" dans les questions).

Afin de répondre à ces problématiques, nous avons développé un système construit sur trois types différents de questions, et un certain nombre d'actions liées aux différents types de réponses. L'ensemble est structuré sous forme d'arborescence permettant d'organiser l'ordonnancement du processus de questionnement.

5.1.1 Les questions

Les questions sont de trois types : celles permettant d'obtenir des informations sur le concept (elles sont réliées aux briques, c'est le cas de la plupart d'entre elles), celles permettant d'obtenir des informations sur les liaisons (elles sont reliées aux opérateurs) et enfin celles permettant d'obtenir des informations de navigation (elles donnent des informations d'accessibilité sur les questions utilisables après la question courante).

Les questions à "réponses - briques"

Les questions à "réponses - briques" (la plupart des questions sont de ce type) permettent la sélection de briques de base liées au concept à définir. Une réponse est liée à une ou plusieurs briques de base (la liste des liaisons est disponible en annexes C).
On observe trois cas (figure 5.3) : aucune sélection de briques (réponse B), une brique sélectionnée (réponse C), ou plusieurs briques sélectionnées (réponse A).

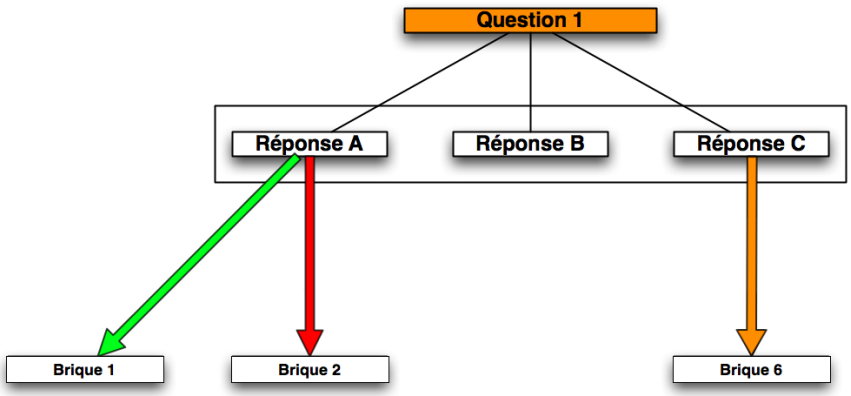


Figure 5.3 — Illustration des différents cas pour les questions à réponses - briques.

Par exemple, dans le système actuel, les réponses à la question "Comment peut-on qualifier le mouvement ?" comporte trois réponses : 'lent', 'moyen' et 'rapide'. Ces réponses sont liées aux briques concernant la vitesse de l'objet en mouvement (VO). Les liaisons sont les suivantes (pour cet exemple, les liaisons sont simples et plutôt évidentes) :

- la réponse 'lent' permet la sélection de la brique VO-lent ;
- la réponse 'moyen' permet la sélection de la brique VO-moyen ;
- la réponse 'rapide' permet la sélection de la brique VO-rapide.

Autre exemple, les réponses aux questions liées au mouvement de caméra comme par exemple "Y a-t-il un déplacement par rapport au sol ?", permettent la sélection de 2 à 4 briques basiques. Le mouvement de caméra est relié à plusieurs briques de base concernant le type de caméra (fixe ou mobile), l'orientation du mouvement de caméra (mouvement dominant) et les caractéristiques du flot optique (continu et invariant directionnel). Enfin, les réponses à la première question (qui représente un cas particulier étant le point de départ unique à chaque nouvelle définition) demandant "Quel est le type d'activité ?" ne sont liées à aucune brique.

Les questions à "réponses - liaison"

Les questions à "réponses - liaison" sont plus difficiles à construire. Il s'agit de questions qui permettent la sélection d'opérateurs de liaison entre briques (figure 5.4). Ce qui est particulièrement délicat n'est pas le choix d'un opérateur mais le choix des briques à lier à l'aide de l'opérateur choisi.
Le choix peut être effectué de plusieurs manières distinctes :

- la réponse sélectionne simultanément un opérateur de liaison et des briques de base (figure 5.4 - cas de la réponse A). L'opérateur concerne directement les briques sélectionnées. Ce cas est très rare car il correspond à la sélection directe d'une brique combinée.
- la réponse sélectionne uniquement un opérateur sans sélectionner de briques et sans contrainte ou règle (figure 5.4 - cas de la réponse B). C'est le système qui cherchera à appliquer l'opérateur sur les briques sélectionnées au moment de la définition finale. Ce sont les prototypes validés qui permettront de définir les briques effectivement liées à l'aide de l'opérateur.
- la réponse sélectionne un opérateur et une règle de liaison qui définit les briques admissibles à l'usage de l'opérateur (figure 5.4 - cas de la réponse C). Ainsi, si les briques admissibles sont déjà sélectionnées, elles sont liées par l'application de l'opérateur. Sinon, on attend leur sélection. Dans le cas où elles ne sont pas sélectionnées à la fin du processus, l'opérateur n'est pas utilisé dans la définition finale.

Par exemple, la question "Peut-on distinguer plusieurs phases distinctes dans le mouvement ?" permet la sélection de l'opérateur de liaison "XmY" (Y démarre après X). Le nombre d'opérateurs "XmY" à utiliser dans la définition est ensuite déterminé par la question "Combien y a-t-il de phases distinctes ?".

Les questions à "réponses - navigation"

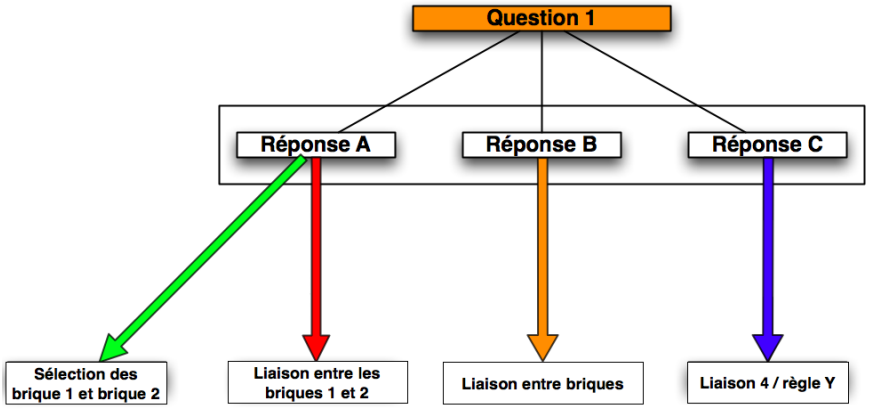


Figure 5.4 — Illustration des différents cas pour les questions à réponses - liaison.

Suivant les applications (concept à définir), certaines questions n'ont pas de sens et donc n'ont pas à être posée comme par exemple la question "Est-ce un dialogue, une manifestation, une réunion ?" quand on souhaite définir le concept "marcher". D'autres n'ont un sens que si elles ont été précédées par d'autres questions ("Le mouvement est-il localisé à une partie du corps en particulier ?" permet par les réponses "Oui/Souvent/Parfois" d'accéder ensuite à la question "Plutôt antérieure ou postérieure ?"). Nous avons développé un mécanisme d'"activation-désactivation" des questions permettant de naviguer dans l'ensemble des questions. La figure 5.5 présente les différents cas et le fonctionnement des questions "réponses-navigation".

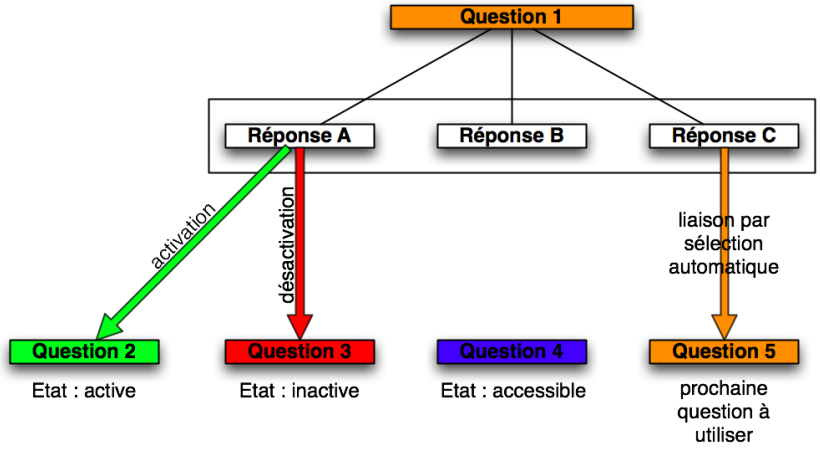


Figure 5.5 — Illustration des différents cas pour les questions à réponses - navigation.

Les questions actives sont les questions potentiellement sélectionnables, les questions inactives ne sont pas sélectionnables et ne le seront plus pour le processus en cours de définition, les questions accessibles sont des questions qui peuvent encore être activées/désactivées et les questions liées sont les questions qui sont utilisées séquentiellement.

Comme nous le verrons dans 5.2.2, les questions sont classées par thèmes. Ceux-ci ne sont ouverts que par certaines réponses permettant ainsi d'avoir une gestion du lot de questions utilisables

5.1.2 Les réponses

Les systèmes de questions/réponses proposent à l'utilisateur de répondre selon des modalités différentes qui sont notamment dépendantes du type de réponses utilisées. Dans le cas général, on peut distinguer plusieurs types de réponses :

- les réponses ouvertes : ce sont des réponses où on ne propose pas de réponses à la question posée. L'utilisateur est libre de répondre ce qu'il souhaite, soit par saisie de valeur soit par l'utilisation d'un curseur de positionnement
- les réponses fermées : ce sont des réponses où on ne propose qu'un choix restreint à la question posée. L'utilisateur doit choisir parmi celles proposées obligatoirement, en sélectionnant une seule case (choix simple) ou plusieurs cases (choix multiples).
- les réponses à valeurs discrètes : ce sont des réponses dont les valeurs sont dans un ensemble discret comme par exemple les réponses jour/nuit ou intérieur/extérieur.
- les réponses à valeurs continues : ce sont des réponses dont les valeurs sont dans des ensembles de définition continus comme par exemple les réponses à une question sur la vitesse d'un objet et qui nécessitent la réponse sous forme numérique.

Voici quelques exemples de questions ouvertes/fermées sur des réponses à valeurs discrètes/continues :

- ouverte discrète : concernant le temps - beau, chaud, nuageux, gris, sombre, etc. ;
- ouverte continue : concernant la vitesse - saisie d'une valeur algébrique ;
- fermée discrète : concernant le temps en laissant le choix entre beau et mauvais ;
- fermée continue : concernant la vitesse - découpage en nombre fini d'intervalles continues (lent/moyen/vite) qui correspondent à des sous-ensembles de l'espace admissible.

Dans notre application, nous avons choisi d'utiliser des réponses à valeurs discrètes, à choix fermé et multiple. Ce choix permet de faciliter le travail de création des liaisons avec les briques basiques et les opérateurs par l'expert en traitement d'images.

On peut considérer quatre cas qui permettent d'obtenir un niveau de certitude sur la qualité de la réponse donnée :

- aucune réponse sélectionnée : dans ce cas, aucune réponse n'a satisfait l'utilisateur. Ce qui signifie que la question n'était pas appropriée.
- une seule réponse sélectionnée : dans ce cas, la réponse est précise et le niveau de qualité maximal.
- deux ou plusieurs réponses sélectionnées : dans ce cas, les réponses sont imprécises et la sélection des prototypes sera peu contrainte c'est à dire qu'on aura une baisse de la précision sur la sélection. En résumé, en ce qui concerne la sélection des prototypes, plus le nombre de réponses est important, plus le rappel est bon mais la précision faible car on récupère plus de prototypes par rapport à une sélection d'une seule réponse entraînant une définition plus fine.

– toutes les réponses sélectionnées : dans ce cas, toutes les réponses satisfont l'utilisateur. La question n'est donc pas sélective pour le concept en cours de définition.

Le cas où aucune réponse n'est sélectionnée et le cas où toutes les réponses sont sélectionnées sont particulièrement problématiques. Cela indique que la question n'était pas appropriée ou inutile.

5.2 Modèle de représentation

5.2.1 Introduction

Le système de questions/réponses représente les connaissances de l'Expert en Traitement d'Image sur les caractéristiques extraites de la vidéo et sur l'interprétation générique qui peut en être faite. Afin qu'il soit efficace, ce système doit être bien structuré. La représentation des connaissances est au cœur des recherches en psychologie cognitive et en intelligence artificielle [Ferber, 1989].

5.2.2 Structuration choisie

Nous avons choisi d'utiliser le formalisme objet, qui n'est pas spécifique à l'IA, organisé sous forme de graphe de type arbre. Cette représentation permet de modéliser les questions/réponses où chaque entité est composée d'une question, de réponses associées, et de relations vers des briques, des opérateurs ou d'autres questions mais aussi de représenter la hiérarchie entre questions et le regroupement thématique. Chaque question/réponses est un objet composé d'un objet question et d'objets réponse. Les informations sur chaque question sont représentées par les attributs de l'objet et les relations entre objets. La question de la sélection de la question est un problème important. En effet, dans le but de définir de manière précise et rapide le concept, il est nécessaire d'utiliser les questions les plus pertinentes c'est à dire celles qui vont apporter le plus d'information. Par exemple, une question liée aux mêmes briques qu'une autre question à laquelle on a déjà répondu n'apporte pas d'information, elle ne fait que confirmer une information dont on dispose déjà. Le deuxième point qui nous semble important est la cohérence des questions. Par exemple, si on commence à poser des questions sur l'environnement dans lequel apparaît le concept, il ne semble pas très pertinent de l'interroger directement sur la vitesse du mouvement pour revenir à la question suivante sur l'environnement.

Nous avons choisi de représenter les questions/réponses sous forme d'arbre planaire enraciné [3]. Cette représentation permet un parcours des sommets dans l'ordre lexicographique en utilisant le parcours en profondeur préfixé. Chaque nœud représente une question/réponses qui dispose d'un statut (accessible/activée/désactivée/utilisée). Seules les questions activées peuvent être sélectionnées. Cette représentation nous permet de regrouper les questions par niveau (notion de hiérarchie) en formant des sous-graphes (groupes thématiques)

3. L'arbre planaire est un graphe que l'on peut dessiner dans le plan sans que ses arêtes ne se touchent, sauf à leurs extrémités. Si on choisit un sommet r quelconque dans un arbre, il est possible d'enraciner l'arbre en r, c'est-à-dire orienter toutes les arêtes de sorte qu'il existe un chemin de r à tous les autres nœuds.

et en représentant les relations inter-objets (arêtes/arcs). Par définition, un sous-graphe est un sous-ensemble d'un graphe. Ainsi, dans notre modèle, chaque thème forme un sous-graphe de l'arbre qui peut lui-même contenir des sous-graphes.
Le modèle de l'arbre "objet" proposé se représente ainsi :

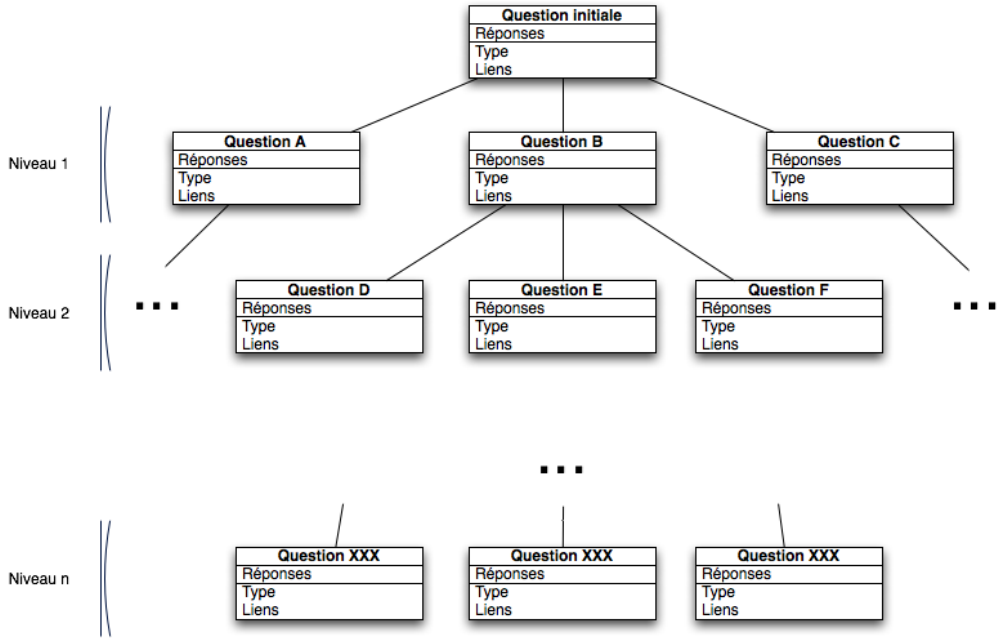


Figure 5.6 — Modèle de l'arbre objet proposé.

Le fait d'avoir choisi une structure d'arbre planaire enraciné implique que le système de questions/réponses démarrera toujours par une question initiale, la racine.
Le système actuel dispose d'un lot de 32 questions. Ce nombre n'est pas très important mais pour une moyenne de 3 ou 4 réponses pour chacune d'elle, cela permet déjà un large choix de définitions.
Les objets constituants de l'arbre sont les différentes questions/réponses et sont groupées par thématique. Les 32 questions du système actuel sont réparties en 6 thèmes qui sont : communication, environnement, mouvement, objet, personnage et déplacement (tableau 5.1). Ces thèmes ont été définis de manière empirique pour couvrir l'ensemble des modèles de briques disponibles. Dans la construction de l'arborescence par les thèmes et par les questions, il est difficile de rester complètement générique. Nous parlerons de ce point en particulier dans le paragraphe suivant.

Par exemple, pour la première question (Quel est le type d'activité ?), la réponse 'communication' ouvre des questions du thème 'communication' et ferme des questions du thème 'déplacement' alors que la réponse 'mouvement' ouvre des questions du thème 'environnement' et 'mouvement'.

Thème	Nombre de questions associées	exemples (questions)
Mouvement	6	Y a-t-il déformation pendant le mouvement ?
Communication	8	Est-ce un dialogue, une manifestation, une réunion ?
Environnement	9	Est-ce une activité intérieur ou extérieur ?
Objet	2	Y a-t-il besoin d'un objet spécifique ?
Personnage	2	Combien de personnes sont concernées ?
Déplacement	4	Y a-t-il un déplacement par rapport au sol

Tableau 5.1 — Les différents thèmes et la répartition des questions par thème.

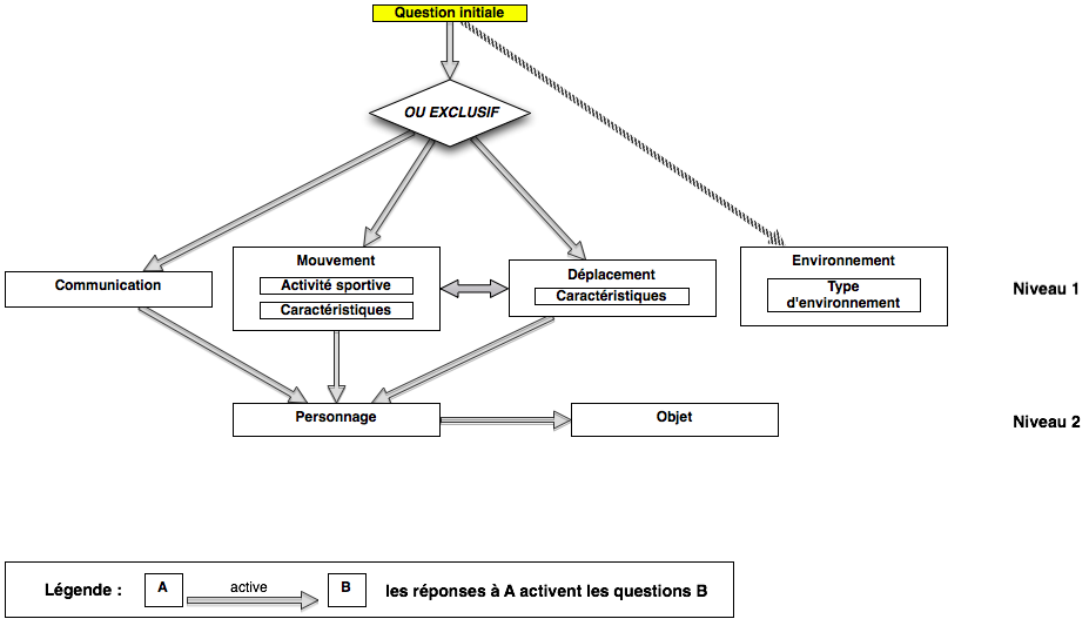


Figure 5.7 — Arborescence des différents thèmes proposés.

5.2.3 Description de la structure

La représentation arborescente hiérarchique des différents thèmes et de leurs sous-thèmes associés est représentée dans la figure 5.7. La sélection des questions à utiliser correspond à un parcours arborescent sous contraintes. Dans l'arborescence actuelle, on peut distinguer 2 niveaux dans les thèmes.

Le premier niveau est accessible directement après la première question et contient les thèmes 'communication', 'mouvement' et 'déplacement' qui sont directement liés à des réponses. Le thème 'environnement' est un thème qui est toujours rendu accessible quelle que soit la première réponse donnée - il s'agit d'un choix délibéré car nous avons considéré

que pour toute activité l'information de l'environnement était importante. La racine est la question initiale (Quel est le type d'activité ? et les réponses associées (Communication/-Mouvement/Déplacement/Indéterminé)).

Dans chacun des thèmes, certaines questions peuvent être hiérarchisées. En effet, certaines sont accessibles dès l'ouverture du thème alors que d'autres ne s'ouvrent qu'après avoir répondu à un certain nombre de questions du thème ou par liaison directe avec une question utilisée. Par exemple, dans le thème 'mouvement', on s'intéresse au nombre de phases distinctes que peut présenter le mouvement en cours de définition. Cette question ne peut intervenir avant celle permettant de demander si effectivement on peut décomposer le mouvement en plusieurs phases distinctes.

On peut également avoir un ou plusieurs sous-groupes thématiques dans un même thème qui ne seront pas accessibles en même temps. Par exemple, on retrouve un sous-groupe 'milieu naturel' dans le thème 'environnement' qui n'est ouvert que si on est en 'extérieur' et que l'on ne se situe pas 'en ville'. La représentation arborescente hiérarchique des différents thèmes et de leurs sous-thèmes associés est représentée dans la figure 5.7.

5.2.4 Parcours dans l'arbre

Pour effectuer ce choix de manière simple et surtout sans utiliser l'étude statistique afin d'être utilisable immédiatement, la sélection des questions s'effectue en faisant un parcours en profondeur de l'arbre et ce uniquement en sélectionnant les feuilles activées (les questions). En effet, l'arbre est construit de telle sorte que chaque branche est un sous-graphe thématique ou un sous-graphe de raffinement. Ainsi, on accède à une question donnée puis aux questions plus fines sur cette même information ou thématique. Une fois le thème épuisé, on remonte dans l'arbre. Afin d'améliorer la dynamique des questions, lorsque l'on remonte dans l'arbre, la sélection de la question à utiliser n'est pas directement la suivante mais elle est sélectionnée par tirage aléatoire parmi les questions actives (figure 5.8), dans la même branche et au même niveau.

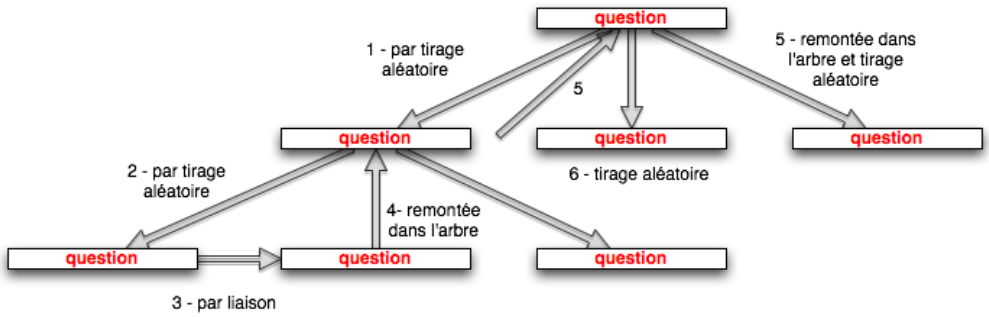


Figure 5.8 — Exemple du parcours de l'arborescence (on suppose que toutes les questions de cette partie sont activées).

5.3 Construction des questions-réponses : vers le passage du fossé sémantique

La construction des questions-réponses est réalisée dans le but de faire en sorte que l'effort du passage du fossé sémantique soit effectué par l'Expert en Traitement d'Images et ainsi minimiser l'effort de l'Expert Applicatif. Dans la mesure du possible, nous avons chercher à construire des questions/réponses présentant un caractère générique indépendant de l'application. Cependant, dans la pratique, cette indépendance vis-à-vis de l'application est souvent difficile à obtenir.

La difficulté réside dans la rédaction de questions/réponses claires, simples et si possible génériques, ainsi que dans les méthodes de proposition des réponses avec notamment le problème de l'évaluation subjective. Enfin la structure de ces questions/réponses est également un point important car il est nécessaire de pouvoir les manipuler facilement et d'avoir les moyens de les choisir par rapport au contexte applicatif pour permettre à l'utilisateur de définir au mieux ces concepts.

Rédaction de questions-réponses

La rédaction des questions et des réponses est une tâche particulièrement délicate. En effet, celles-ci doivent être compréhensibles et accessibles à n'importe quel utilisateur et donc ne pas être trop spécifiques. C'est le **problème du langage**. Le problème du langage est également une difficulté importante dans la création d'un système de questions-réponses où chaque question doit être particulièrement claire et explicite. De plus, les réponses doivent permettre d'éliminer ou de limiter le **problème de la subjectivité** c'est à dire minimiser l'effet de l'affect - ce qui dépend de l'utilisateur, ce qui est personnel à l'utilisateur.

Notion de subjectivité

Le problème rencontré en utilisant un système de questions-réponses est qu'on demande à l'utilisateur de faire l'évaluation d'une donnée et de faire un choix. L'évaluation peut être aisée quand il s'agit de dire si un nombre est plus grand ou plus petit qu'un autre car on dispose d'une unité de mesure et de critères de comparaison. Dans le cas de critères comme le confort, la vitesse, des capacités humaines (managériales, commerciales, etc.), il n'est pas possible d'effectuer directement l'évaluation ou la comparaison, il s'agit du problème de l'évaluation subjective (voir [Grabisch et al., 1997]). Dans notre cas, il s'agit donc de rédiger les questions pour faire en sorte que la comparaison entre deux réponses possibles puissent être faite et ainsi faciliter l'évaluation.
Dans cette thèse, nous n'avons pas abordé cette problématique complexe qui nécessiterait la collaboration avec des spécialistes du langage et de la psychologie cognitive.

5.4 Validation des prototypes

La validation des prototypes est le second processus permettant un dialogue avec l'utilisateur (voir figure 5.9). Dans ce processus, on propose à l'Expert Applicatif les briques concepts qui ont été extraites, c'est à dire des séquences ou des morceaux de séquences contenant le concept tel qu'il a été construit par le système de questions/réponses. Les prototypes ont pour but de valider la définition de la brique-concept et permettent d'affiner celle-ci. L'Expert applicatif décide de valider la brique concept si il trouve effectivement dans celle-ci le concept qu'il a souhaité définir sinon il peut la refuser. L'ensemble des briques validées par l'Expert Applicatif permet de constituer une base d'apprentissage indispensable pour l'indexation automatique. Néanmoins, ces prototypes permettent aussi d'affiner le modèle de briques concepts créé par l'ensemble des réponses données lors de la définition : c'est le processus de correction de la définition.
Une fois les prototypes proposés validés, on peut se retrouver dans plusieurs cas différents comme on peut le voir dans la figure 5.10.

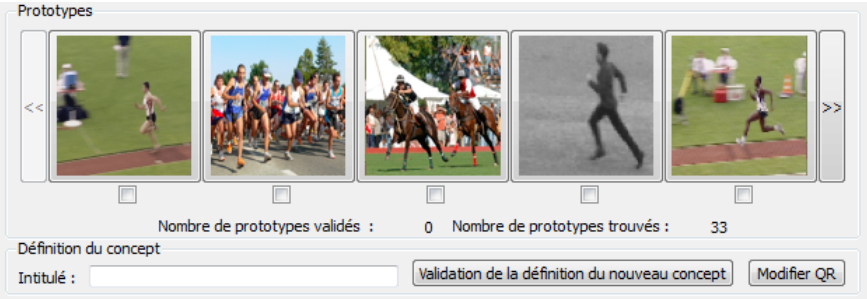


Figure 5.9 — Exemple de proposition du système à l'utilisateur après sa *définition* du concept 'courir'.

Si lors de la validation des prototypes, l'Expert Applicatif sélectionne des prototypes contenant certaines briques en conflit (c'est à dire pour un attribut donné, elles sélectionnent deux propriétés différentes), on se retrouve dans une situation dite d'incohérence. La résolution de ce problème est effectuée en sélectionnant l'union des deux propriétés dans la définition finale. Pour illustrer les différents cas, il convient de préciser que l'on considère qu'un nombre de prototypes minimum pour une définition est de l'ordre de 5 prototypes (environ 10% d'une sélection de prototypes) et que qu'un considère une sélection incohérente si la moitié des attributs est en situation d'incohérence.

- **Cas A** : Les prototypes proposés contiennent le concept et sont validés par l'Expert Applicatif. Le nombre de prototypes est suffisant et ceux-ci sont cohérents. Le processus est donc terminé, on dispose d'une base d'apprentissage (prototypes validés pour les vrais positifs et ceux non validés pour les faux positifs) et d'un modèle de concept adapté.
- **Cas B** : Le nombre de prototypes validés n'est pas suffisant (voir aucun prototype n'est validé) ou les prototypes choisis ne sont pas cohérents. Deux possibilités peuvent être offertes par le système : soit compléter la définition avec une série de nouvelles ques-

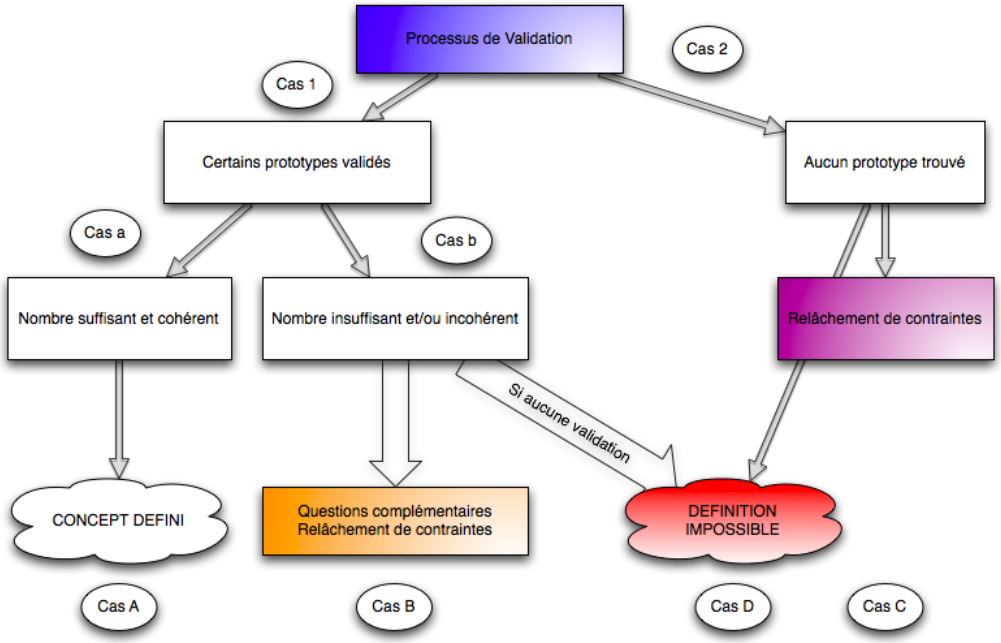


Figure 5.10 — Les différents cas à l'issue de la validation des prototypes.

tions qui permet de corriger ou de confirmer les réponses précédentes, soit relâcher des contraintes sur le choix des modèles de briques, sur le choix de certaines opérateurs,... pour sélectionner des prototypes proches pouvant contenir le concept.

- **Cas C** : Aucun prototype n'est proposé avec le modèle de concept défini par les questions/réponses. Le système peut relâcher certaines contraintes pour proposer de nouveaux prototypes. Si après plusieurs relâchements, aucun prototype n'est sélectionné (cas C) ou aucune n'est validé (cas B) alors le système passe en cas D.
- **Cas D** : Aucun prototype n'est proposé à l'issue de toutes les questions possibles et malgré les relâchements. Le système s'arrête, le modèle de concept n'est pas cohérent avec la base de vidéos ou le système de questions/réponses n'est pas en mesure de construire la définition.

Seuls les cas A et D (concept défini et définition impossible) sont des cas terminaux du système et ont été implémentés. Pour les autres cas, un bouclage de pertinence [Vannoorenberghe, 2004; Belkin, 1980; Baeza-Yates et B.Ribeiro-Neto, 1999; Büttcher *et al.*, 2010] pourrait être mis en œuvre afin d'affiner la définition (questions complémentaires) ou de l'élargir (relâchement de contraintes). Pour des contraintes de temps, ce bouclage n'a pas été introduit dans notre système.

5.5 Correction des définitions

La dernière étape du processus est la correction de la définition effectuée par le système de questions/réponses en utilisant les définitions issues des prototypes. La correction dépend donc du nombre de prototypes validés par l'utilisateur et peut être de plusieurs

types différents :

- ajout de briques dans la définition : on observe que dans tous les prototypes validés (ou une partie), une brique spécifique apparaît. Elle peut donc être ajoutée à la définition
- ajout des informations de séquentialité : lors de la sélection des briques par le système de questions/réponses, on ne dispose d'aucune information sur les briques en œuvre dans la séquentialité. Les prototypes permettent ainsi de définir la séquentialité avec précision.
- suppression de briques dans la définition : on observe que dans tous les prototypes validés (ou une partie), une brique spécifique présente dans la définition n'apparaît pas. Elle pourrait donc être supprimée ou être facultative.

Ajout de briques

A partir des prototypes validés, on effectue une comparaison entre la définition formulée et la définition de chaque prototype validé. Si une brique, non présente dans la définition formulée, apparaît sur toutes les séquences, elle est automatiquement ajoutée à la définition. Dans le cas où elle apparaitrait souvent sans pour autant être systématique, il sera nécessaire de disposer de la possibilité de relâcher les contraintes afin de pouvoir sélectionner les prototypes ne disposant pas de cette brique mais présentant néanmoins le concept sinon, la précision sera bel et bien augmentée mais le rappel fortement diminué.

Ajout d'informations de séquentialité

Dans une définition contenant l'opérateur **'m'**, il est nécessaire d'effectuer la recherche de la séquentialité. Pour cela, on effectue une recherche de séquences sur chaque type de briques contenu dans la définition (compacité, type caméra, nombre de STIP, etc.) pour chacun des prototypes validés. Si une séquence ou plusieurs séquences sont retrouvées sur l'ensemble des prototypes validés alors l'information de séquentialité est ajoutée à la définition.

Suppression de briques

La suppression de briques n'est envisageable que dans le cas d'un relâchement de contraintes où on a accepté la sélection de prototypes ne contenant pas 'intégralement' la définition formulée. Le processus est alors le même que pour l'ajout de briques.

5.6 Structuration et description de la base de données questions/-réponses

5.6.1 Représentation du modèle

Dans le but d'obtenir un modèle permettant de représenter les données i.e. les questions et les réponses mais aussi les liaisons entre les réponses et les briques, entre les réponses et d'autres questions, nous avons choisi d'utiliser une grammaire. Une grammaire est définie comme un ensemble de règles qui régissent le fonctionnement d'une langue. Elle est facilement modifiable, permet de représenter l'ensemble des données et leurs interactions tout en proposant une syntaxe vérifiable et compréhensible. A partir de cette grammaire, on peut générer une représentation des données sous forme d'un arbre XML. Quel que soit le type de structure informatique utilisée ensuite, le fichier XML est un outil particulièrement adapté à l'analyse syntaxique [Michard, 1998].
La première étape est la définition de la grammaire qui s'effectue sous forme d'une DTD (Definition Type Document) qui respecte certains types prédéfinis (tableau 5.2) et certains opérateurs (tableau 5.3).

Type prédéfini	Description
ANY	L'élément peut contenir tout type de données
EMPTY	L'élément ne contient pas de données spécifiques
# PCDATA	L'élément doit contenir une chaîne de caractères

Tableau 5.2 — Les types prédéfinis dans la grammaire.

Opérateur	Signification	Exemple
+	L'élément doit être présent au minimum une fois	A+
*	L'élément peut être présent plusieurs fois (ou aucune)	A*
?	L'élément peut être optionnellement présent	A ?
\| | L'élément A ou l'élément B peuvent être présent | A\|B |
, | L'élément A doit être présent et suivi de l'élément B | A,B |
() | Les parenthèses permettent de regrouper des éléments afin de leur appliquer les autres opérateurs | (A,B)* |

Tableau 5.3 — Les opérateurs disponibles dans la grammaire.

A partir de ces types prédéfinies (tableau 5.2) et de ces opérateurs (tableau 5.3), on souhaite modéliser : la définition des questions/réponses c'est-à-dire le lien entre les questions et les réponses (structure générale), entre les réponses et les questions (navigation), entre les réponses et les caractéristiques (briques et opérateurs).
Par exemple, pour modéliser une question et les réponses, on doit définir un nouvel élément 'QUESTION' comme composé d'un texte, de réponses et d'un état. En particulier, on souhaite qu'il y ait *au moins deux réponses possibles* : on utilise donc le type ELEMENT suivi de 'question' pour la définition de l'élément. On indique ensuite qu'on attend le texte

puis deux réponses '(réponse,réponse)' pour gérer au moins deux réponses. On ajoute une réponse facultative qui peut être présente plusieurs fois ou aucune avec l'opérateur * ('réponse*') permettant de gérer la définition de plus de deux réponses et enfin, l'état. Ce qui nous donne la définition suivante :

$$<!ELEMENT\ question\ texte, (reponse, reponse), reponse*, etat > \tag{5.1}$$

Il faut maintenant définir dans la grammaire les type 'texte', 'reponse' et 'etat'. Le type réponse est structuré d'une manière analogue à question mais texte est un type simple. Il se définit directement avec la chaîne de caractère (s'écrivant # PCDATA).
On effectue ces définitions pour chacun des types nécessaires. Pour notre application, la grammaire définie (définition listing 5.1) pour la représentation de nos questions-réponses est donc la suivante :

```
<?xml version="1.0" encoding='ISO-8859-1'?>
<!-- Definition de la question -->
<!ELEMENT question texte,(reponse,reponse),reponse*,etat >
<!-- Definition de la reponse -->
<!ELEMENT reponse texte,liens*,liensquestions* >
<!-- Definition des autres donnees -->
<!ELEMENT texte (#PCDATA) >
<!ELEMENT liens ID >
<!ELEMENT liensquestion IDquestions,effet* >
<!-- Definition des attributs -->
<!ATTLIST identifiantBrique Attribut (#PCDATA) >
<!ATTLIST identifiantQuestion Attribut (#PCDATA) >
<!ATTLIST effet Attribut (suivante | active | desactive ) >
<!ATTLIST etat Attribut (active | non active | utilise | prioritaire |
   interdite ) >
```

Listing 5.1 — Définition Type Document (DTD) de notre grammaire

A partir de cette grammaire, on peut maintenant représenter l'ensemble de nos données questions/réponses (voir annexes C).

5.6.2 Structuration informatique

Comme les briques du chapitre précédent, les données ont été organisées dans une base de données. Les tables concernant le stockage des données sont au nombre de deux : une contenant les questions (table 'question') et une contenant les réponses (table 'réponse'). La figure 5.11 illustre la représentation de ces deux nouvelles tables et la figure 5.12 illustre la représentation de ces deux tables et leurs relations avec les tables déjà présentées (celles concernant les briques basiques, combinées et les opérateurs de combinaison). Il est important de souligner que la table réponse n'est liée qu'à la table concernant les briques basiques (et pas à celle concernant les briques combinées) et à des opérateurs.

On note que les contraintes données par la grammaire se retrouvent en contraintes de base de données :

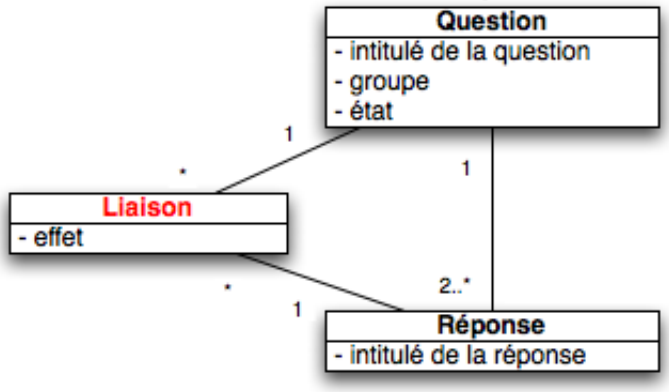


Figure 5.11 — Entités 'question' et 'réponse'

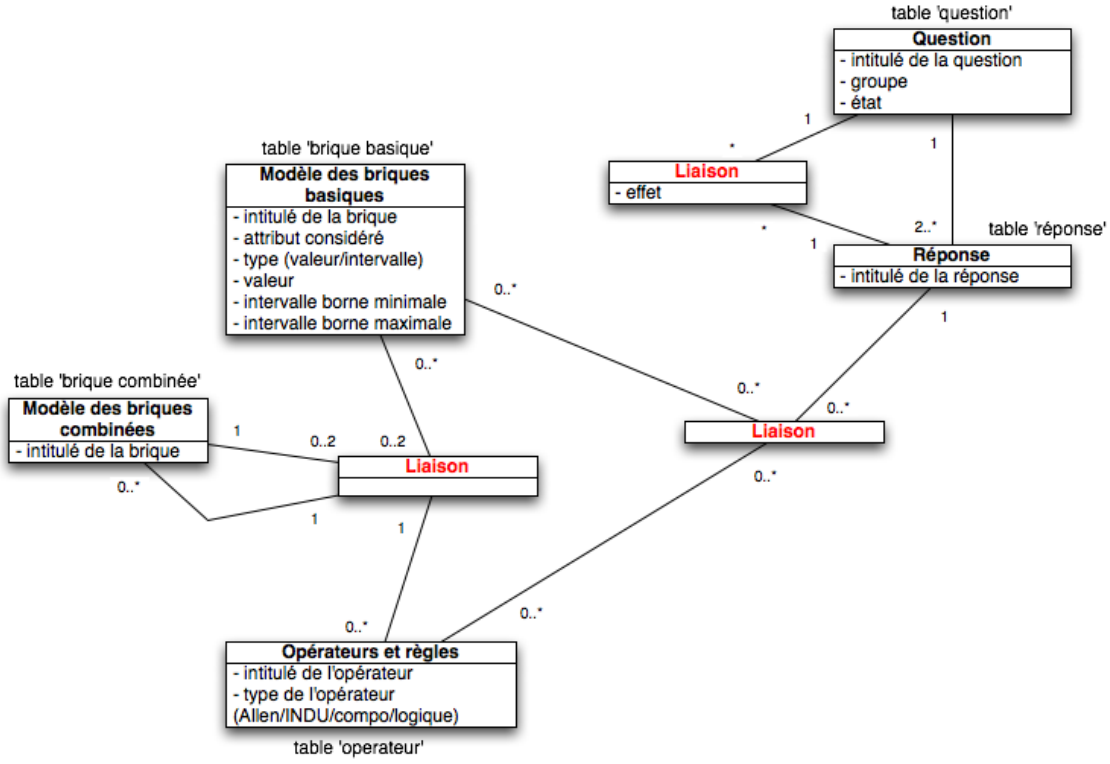


Figure 5.12 — Entité des questions/réponses et les liaisons avec le reste des données.

– une question est liée (composition) à 2 ou plus réponses alors qu'une réponse est liée (composition) à une question ;
– une réponse est liée (sélection) à 0 ou plusieurs briques ;
– une réponse est liée (effet) à 0 ou plusieurs questions.

Ces tables ne sont pas destinées à évoluer à chaque définition d'un nouveau concept

mais uniquement lors des opérations d'insertion ou de suppression de nouvelles questions ou de nouvelles réponses. Pour cela, il faut ajouter ou supprimer dans le fichier de définition en XML la question souhaitée en respectant la grammaire. Le fait d'utiliser une grammaire et un fichier de définition permet également de pouvoir faire évoluer le système de stockage de données vers une autre représentation informatique qu'une base de données. Pour faire évoluer le système vers une autre définition, il suffirait pour cela de définir une nouvelle grammaire et de fournir un nouveau fichier de définition.

5.6.3 Déroulement du processus

A l'état initial, l'ensemble des questions est initialisé (état=accessible) et la question racine est activée (état=activée). Ensuite, à chaque réponse, on effectue une mise à jour de l'ensemble des questions. La question courante passe à utilisée (état=utilisée), les liaisons avec les autres questions sont utilisées pour activer les questions utilisables (état=activée) et pour désactiver les questions devenues obsolètes vis-à-vis de la définition courante (état=désactivée). La dernière étape consiste à sélectionner la prochaine question à utiliser. Si la question utilisée était reliée à une question, la question courante sélectionnée est celle désignée par la liaison. Dans le cas normal, on effectue un tirage aléatoire parmi les questions en état actif.
Ce déroulement séquentiel (question/mise à jour/sélection) s'effectue jusqu'à ce que le système ne puisse plus proposer de questions ou qu'il soit arrêté par l'Expert Applicatif.
Le tableau 5.4 présente les 32 questions du système actuel de questions/réponses et les liens effectifs pour chacune d'entre elles. En réalité, c'est une approximation car les liaisons sont entre les réponses et les briques. Cela représente plus d'une centaine de liaisons. L'ensemble de ces liaisons est donné dans l'annexe C.

Exemple de processus de définition :

1. Quel est le type d'activité ? réponse choisie : déplacement
   Activation des thèmes "Mouvement", "Déplacement" et "Environnement"
2. Y a-t-il un déplacement par rapport au sol ? réponse choisie : oui
   Sélection des briques : caméra (cam-mobile) - translation
3. Y a-t-il un déplacement dans l'environnement ? réponse choisie : oui
   Sélection des briques : caméra (cam-mobile) - orientation possible (fo-$*$)
4. Est-ce un mouvement régulier, ponctuel, unique, séquentiel ? séquentiel
   Sélection des briques : opérateur de séquentialité (m)
5. Est-ce que l'activité peut être décomposée en plusieurs actions ? réponse choisie : non
   Sélection des briques : /
6. Quelle est la rapidité/l'intensité de l'action ? réponse choisie : plutôt rapide
   Sélection des briques : flot optique (fo-io-moyen)
7. Le mouvement est-il localisé à une partie du corps en particulier ? réponse choisie : oui, plutôt
   Activation et sélection de la question "Plutôt antérieure ? postérieure ?" comme question suivante.

8. Plutôt antérieure ? postérieure ? réponse choisie : antérieure
   Sélection des briques : STIP par cadran (stip-qo-fort cadrans bas)
9. Le concept concerne-t-il un (ou des) objet(s) d'intérêt ? réponse choisie : 1
   Sélection des briques : nombre d'objets = 1 (no-1)
10. Y a-t-il déformation pendant le mouvement ? réponse : oui
    Sélection des briques : compacité (c-f/c-m/c-F)
11. Peut-on prédire la suite du mouvement ? réponse : oui
    Sélection des briques : durée
12. Est-ce en intérieur ou en extérieur ? réponse : indifférent
    Sélection des briques : /
13. Y a-t-il un environnement particulier ? réponse : non
    Sélection des briques : /

En utilisant le système de questions/réponses proposé, une première définition de la course à pied obtenue est :

$$\{c-f \ \oplus c-m \ \oplus c-F\,, \mathbf{m}\ \}\ \ \mathbf{d}\ no-1\ \mathbf{d}\ fo-io-moyen\ \mathbf{d}\ fo-*^{1} \tag{5.2}$$

Ce qui peut être traduit par : présence des briques de compacité (c-f/c-m/c-F) de manière séquentielle (opérateur 'm') tout en ayant un seul objet en mouvement (no-1), une intensité du flot optique plutôt rapide (fo-io-moyen) et une orientation de celui-ci quelconque (fo-$*$) mais plutôt unique (translation ^{1}).

5.7 Perspectives d'amélioration

La première perspective est l'amélioration des questions et surtout des liens entre réponses et briques. Un certain nombre de briques ne sont pas directement liées aux questions/réponses alors que d'autres sont beaucoup utilisées.
Le dispositif décrit dans ce chapitre est séquentiel sans rebouclage par rapport aux actions de l'utilisateur. Les perspectives d'amélioration se situent donc essentiellement à ce niveau : prendre en compte immédiatement toute intervention de l'Expert Applicatif afin d'améliorer les performances du système.

5.7.1 Évolution dynamique

Comme on l'a indiqué en introduction, la définition d'un concept n'est pas figée, elle peut évoluer dans le temps soit parce que l'application évolue ou parce que l'expert a évolué ou est différent. Afin de gérer cette évolution, il est nécessaire d'introduire un mécanisme d'évolution dynamique. Après la définition d'un nouveau concept, il semble judicieux de faire une phase d'ajustement des concepts déjà définis, surtout si la définition du nouveau concept est proche d'une définition déjà existante.
On peut distinguer plusieurs niveaux de rebouclage :

- au cours des questions/réponses ;
- à l'issu des questions/réponses ;

Questions	Liens avec les briques
Quel est le type d'activité	navigation
Combien de personnages sont concernés	nombre d'objets
Y a-t-il des spectateurs ? des observateurs ?	STIP
Y a-t-il un environnement particulier ?	navigation
Est-ce en intérieur ou en extérieur ?	luminance
Est-ce plutôt le jour ou la nuit ?	luminance
Est-ce plutôt au vert (campagne, nature, gazon, herbe) ?	couleur
Est-ce plutôt une activité nautique ? aérienne ?	couleur/SIP
Est-ce une activité de montagne ? à la neige ?	couleur/SIP
Est-ce plutôt en ville ? en stade ? équipement sportif ?	couleur/SIP/hough
Est-ce une activité sportive ?	navigation
Est-ce une activité courte ou longue ?	durée
Est-ce que l'activité peut être décomposée en plusieurs actions ?	navigation
En combien peut-on la décomposer ?	STIP
Peut-on prédire la suite du mouvement ?	STIP/durée
Peut-on savoir quand le mouvement va se terminer (prédiction) ?	durée
Peut-on savoir quand le mouvement est terminé (constat) ?	durée
Quelle est la rapidité/l'intensité de l'action ?	FO/STIP
Est-ce un mouvement régulier, ponctuel, unique, séquentiel ?	STIP
Est-ce un dialogue, une manifestation, une réunion ?	SIP/STIP
Y a-t-il besoin d'un objet spécifique ?	navigation
Le mouvement est-il localisé à une partie du corps en particulier ?	navigation
Plutôt antérieure ? postérieure ?	STIP
L'objet principal est-il un personnage ?	navigation
Y a-t-il un déplacement dans l'environnement ?	caméra/FO
Y a-t-il un déplacement par rapport au sol ?	caméra/FO/STIP
Le concept concerne-t-il un (ou des) objet(s) d'intérêt ?	nombre d'objets
Concerne-t-il un objet principal ?	nombre d'objets
Y a-t-il déformation pendant le mouvement ?	taille/compacité/STIP
Y a-t-il des changements dans le mouvement ?	compacité/STIP
Le personnage se lève-t-il ?	changement
Utilise-t-on un moyen de locomotion ?	FO

Tableau 5.4 **—** Liste des questions et leurs liaisons avec les briques.

– à l'issu de la validation des prototypes pour la réadaptation du modèle du système de questions/réponses.

5.7.2 Mesures d'incertitude ou de qualité

A l'issu des processus de questions/réponses et d'extraction, on dispose d'un certain nombre de prototypes (briques concepts). Il serait intéressant de définir un indicateur que

l'on pourrait associer aux prototypes et qui permettrait de les classer et de les proposer à l'utilisateur par ordre de pertinence.
Les mesures d'incertitude et de qualité [Azé, 2003] permettent de diminuer le nombre de prototypes qui vont être visionnés par l'utilisateur lors de la validation en lui montrant en priorité les prototypes qui semblent être le plus pertinent.
Il est nécessaire de définir des méthodes calculant des mesures d'incertitude c'est à dire des valeurs indiquant l'évaluation de la qualité des prototypes présentés par rapport à la définition formulée.
On peut distinguer deux types de mesures d'incertitude :

- *Les mesures d'incertitudes statiques* : ce sont les mesures effectuées dès la recherche de prototype. Il s'agit d'effectuer la comparaison entre la définition produite et la définition trouvée dans chaque prototype. Elle est effectuée lors de la recherche.
- *Les mesures d'incertitude dynamiques* : ce sont les mesures effectuées lors de la validation des prototypes. A chacune des validations effectuées, la définition de chaque prototype peut servir de référence d'évaluation pour les autres prototypes. Ainsi, on réévalue dynamiquement les autres prototypes à chacune des validations. Il s'agit donc d'un rebouclage.

5.7.3 Évaluation dynamique

Actuellement, le système de questions/réponses s'arrête automatiquement quand il n'a plus de questions à poser, généralement après une douzaine à une petite vingtaine de questions. La définition peut être telle que le nombre de prototypes correspondant au concept que l'on souhaite définir est très faible voir nul.
L'évaluation dynamique est un processus permettant d'apporter en temps réel une information à l'Expert Applicatif aidant à prendre la décision de continuer ou d'arrêter le processus tout en optimisant les chances d'obtenir un nombre de prototypes suffisant. Initialement, toutes les briques sont élligibles pour le concept ce qui entraîne que la quantité de prototypes valides est maximale. Puis, au fil des questions, la quantité de briques élligibles diminue et le nombre de prototypes correspondant également.
Ainsi, l'évaluation dynamique consiste à indiquer à chaque réponse de l'Expert Applicatif, une évaluation du nombre de prototypes extraits correspondant à la définition du concept en cours de définition. De cette manière, si celui-ci ne souhaite pas continuer le processus mais visualiser les prototypes, il peut l'indiquer.
Cette démarche d'évaluation dynamique peut être assimilée à la possibilité offerte par certains systèmes experts d'expliciter leurs raisonnements, à la fois à des fins de vérification et de formation. Ceci peut être particulièrement intéressant pour optimiser le fonctionnement du processus en fonction du niveau de compréhension de l'utilisateur.

5.7.4 Insertion de questions

La quantité de questions/réponses est une problématique assez importante. Elle garantit la représentativité de choix possibles c'est à dire le nombre de définitions possibles. Le

système a initialement été construit avec un certain nombre de questions qui ne sera pas suffisant pour différencier deux concepts plus ou moins proches. Il peut donc être intéressant d'insérer de nouvelles questions dans le système.
Techniquement, l'insertion d'une nouvelle question ne pose pas de difficulté puisque le modèle choisi le permet facilement. Il s'agit plutôt de la définition des liaisons entre les réponses liées à cette question et les briques/opérateurs à choisir sans faire intervenir de nouveau un expert en traitement d'images comme cela a été fait initialement. Pour cela, il sera nécessaire d'effectuer une période d'apprentissage sur cette nouvelle question. Lors de la définition d'un concept, la nouvelle question sera introduite aléatoirement (dans le groupe lui correspondant). Elle n'apportera aucune information puisqu'elle n'est pas encore définie mais permettra quand on l'aura utilisée suffisamment de fois, d'effectuer une liaison avec une ou plusieurs briques en utilisant des méthodes d'apprentissage.
Le système a également été construit autour d'un certain nombre de caractéristiques et il est tout à fait envisageable que celles-ci évoluent notamment avec l'ajout de nouvelles caractéristiques. Si on ne peut pas les lier directement à des questions existantes, il faudra également en insérer une ou plusieurs nouvelles.
L'insertion de nouvelles questions peut-être effectuée par n'importe qui, y compris par l'Expert Applicatif. L'insertion de caractéristiques est obligatoirement réalisée par l'Expert en Traitement d'Images.

5.7.5 Suppression de questions

Dans le cas où une question est inaccessible (aucune chemin dans l'arbre ne permet de l'activer), dans le cas où elle n'apporte aucune information ou si elle apporte une information systématiquement redondante, on peut se demander si la question reste intéressante et si il ne serait pas souhaitable de la supprimer de l'ensemble des questions. Attention, ce n'est pas le cas d'une question jamais utilisée qui n'a jamais été utile dans la définition de concepts mais qui pourrait l'être lors d'une prochaine définition.
Le mécanisme de suppression de question est un mécanisme automatique qui supprime les questions inutiles. L'évaluation de l'apport d'information est effectuée en comparant la définition initiale c'est à dire la définition à l'issu du processus de questions/réponses et la définition finale c'est à dire la définition issue de la définition initiale corrigée par l'information provenant des prototypes sélectionnés. Si la comparaison montre qu'une brique sélectionnée initialement n'apparaît pas finalement, soit la réponse ayant amenée cette brique est erronée soit la question était inutile. L'étude fréquentielle de ce cas sur une question donnée permet de déterminer le niveau informatif d'une réponse puis d'une question.
La redondance est détectée à la fin du processus de questions/réponses. Si on dispose de plusieurs réponses liées à la même brique, une au moins est redondante. L'étude fréquentielle permet de déterminer la ou les questions qui n'apportent aucune information supplémentaire. Ce mécanisme est à évaluer car la redondance permet également de gérer le cas de réponses erronées fournies par l'utilisateur.

5.7.6 Modification des liaisons des QR et des briques

On peut identifier deux manières d'effectuer des modifications des liaisons entre les questions/réponses et les briques. Soit manuellement, soit par la mise en œuvre du mécanisme d'apprentissage de paramètres similaire à celui permettant l'insertion d'une nouvelle question. La correction d'une liaison sur laquelle les résultats ne semblent pas pertinents peut être effectuée en lançant la méthode d'apprentissage de liaison.

6 Performances et évaluation du système

« Tout avantage a ses inconvénients et réciproquement. »

Jacques Rouxel

Le modèle de briques défini dans le chapitre 4 a été utilisé dans le système de questions/-réponses proposé dans le chapitre 5. Ces travaux ont donné lieu au développement de deux prototype logiciels : le logiciel 'Features eXtraction' FX permettant l'extraction des caractéristiques et le stockage des briques basiques correspondantes, et le logiciel 'BRIk Knowledge manager' BRIK permettant la définition de concepts ainsi que la recherche en utilisant un système de questions/réponses dont la structure est décrite section 6.1. Nous avons mis au point puis testé ces deux prototypes à l'aide d'un certain nombre de vidéos appartenant à des bases d'origines diverses, et ayant trait au mouvement et plus particulièrement à celui d'un personnage. La description de ces différentes bases est donnée section 6.2. Enfin, dans la section 6.3, nous proposons d'évaluer notre approche à l'aide de quelques mesures obtenues sur les bases de vidéos. Au niveau des points d'intérêt, la description des tests et les résultats ont déjà été présentés au chapitre 3. Les tests présentés dans ce chapitre concernent plus précisément l'extraction des briques et le temps d'exécution des algorithmes, la capacité d'extraire des prototypes à partir de modèles de concepts, et enfin une estimation de la pertinence des Questions/Réponses en relation avec une base de données de vidéos.

6.1 Prototypes développés

Dans le cadre de ces travaux de thèse, deux prototypes ont été développés, l'un concernant l'extraction de briques intitulé 'Features eXtraction' et le deuxième permettant d'extraire des prototypes de concept à partir du modèle obtenu à partir du jeu de Questions/Réponses. Ce deuxième logiciel est appelé 'BRIk Knowledge manager'.

Les deux logiciels ont été développés sous Eclipse Helios [1], en C++ en utilisant le compi-

1. http ://www.eclipse.org/

lateur MinGW [2]. La librairie Intel OpenCV [3] nous a fourni un certain nombre d'outils pour le traitement d'images et la libraire Nokia Trolltech Qt [4] nous a permis de construire les interfaces graphiques. La base de données a été créée sous Microsoft Access. Enfin, le module Motion2D [5]a été intégré au logiciel afin d'effectuer les compensations de mouvement de caméra.

Les tests ont été réalisés sur deux machines différentes : un portable équipé d'un Intel Core 2 Duo et une machine de calcul équipée d'un Intel Quad Core. Ces deux plateformes fonctionnent actuellement sous Microsoft Windows Seven 32 bits mais ont été conçues avec des outils Open Source compatibles avec les noyaux Unix (Linux et Mac OS).

6.1.1 Logiciel FX - Features eXtraction

Le logiciel FX permet l'analyse de séquences vidéo, l'extraction d'informations et la structuration en briques basiques. Les caractéristiques extraites sont les suivantes :

- points d'intérêt spatiaux : méthode de Jianbo Shi [Shi et Tomasi, 1994] (amélioration du détecteur de Harris)
- points d'intérêt spatio-temporels : méthode de Ivan Laptev [Laptev et Lindeberg, 2003]
- flot optique : méthode de Lucas-Kanade [Lucas et Kanade, 1981a]
- Hough : méthode de Duda [Duda et Hart, 1972]
- couleurs dominantes : méthode de Ravishankar [Ravishankar *et al.*, 1999]

Le logiciel effectue également les traitements suivants :

- compensation de mouvement : Motion 2D de Jean-Marc Odobez [Odobez et Bouthemy, 1995]
- extraction de fond : méthode de Giai-Checa [Giai-Checa *et al.*, 1993]
- calcul du niveau d'activité : méthode de Robert Laganière [Laganière *et al.*, 2008]

Ces attributs sont ensuite utilisés pour construire des caractéristiques plus élaborées.

Le modèle utilisé

L'application a été conçue selon une architecture de type *Modèle-Vue-Contrôleur* (MVC) qui consiste à séparer les données (le modèle de données, le traitement des données et la récupération des données), la présentation (interface utilisateur) et la gestion (logique de contrôle, organisation, synchronisation et gestion des évènements) en organisant l'application en trois parties distinctes :

- la *vue* (classe "concept" dans le schéma UML figure 6.1) permet le pilotage de l'interface et les interactions avec l'utilisateur. Elle a deux tâches principales qui sont de présenter des résultats fournis par le modèle et de recevoir les actions de l'utilisateur (clics souris, boutons, saisie clavier). Ces évènements sont envoyés au *contrôleur* (classe "controler" dans le schéma UML). Dans notre application, la *vue* est chargée de récupérer des informations auprès de l'utilisateur c'est à dire les réglages des paramètres

2. http ://www.mingw.org/
3. http ://opencv.willowgarage.com/wiki/
4. http ://qt.nokia.com/products/
5. http ://www.irisa.fr/vista/Motion2D/

et d'afficher des résultats, en l'occurrence, les images traitées et les caractéristiques calculées.
- le *modèle* représente le comportement de l'application : il gère les données, il effectue les traitements et interagit avec les bases de données (insertion, récupération, mise à jour) tout en garantissant l'intégrité de celles-ci. Il peut être décomposé en plusieurs modèles dédiés. Dans notre application, le *modèle* est décomposé en 3 sous-modèles : un sous-modèle effectuant les opérations de traitement d'images, un autre se chargeant de la gestion des Entrées/Sorties pour les fichiers vidéo (lecture et enregistrement) et un troisième dédié à la fusion pour le calcul des caractéristiques. A ces sous-modèles s'ajoute un module assurant la communication avec la base de données.
- le *contrôleur* permet la coordination de l'ensemble du système en prenant en charge la gestion des évènements, la mise à jour de la *vue*, la synchronisation avec le *modèle*. Il analyse les demandes de la *vue*, demande le traitement adéquat au *modèle* avant de renvoyer à la *vue* le résultat de cette demande. Le *contrôleur* ne modifie jamais la *vue*, les données et n'effectue aucun traitement. Dans notre application, il se charge de passer les paramètres au début d'un traitement de la *vue* aux *modèles*, de renvoyer les résultats du *modèle* à la *vue* et de demander les mises à jour en base de données. Le *contrôleur* est le chef d'orchestre de l'application.

Cette organisation offre un cadre intéressant pour structurer l'application en permettant la mise à jour des composants de manière indépendante. On peut aisément modifier l'interface (la vue) ou le système de base de données en modifiant une partie limitée du système sans impacter les autres composants. Ainsi la maintenance et l'amélioration sont facilitées.

Architecture générale

Le schéma UML représenté figure 6.1 présente l'architecture globale du système et les différentes classes. Chaque classe comporte un certain nombre de propriétés et des opérations accessibles.

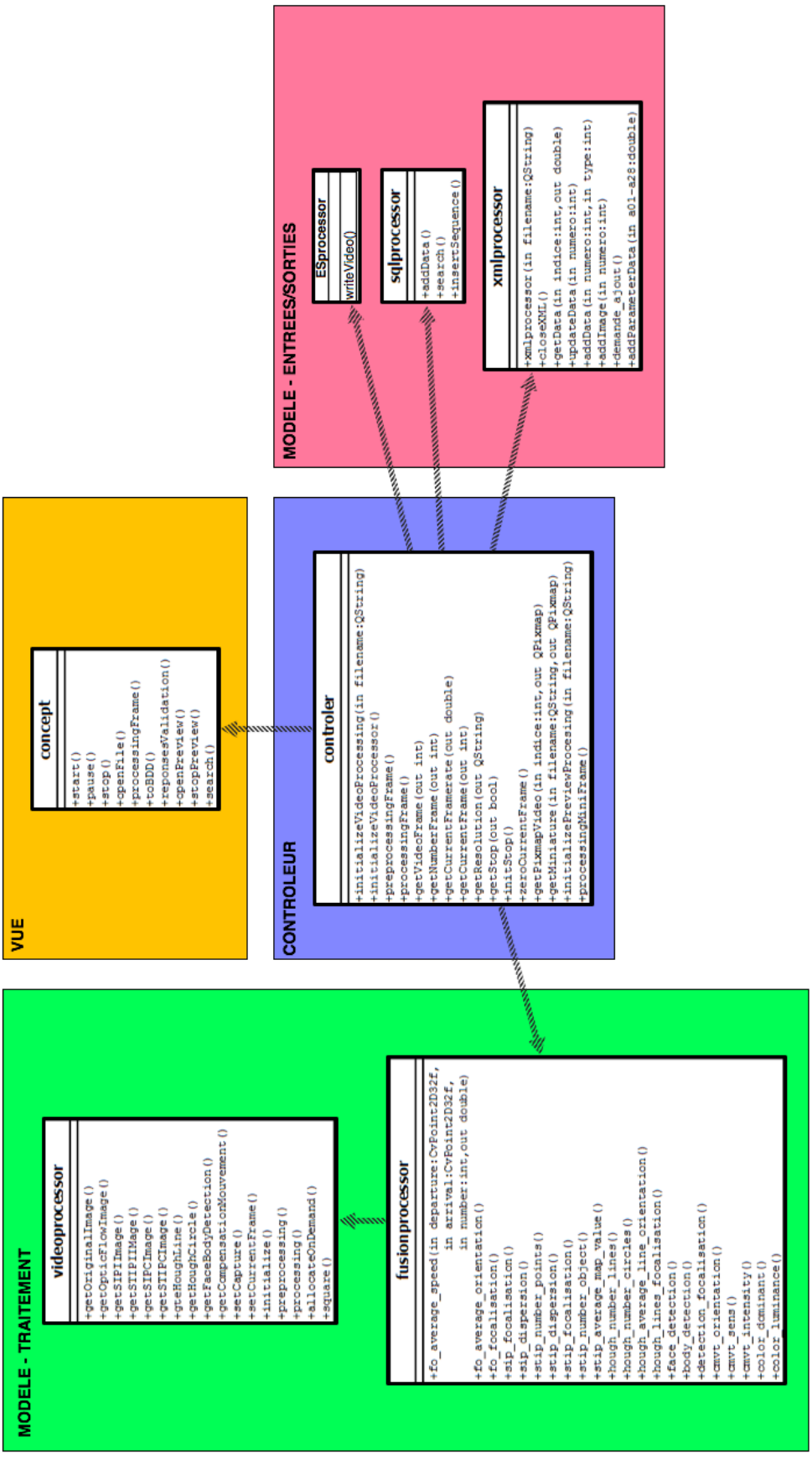


Figure 6.1 — Schéma de l'application Features eXtraction (FX) : le module "concept" correspond à la vue, le module "controler" coordonne l'ensemble du système et les différents modules "processor" correspondant au "modèle" permettent d'effectuer les différents traitements et entrées/sorties.

Les classes 'processor' correspondant au *modèle* sont destinées à effectuer tous les traitements :

- 'sql' pour gérer les bases de données : permet l'insertion, la consultation et la mise à jour des données. Ce module permet également la création des requêtes dynamiques présentées dans le chapitre 4.5.1.2.
- 'xml' pour gérer les documents XML : permet la gestion des entrées/sorties sur les fichiers XML. Ce module gère l'importation et l'exportation de profils de paramétrage de l'application.
- 'video' pour effectuer les traitements sur les vidéos : permet l'extraction des différentes caractéristiques référencées pour chaque séquence, chaque image et chaque objet.
- 'fusion' pour effectuer les calculs sur la fusion de caractéristiques : il permet le calcul des caractéristiques présentées à la fin du chapitre 3.
- 'ES' pour effectuer les lectures et les écritures sur les fichiers vidéos.

Certaines classes secondaires ne sont pas représentées dans ce schéma simplifié. La classe 'parameter' permet de paramétrer les algorithmes de traitements (le paramétrage des différents algorithmes d'extraction est disponible dans l'annexe D) et la classe 'tools' regroupe les outils de calculs génériques comme le calcul de distance entre deux points dans un plan. Ces deux classes sont communes et génériques. Enfin, les classes 'Gaussian', 'TempoFilter' sont des classes utilitaires pour l'extraction des STIP.

Cycle de fonctionnement du contrôleur

Le cycle du contrôleur est représenté figure 6.2 : une fois initialisés, les algorithmes de traitements (opérations de traitement d'images et de calcul de données), les entrées/sorties (séquences d'images à lire, fichiers vidéos à écrire, base de données à remplir), le contrôleur rentre dans une boucle opérationnelle dont il ne pourra sortir que par l'action "sortir" de la vue ou par la fin du traitement de la séquence d'images.
Pour chaque image de la séquence d'images à traiter, 'videprocessor' récupère l'image courante puis lance chacun des algorithmes d'extraction de caractéristiques. Une fois les algorithmes exécutés, le contrôleur envoie les résultats à 'fusionprocessor' qui se charge d'effectuer tous les calculs pour obtenir les caractéristiques. Une fois les calculs effectués, le contrôleur envoie ses résultats dans 'sqlprocessor' afin qu'ils soient stockés en base de données. Les images issues de l'application des algorithmes de traitement d'images du 'videoprocessor' sont envoyées à 'ESprocessor' qui écrit les données dans des fichiers vidéos avant d'être envoyées avec tous les résultats du 'fusionprocessor' vers la "vue" qui est mise à jour.

Utilisation du système

Un lancement par script a été écrit afin de permettre le lancement du processus sur un cluster de calcul. Il permet la suppression de la "vue" pour effectuer uniquement les traitements. L'appel de ce script est présenté au listing 6.1. Dans sa version simplifiée, seul le nom du fichier vidéo à traité est spécifié et les paramètres utilisés sont ceux par défaut. Les para-

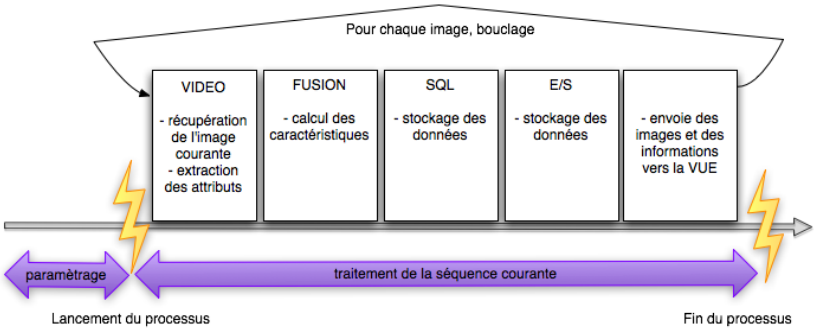


Figure 6.2 — Cycle de fonctionnement du contrôleur.

mètres entre crochets permettent de modifier la configuration des algorithmes. L'ensemble des paramètres est décrit dans l'annexe D. Le code source, le programme et le manuel d'utilisation sont disponibles sur SourceForge via http ://sourceforge.net/projects/fextraction/.

```
./concept.exe {fichier_video} [-out nom_fichier -sipsigma valeur -
    sipthreshold valeur....]
```

Listing 6.1 — Appel de FX

L'interface graphique

Pour faciliter le paramétrage, ainsi que pour l'évaluation visuelle des traitements, une interface graphique présentant les vidéos résultats des traitements a été développée (figure 6.3). On distingue trois zones :

- une zone d'affichage d'images, située à gauche et formée par 6 images correspondant aux différents algorithmes d'extraction d'attributs : la compensation de mouvement en haut à gauche, le masque des objets en mouvement en haut à droite, le flot optique calculé sur les SIP au milieu à gauche, les lignes caractéristiques au milieu à droite, les SIP en bas à gauche et les STIP en bas à droite. Le choix des données affichées est modifiable via un sélecteur situé en bas à droite de l'interface.
- une zone de paramétrage avec un panneau muni de plusieurs volets en haut à droite. Chacun de ces volets se déplie pour permettre la saisie des différents paramètres. Les différents volets sont affichés dans la figure 6.4. Une fois un traitement lancé, cette zone n'est plus accessible.
- une zone d'affichage de résultats en bas à droite qui permet d'afficher les caractéristiques extraites et calculées ainsi que la vitesse de traitement.

Via cette interface, il est nécessaire de charger les séquences une à une pour les traiter et extraire les informations correspondantes. L'insertion des informations en base de données est automatique.

L'interface est dotée d'un panneau de raccourcis (figure 6.5) permettant dans l'ordre : d'ouvrir une séquence, supprimer le traitement courant et réinitialiser l'interface, lancer ou re-

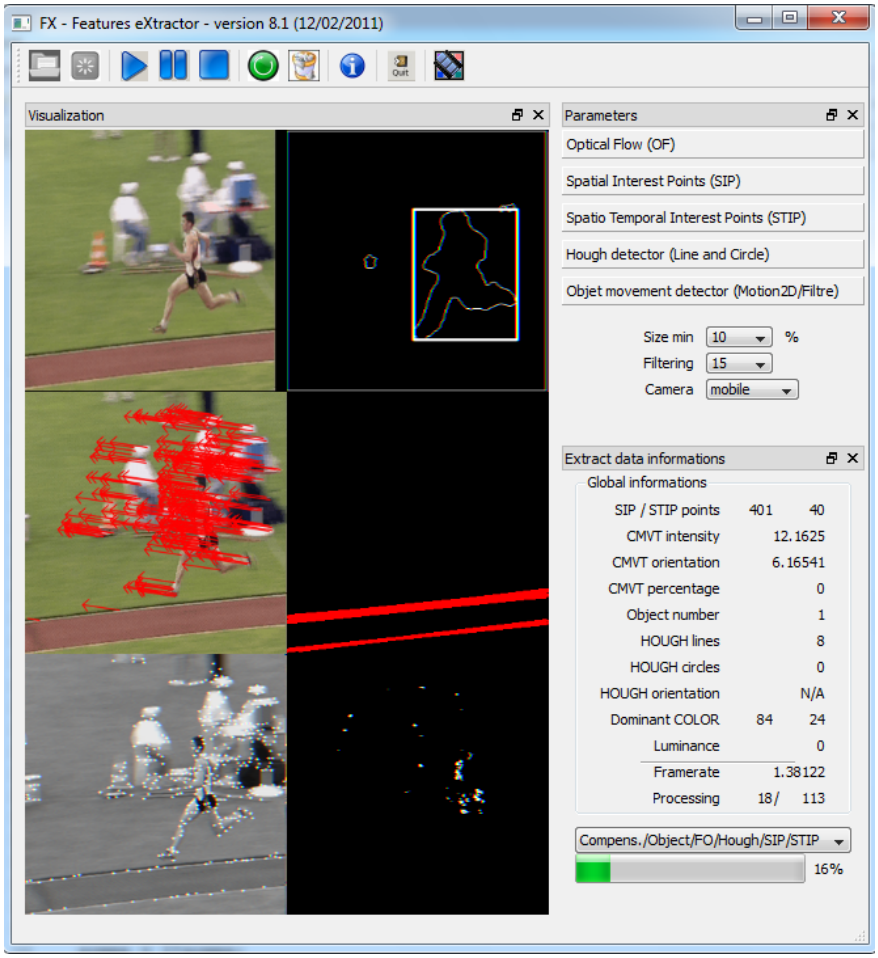


Figure 6.3 — Interface complète du logiciel FX : les vidéos résultats (compensation de mouvement, masque de mouvement, flot optique, lignes caractéristiques, SIP et STIP) dans la colonne de gauche, les volets de paramétrages en haut à droite et le panneau d'affichage des résultats en bas à droite.

lancer un traitement avec les paramètres en cours, mettre en pause, arrêter le traitement en cours, retirer le dernier traitement de la base de données, vider la base de données, obtenir les informations sur le système, quitter le système et générer les briques à partir de la base de données actuelle.

Améliorations

Des améliorations peuvent être ajoutées au logiciel FX - Features eXtraction. A court terme, il s'agit d'ajouter de nouvelles caractéristiques en incluant dans le système de nouveaux extracteurs comme notamment les 'Scale Invarian Features Transform' (SIFT), les 'Histogrammes des Gradients' (HoG), etc. mais aussi de permettre l'utilisation de l'outil pour extraire des caractéristiques sans nécessairement les mettre dans la base de données. Dans le but que le système puisse être utilisé indépendamment de la base de données, il est

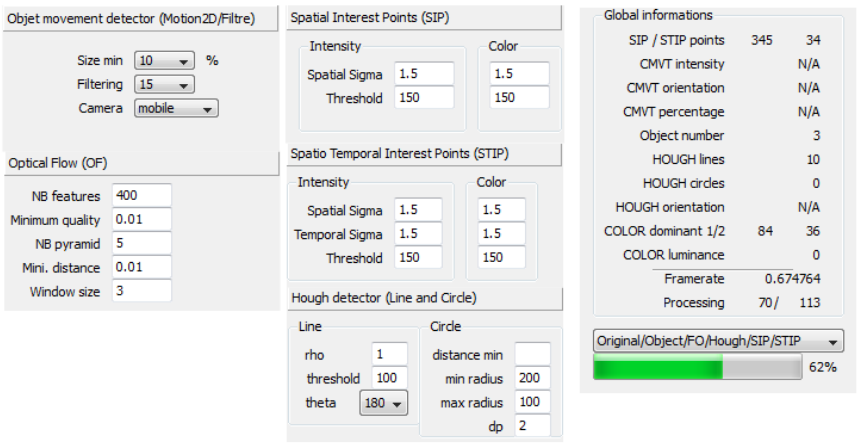


Figure 6.4 — Interface de paramétrage des extracteurs et de visualisation des informations. Cette vue est le développement des volets de paramétrages situés en haut à droite de l'interface globale.

Figure 6.5 — Panneau des raccourcis de l'interface.

également prévu d'ajouter le support des fichiers XML afin d'avoir un outil d'extraction des données dans une formulation utilisable par d'autres systèmes.
A plus long terme, ce logiciel pourrait bénéficier des technologies récentes de parallélisation particulièrement bien adaptées pour le traitement d'images et l'extraction de caractéristiques. Une perspective intéressante est donc une modification du logiciel destinée à ajouter le support du traitement parallèle comme NVidia CUDA. L'usage du traitement parallèle CUDA permet des améliorations d'un facteur de l'ordre de 100 sur du traitement d'images. Ainsi, par exemple, le traitement (extraction des caractéristiques) de la base des 10000 plans s'effectueraient en 17 minutes au lieu de 28 heures (voir le paragraphe 6.3.1). De cette manière, l'ajout de nouveaux extracteurs augmenterait le temps de calcul mais celui-ci pourrait être compensé en ajoutant des unités de calcul parallèle.

6.1.2 Logiciel BRIK - BRIk Knowledge management

Le logiciel "BRIK" est le système qui permet de construire la définition d'un concept et d'extraire des prototypes dans la base de vidéos. Il est constitué du système de Questions/Réponses exposé dans le chapitre 5, du système de visualisation et de validation de séquences 'prototypes' et de définition du concept final.

Elaboration du système

Nous avons construit un système de questions/réponses composé de 30 questions clas-

sées par thèmes et organisées sous forme arborescente. Chacune d'elles donnant lieu à 2, 3 ou plusieurs réponses. Le choix des réponses laissé libre, ainsi on peut sélectionner aucune, une, plusieurs ou toutes les réponses proposées. Chaque réponse permet la sélection de briques et/ou d'opérateurs.

Architecture générale

Le schéma UML représenté figure 6.6 présente l'architecture générale du système. La classe 'brik' effectue la gestion de l'interface et la gestion du processus. La classe 'sql' permet de communiquer avec la base de données et 'lectureVideo' se charge de la lecture de séquences d'images.

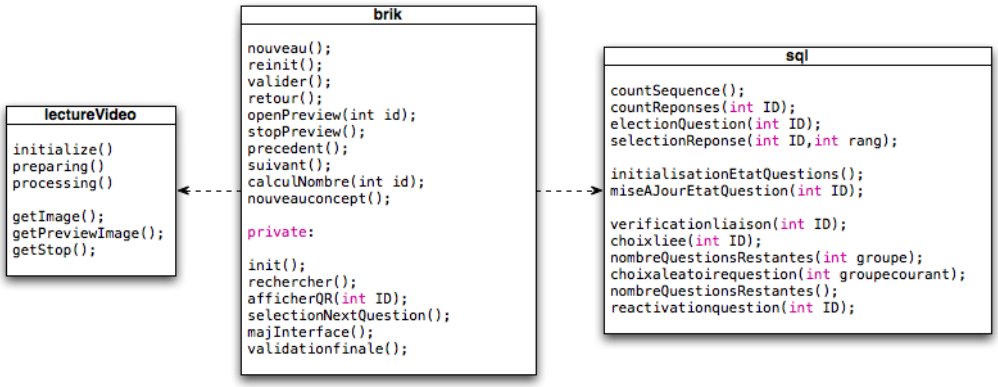


Figure 6.6 — Schéma UML du logiciel BRIK : la classe 'brik' jouant le rôle de vue et de controler, la classe 'lectureVideo' étant une sous-vue et la classe 'sql' étant un modèle.

Déroulement du processus

La définition d'un concept via l'interface BRIK est composée de trois phases :

- la phase de dialogue avec l'utilisateur (figure 6.7) : le système sélectionne une question parmi celles disponibles, utilisables et accessibles en fonction du contexte ainsi que les réponses associées à proposer. Après avoir sélectionné la ou les réponses voulues, la validation met à jour le contexte et fait boucler le système sur la sélection d'une nouvelle question jusqu'à ne plus avoir de questions à poser. Le système lance alors la phase suivante.
- la phase de recherche de séquences 'prototypes' : à partir de l'ensemble des réponses données et des liens entre réponses et briques basiques/opérateurs, le système construit une définition du concept comme étant une combinaison de briques extraites par le système FX. Cette définition permet ensuite la génération d'une requête pour rechercher les briques correspondantes dans la base et les séquences d'images correspondantes.

– la phase de visualisation et de sélection des prototypes (figure 6.8) : les prototypes des séquences d'images sont proposés à l'utilisateur qui peut les visionner pour vérifier la correspondance du contenu avec la définition qu'il a effectuée lors de la phase de Question/Réponse. Quand le contenu correspond, il peut valider ce prototype. Une fois qu'il a visionné les prototypes et validé ceux qui étaient intéressants, il peut donner un nom au concept et le valider. Cette validation permet de stocker la définition (modèle de briques défini), le nom du concept et les prototypes en base de données.

A chaque définition d'un concept, il est nécessaire d'effectuer l'ensemble des ces trois phases.

Interface

Le logiciel est constitué d'une zone principale qui varie en fonction des étapes. Pendant l'étape de QR, elle permet l'affichage des questions, des boutons réponses et de validation/retour en arrière (figure 6.7). Ensuite, lors de l'étape de visionnage et de validation des prototypes, elle affiche une liste horizontale de miniatures des séquences sélectionnées (figure 6.8). Enfin lorsqu'il y a un clic sur une miniature, un lecteur est ouvert et la séquence d'images est lue. Un nouveau clic sur le lecteur interrompt celui-ci et le supprime.

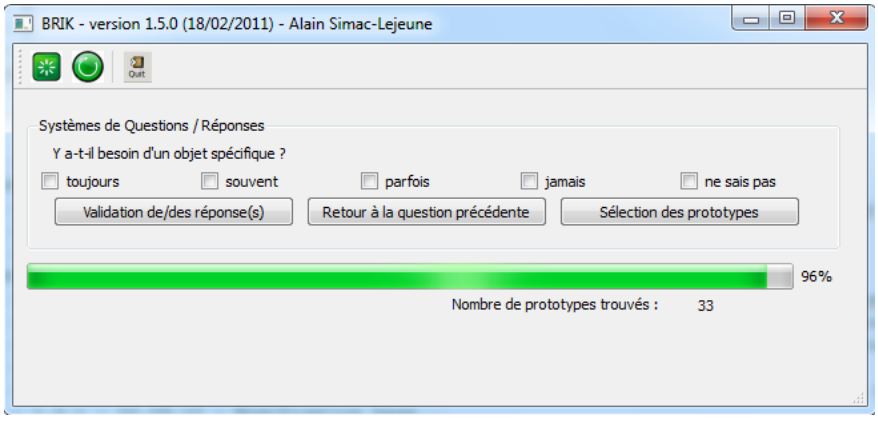


Figure 6.7 — Interface du système de QR.

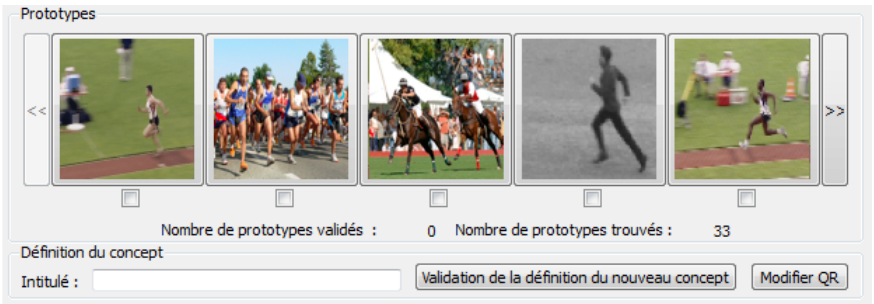


Figure 6.8 — Interface de validation des prototypes.

L'ensemble du code source, du programme et le manuel d'utilisation sont disponibles sur SourceForge via http ://sourceforge.net/projects/brik/

Améliorations

Une migration du logiciel est envisagée sous une forme web permettant de l'utiliser de manière collaborative. Il s'agit de faire migrer le logiciel C++/Qt et la base Microsoft Access dans une version HTML/CSS/PHP en utilisant une base de données SQL et un serveur Apache. La lecture de vidéo serait effectuée via HTML5 en effectuant des transcodages à la volée par un serveur Red5.

L'objectif est de rendre le système collaboratif c'est à dire rendre l'outil disponible à tous et permettre l'utilisation par une multitude d'utilisateurs. Les définitions effectuées successivement par tous ces utilisateurs permettraient l'obtention d'informations qui rendraient fonctionnels les différents algorithmes d'apprentissage et permettraient d'améliorer les performances du système.

6.2 Bases de données vidéos

Les bases de données vidéos sont utilisées d'une part pour évaluer la pertinence de la détection des points d'intérêt spatio-temporels décrits dans le chapitre 3, et d'autre part pour évaluer le système de définition de concepts, et donc les logiciels FX et BRIK. Dans ce paragraphe, nous présentons les différentes bases qui ont été utilisées dans cette thèse.

6.2.1 Bases de données vidéo pour l'évaluation de STIP

Les points d'intérêt spatio-temporels sont des points d'intérêt dans l'espace image qui présentent une évolution temporelle irrégulière. Pour étudier ces points, nous avons utilisé trois bases de vidéos différentes :

- Base de vidéo de synthèse ;
- Base mixte MARAT ;
- Base de films d'animation.

Ces deux dernières bases ont été choisies parce qu'elles ont donné lieu à des travaux de recherche par d'autres doctorants du LISTIC et de Gipsa-lab.

6.2.1.1 SYNTHÈSE - Séquences de synthèse

La base de séquences de synthèse (figure 6.9) est une base construite dans le but d'avoir des objets simples en mouvement contrôlé afin d'effectuer des mesures qualitatives sur les points d'intérêt. Chacune de ces vidéos présente le mouvement d'un seul objet simple (rectangle, cercle, triangle et polyligne) qui est en translation simple, avec de temps en temps des changements brutaux de trajectoire, ou a des trajectoires elliptiques. La couleur des ob-

jets est également contrôlée afin de faire des évaluations sur la couleur (blanc, couleurs, changement). Au total, 50 séquences (13 triangles, 15 cercles, 17 carrés, 5 polylignes). Elles durent 200 images à 25 images par seconde, soit environ 8 secondes, et ont une résolution de 288×288.

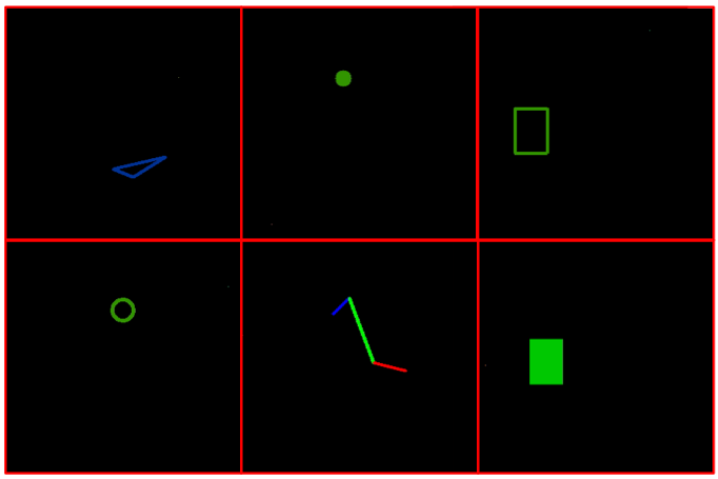


Figure 6.9 — Les différentes catégories de la base synthèse.

6.2.1.2 TELEVISION - Mixte - Base MARAT

Nous disposions d'une base de séquences issues de la télévision qui a été constituée par Sophie Marat lors de ces travaux de thèses [Marat *et al.*, 2009]. Cette base (figure 6.10) a été construite à parmi d'un lot de 53 vidéos qui ont été sélectionnées en provenance de diverses sources : films et séries, émissions de télévision, journaux télévisés, films d'animation, publicités, émissions sportives, clips musicaux et concerts. Ces vidéos ont été choisies pour représenter des scènes dynamiques aussi variées que possible. Elles rassemblent des scènes d'intérieur, d'extérieur, des scènes de jour et de nuit. Ces vidéos ont été découpées en extraits de 1 à 3 secondes ($1.86s \pm 0.61$) dans une résolution de 720×576 pixels par image et en 25 images par seconde.
Cette base a également servi dans l'étude comparative qui a été réalisée entre cartes de saillance et points d'intérêt, étude présentée dans le paragraphe 3.3.5.

6.2.1.3 ANIMATION - Base de films d'animation

Avec son festival [6] et son marché du film d'animation [7], Annecy est devenue une capitale mondiale de l'animation. Le LISTIC a développé depuis plusieurs années une colloration avec CITIA [8]

6. http ://www.annecy.org/edition-2011/festival/presentation-festival
7. http ://www.annecy.org/edition-2011/mifa/presentation-mifa
8. "Cité de l'image en mouvement". Etablissement à Caractère Culturel EPCC, gérant le festival et le marché d'animation - http ://www.annecy.org/home

Figure 6.10 — Quelques exemples de la base hétérogène MARAT.

Dans le cadre de cette collaboration, le LISTIC dispose de quelques films qui peuvent être utilisés à des fins de recherche[Ionescu, 2007; Païs, 2009]. Nous en avons sélectionné quelques uns : "Le moine et le poisson", "François le Vaillant", "Au bout du monde" (figure 6.11). A partir de ces 3 films d'animation, nous avons découpé et sélectionné 2000 plans. Ces plans sont en 25 images par seconde et respectivement dans une résolution de 720×560, de 352×264, et de 320×240.

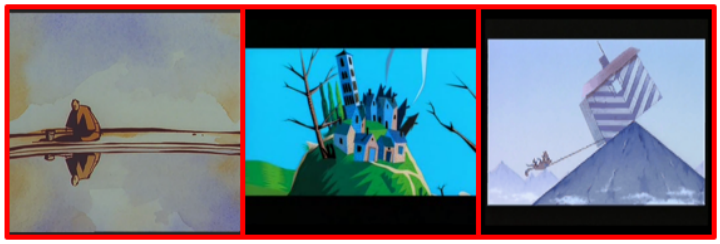


Figure 6.11 — Les trois films d'animation choisi : "Le moine", "François" et "Au bout du monde".

Ces séquences ont été utilisées dans le chapitre 3 pour l'étude sur les points d'intérêt ainsi que dans la définition de concepts notamment 'courir', 'marcher' et 'sauter'.

6.2.2 Vidéos pour la définition de concepts spatio-temporels

Dans le cadre de cette thèse, nous nous sommes intéressés à des concepts de haut niveau sémantique ayant trait aux mouvements de personnes. Les bases de données utilisées sont constituées uniquement de plans c'est à dire de séquences d'images dont l'acquisition a

été continue et ne présentant pas de coupures, pas de transitions et pas de changement de caméra. Nous avons choisi des vidéos concernant le mouvement sportif d'une part, puis le mouvement de personnes d'autre part.

6.2.2.1 SPORT - Bases UCF Sports Dataset et UCF-50

La base 'UCF Sports Action' (UCF-SA) est une base constituée par l'Université de Floride (University of Central Florida) [9]. Elle est composée de 200 vidéos issues de la BBC et d'ESPN (télévision) réparties sur 9 sports comme suit (tableau 6.1) :

Activité	Nombre de plans
plongeon	16 vidéos
golf (swing)	25 vidéos
course à pied	15 vidéos
marche à pied	22 vidéos
course à cheval	24 vidéos
patinage artistique	23 vidéos
boxe	25 vidéos
haltérophilie	15 vidéos
baseball	35 vidéos

Tableau 6.1 — Liste des activités de la base UCF-SA et leur quantité.

Ces séquences durent environ 3/4 secondes soit de 40 à 90 images et ont une résolution de 720×480.

La base 'UCF-50' (figure 6.12) est composée de séquences collectées sur YouTube et représentant 50 catégories différentes [10]. Au total, 5000 séquences (100 séquences par catégorie) d'environ 100 images et 4 secondes, en 25 images par secondes, dans une résolution de 320×240.

6.2.2.2 SPORT - Séquences d'athlétisme - Base RAMASSO

Cette base est composée de 60 plans de saut d'athlétisme. Elle a été constituée au GIPSA-Lab par Emmanuel Ramasso dans le cadre de sa thèse [Ramasso, 2007] (figure 6.13). On dispose de 4 sauts différents : le saut en longueur, en hauteur, le triple saut et le saut à la perche. Pour chacun des sauts, on a 15 plans mais dont les orientations des prises de vue peuvent

9. http ://www.cs.ucf.edu/vision/

10. Baseball Pitch, Basketball Shooting, Bench Press, Biking, Biking, Billiards Shot,Breaststroke, Clean and Jerk, Diving, Drumming, Fencing, Golf Swing, Playing Guitar, High Jump, Horse Race, Horse Riding, Hula Hoop, Javelin Throw, Juggling Balls, Jump Rope, Jumping Jack, Kayaking, Lunges, Military Parade, Mixing Batter, Nun chucks, Playing Piano, Pizza Tossing, Pole Vault, Pommel Horse, Pull Ups, Punch, Push Ups, Rock Climbing Indoor, Rope Climbing, Rowing, Salsa Spins, Skate Boarding, Skiing, Skijet, Soccer Juggling, Swing, Playing Tabla, TaiChi, Tennis Swing, Trampoline Jumping, Playing Violin, Volleyball Spiking, Walking with a dog, and Yo Yo

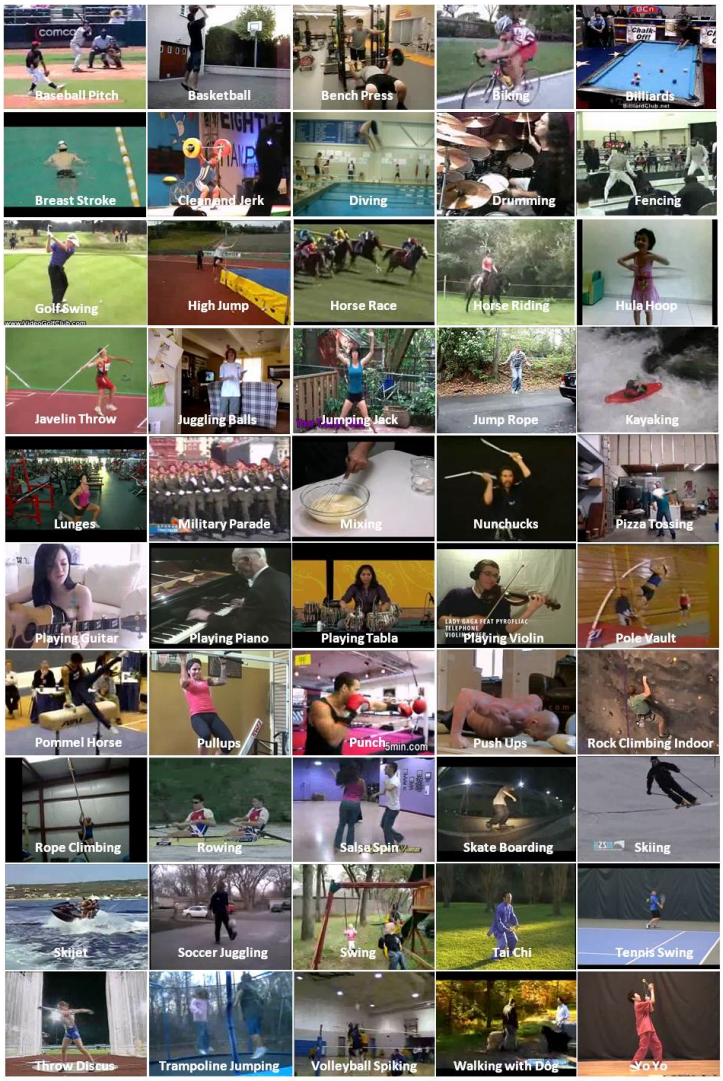


Figure 6.12 — Les différentes catégories de la base UCF-50.

varier. Ils durent de 100 à 160 images à 25 images par seconde, soit environ 5 secondes, et ont une résolution de 300×300. Il est important de noter que les plans de saut en hauteur sont, en fait, des séquences car on a un généralement un changement de caméra lors de la transition course/saut. Ainsi, cette transition sera confondue avec un changement dans le mouvement (chapitre 3). Malgré tout, la transition étant généralement située à un changement dans le mouvement, nous avons gardé ces séquences pour les différentes évaluations.

6.2.2.3 MOUVEMENT - Base KTH

La base "KTH" [11] a été construite par Ivan Laptev et Barbara Caputo (figure 6.14). Elle est composée de plus de 2000 plans qui constitue 600 séquences par combinaison de 25 sujets, 6

11. http ://www.nada.kth.se/cvap/actions/

Figure 6.13 — Les 4 types de saut de la base Ramasso : longueur/triple/hauteur/perche.

actions et 4 scénarios. Nous avons gardé les 2000 plans répartis sur 6 actions. La résolution est de 160 $\times$ 120 pixels par image et ces plans durent environ 4 secondes soit 100 images en 25 images/seconde. Cette base est particulièrement intéressante car elle a été utilisée dans plusieurs publications d'évaluation des méthodes de classification et d'indexation comme dans [Laptev *et al.*, 2007b].

Figure 6.14 — Les 6 différentes actions de la base KTH (par colonne : marcher, trottiner, courir, boxer, faire l'oiseau, applaudir).

6.2.3 MOUVEMENT - Base personnelle

Afin de disposer de plans en haute résolution, nous avons nous-même constitué une base de plans (figure 6.15). Ceux-ci ont été construits pour pouvoir disposer de plans à caméra fixe et de plans à caméra mobile. L'objectif de cette base est de pouvoir tester notre système sur des données plus volumineuses.

Les plans sont réalisés à l'aide de deux dispositifs différents : une caméra numérique JVC Evasio filmant en haute-résolution 1920 $\times$ 1080 en 25 images par seconde nous permettant de filmer en extérieur et une caméra numérique Panasonic (filmant en YUV422 format raw) dont les caractéristiques sont les suivantes : résolution de 640 $\times$ 480, 25 images/sec.

Nous avons filmé des plans en caméra fixe à l'aide des deux caméras et des plans en caméra mobile à l'aide de la caméra numérique haute-résolution. Les activités effectuées sont similaires à celles de la base KTH. Au final, nous disposons pour la caméra fixe de 20 plans pour chacune des actions suivantes : 'boxer', 'oiseau', 'applaudir', 'courir' et 'marcher

parallèle à la caméra', ainsi que 5 plans pour 'marcher vers la caméra' et 'marcher depuis la caméra vers l'extérieur', soit au total 110 plans. Pour la caméra mobile, on dispose de 10 plans pour chacune des actions suivantes : 'marcher', 'courir', 'marcher' et 'courir' en faisant des sauts.

Figure 6.15 — Exemple de plan filmé avec nos caméras.

6.2.4 Récapitulatif des différentes bases

L'ensemble des caractéristiques des différentes bases est donné dans 6.2. Au total, la base de plans est constituée de 10000 plans pour une taille totale de 40 Go (soit environ 45000 secondes - 12 heures 30).

Base	Résolution	Taille	Durée	Nombre
SYNTHESE	288x288	200 images	8 sec.	50
RAMASSO	300x300	100-160 images	5 sec.	60
MARAT	300x300	45 images	2 sec.	305
UCF Sports	720x480	40-90 images	3-4 sec.	200
UCF-50	320x240	100 images	4 sec.	5000
KTH	160x120	100 images	4 sec.	2000
Animation	variable	60-200 images	4-8 sec.	2000
Laboratoire	640x480	250 images	10 sec.	110
Laboratoire HD	1920x1080	250 images	10 sec.	30

Tableau 6.2 — Récapitulatif des différentes bases.

6.3 Évaluation

Dans cette section, nous proposons un certain nombre de tests permettant d'évaluer les capacités des systèmes que nous avons développés. Le premier test nous permet d'évaluer les temps de calcul nécessaires aux différents traitements. L'objectif est d'estimer le temps nécessaire à l'annotation assistée par rapport à une annotation manuelle complète de la base d'apprentissage. Le deuxième test permet d'évaluer les capacités du système à extraire des prototypes pertinents de la base de vidéos. Ces performances seront aussi comparées à celles obtenues de façon automatique sur la base KTH. Enfin, le dernier test propose d'évaluer la

pertinence du système de Questions/Réponses quant à sa capacité à définir des concepts. Il est important de noter que pour le second test (pertinence des prototypes) et pour le test comparatif sur KTH, les questions ont une grande importance puisqu'elles dépendent en partie de l'application. En l'occurrence, les questions actuelles ont été créées pour être spécifiquement utilisées sur ces bases.

6.3.1 Évaluation du temps de calcul et impact de la résolution des images

L'objectif de cette évaluation est l'étude des performances en temps de calcul des différentes étapes. Dans un premier temps, la base de vidéos est traitée automatiquement par le logiciel FX. Cette opération se fait indépendamment de l'Expert Application. Elle correspond au groupe d'*Extraction* offline. La seconde phase correspond à la *Définition* du concept par le système de questions/réponses puis à l'extraction de prototypes. Cette opération correspondant au groupe *définition* qui s'appuie sur les compétences de l'Expert Application et se fait online.

1. Groupe *Extraction* (offline) :
   - Tâche A : extraction des attributs images
   - Tâche B : extraction des plages de valeurs correspondantes dans les séquences
   - Tâche C : filtrage des données pour création des briques
2. Groupe *Définition* (online) :
   - Tâche D : définition d'un concept (20 questions en moyenne)
   - Tâche E : recherche des prototypes
   - Tâche F : annotation des prototypes (visualisation et OK/NON)
   - Tâche G : définition finale du concept

Dans le cadre de ce test, on évalue aussi l'impact de la résolution des images constituant les plans dans les temps des étapes A,B et C.

Pour ces tests de performances, deux machines ont été utilisées :
- un ordinateur portable (intitulé X) équipé d'un processeur Intel Core 2 Duo 1,86 Ghz, 2 Go Ram, disque dur en SSD pour "Solid State Drive" - Disque à mémoire Flash (100Mo/sec.)
- une machine de calcul (intitulé Y) équipée d'un Intel Quad Core à 3,2 Ghz, 4 Go de Ram, disque dur mécanique (10Mo/sec). Afin d'obtenir une base de comparaison, un seul des 2 'core' du Core 2 Duo et un seul du Quad Core ont été utilisés.

6.3.1.1 Évaluation des temps de traitement

Les données utilisées

Pour réaliser cette étude, nous avons défini deux bases de données vidéos différentes. La première (base 1) correspond à la totalité des vidéos décrites au paragraphe 6.2 soit 40 Go, et la seconde base (base 2) correspond uniquement à la base UCF-50 constituée des 1000 plans soit 20 par activité pour un total d'environ 3,5 Go. L'objectif est d'étudier les temps

de traitement dans un contexte de base variée et volumineuse (base 1) ainsi que dans un contexte de petite base spécifique (base 2).

Objectif du test

Le test consiste à mesurer le temps nécessaire pour chacune des sept tâches identifiées sur les deux machines de test et sur les deux bases présentées. Le tableau 6.3 donne, pour chacune des tâches, le temps total d'exécution. De plus, pour les tâches A et C, cette mesure temporelle est complétée par une mesure de vitesse de calcul, le nombre d'images traitées par seconde pour la tâche A et le nombre d'opérations effectuées par seconde pour la tâche C.

Résultats

	X		Y	
Tâche	Base 1	Base 2	Base 1	Base 2
extraction des attributs images (A)	28 heures	3 heures 6	9 heures 17	55 min.
	11,1 images/sec		30,4 images/sec	
extraction des plages de valeurs (B)	7 min 12 sec.	52 sec.	9 min 43 sec.	59 sec.
filtrage des données (C)	2h	14 min.	3h20min	19 min.
	33300 opérations/sec		13890 opérations/sec	
définition d'un concept (D)	estimé à environ 2 minutes			
recherche des prototypes (E)	5,1 sec.	476 ms	7,6 sec.	690 ms
annotation d'un lot de plans (F)	estimé à environ 5-7 minutes			
définition finale du concept (G)	<100ms			

Tableau 6.3 **—** Les différentes résultats obtenus sur chacune des tâches pour le test global.

Pour les tâches A, B et C, il convient de préciser que les paramètres de traitement sont identiques tout comme les attributs extraits et les définitions de briques élémentaires effectuées. Les étapes D, E, F et G ont été réalisées pour la définition des mêmes concepts. La différence au niveau de l'étape F est due uniquement à la quantité de prototypes retournés, un peu plus importante pour la base 1 que pour la base 2.

Analyses des résultats

On considère deux phases : la phase d'extraction (tâches A, B et C) qui est une phase 'off-line' et la phase de définition (D, E, F et G) qui est effectuée par un utilisateur donc 'on-line'.

Phase d'extraction

La phase d'extraction des attributs est logiquement la phase la plus longue : de 55 minutes à 3 heures pour la base 2, et de 9 à 28 heures pour la base 1. La puissance du processeur permet la diminution du temps de calcul.
L'opération d'extraction des valeurs et de filtrage (étapes B et C) ne fait intervenir que le disque de stockage par le biais de la base de données avec de nombreuses E/S. Le disque SSD (de la machine X) permet d'obtenir les opérations les plus rapides. Il n'est pas possible de paralléliser le traitement car il ne s'agit que d'accès à la base de données.
Ces trois tâches sont effectuées offline c'est à dire avant que l'Expert Application n'utilise le système.

Il est intéressant de noter que le processus étant parallélisable, la réactivation des 4 cores de la machine Y permet de passer de 9 heures à un peu plus de 2 heures.

Phase de définition

Cette seconde phase est difficile à évaluer. Le temps nécessaire à sa réalisation varie en fonction du temps passé sur chacune des questions, du nombre de prototypes disponibles, du nombre de prototypes visionnés etc. L'évaluation effectuée est donc très subjective et discutable. Cependant, il s'agit de déterminer un coût général en temps. Et comme beaucoup de processus, plus il est maîtrisé et moins le coût est important. Dans cette proposition d'évaluation, il faut environ *550 à 600 secondes* (9/10 minutes environ) pour effectuer cette deuxième phase consistant *à définir un concept* (questions/réponses puis prototypes).

Dans ce groupe, la phase de définition d'un concept, c'est à dire l'utilisation du système de questions/réponses, est effectuée indépendamment de la base de vidéos et de la machine pour 20 questions en moyenne. Il est communément admis qu'il faut environ 20 questions pour définir un concept en environ 2 minutes en se basant sur une estimation de 5 secondes pour répondre à chaque question et en arrondissant le résultat obtenu. Pour estimé ce temps, un expert a réalisé 20 définitions de concepts (base UCF). Cette phase est dépendante de l'Expert Applicatif. Il est important de noter ici, que l'Expert avait déjà utilisé le système et connaissait les questions. La réflexion était ainsi plus rapide que pour un Expert découvrant le système.
La tâche de recherche de prototypes se résume à des E/S sur la base de données et dure de 5,1 à 7,6 secondes pour la première base, et de 476 à 690 millisecondes pour la seconde base.
L'annotation d'un lot de plans s'effectue indépendamment de la machine mais dépend étroitement du nombre de prototypes trouvés et proposés à l'utilisateur ainsi que des visionnages effectivement réalisés par l'utilisateur (s'il décide de tous les regardent ou de n'en regarder aucun). La sélection de prototypes comporte plus de séquences sélectionnées lorsqu'on utilise la première base. Afin d'avoir une base de comparaison intéressante, nous avons choisi d'établir que l'utilisateur regardait tous les prototypes sélectionnés dans leur intégralité (le nombre de vidéos retournées restant limité). Ainsi, cette tâche dure

approximativement 7 minutes pour la première base et 5 minutes pour la seconde base. A l'instar de la mesure du temps nécessaire pour répondre aux questions, le temps mesuré ici est le temps moyen passé à regarder la liste des prototypes proposés ainsi que le visionnage de certains prototypes. Ce temps est difficile à estimer puisqu'il est dépendant du nombre de vidéos retournées.

Comparaison avec l'annotation manuelle

Les résultats temporels obtenus par l'approche proposée doivent être comparés au temps nécessaire à l'Expert Application pour extraire des prototypes des bases 1 et 2 correspondant au concept qu'il souhaite définir et réaliser l'annotation complètement manuelle des bases.
L'annotation manuelle consiste à regarder l'intégralité des vidéos et à attribuer l'annotation correspondante. Il est communément admis qu'une personne annotant manuellement ne peut traiter plus de 5 concepts en parallèle. Si elle doit annoter une base sur un index de 10 concepts, elle devra donc faire deux fois le visionnage. Le visionnage se fait néanmoins en accélérant le flux. Généralement, l'accélération est d'un facteur 4.

Cas d'une définition unique

Dans le cas d'une définition unique, en visionnant l'intégralité de la base (12 heures), à vitesse accélérée (X4), l'annotation manuelle de la base consistant juste à déterminer si le concept est présent ou non pour chaque séquence, dure environ 3 heures.
Dans la méthode proposée, la phase 'offline' dure 9 heures (machine la plus performante) et la phase 'online' consistant à effectuer la définition du concept dure 10 minutes.

Cas de définitions multiples

On réalise l'annotation de la totalité des vidéos (toutes les bases) avec tous les concepts à notre disposition (60 à 70). On considère que l'utilisateur peut gérer 5 concepts simultanément ce qui demande 14 visionnages complets. L'annotation manuelle peut donc être estimée à 42 heures (en vitesse accélérée - X4 et pour un utilisateur).
Dans la méthode proposée, la phase 'offline' dure toujours 9 heures et la phase 'online' peut être réalisée en environ 10 heures (70 concepts, 10 minutes par concept).

Bilan

L'annotation manuelle de la base est de l'ordre de 3 heures (en visionnant à une vitesse accélérée pour un seul concept) alors que l'annotation assistée est effectuée en 9 heures (incluant la phase offline) et 10 minutes (sans la phase offline) soit 300% de plus en comptabilisant la phase offline et 95% sans la phase offline.
Si on considère cette fois la base complète de 60/70 concepts, il faut environ 42 heures pour effectuer l'annotation manuelle par un seul expert et en vitesse accélérée. L'annotation par la

	Méthode	Durée
1 concept	Annotation manuelle (estimation)	3 heures
1 concept	Méthode assistée (MA)	**10 minutes**
1 concept	*MA + offline*	*9 heures*
70 concepts	Annotation manuelle (estimation)	42 heures
70 concepts	Méthode assistée (MA)	**10 heures**
70 concepts	*MA + offline*	*19 heures*

Tableau 6.4 — Temps d'annotation avec chacune des méthodes (manuelle et assistée) pour 1 concept et pour 70 concepts en utilisant la base complète de prototypes.

méthode assistée est effectuée en 19h (incluant la phase offline) ou 10 heures (sans la phase offline) soit une amélioration de l'ordre de 55% en comptabilisant la phase offline et 75% sans la phase offline.
En conclusion, la méthode proposée permet de gagner un temps très important sur l'annotation d'une base de vidéos en terme de coût humain (temps). Cependant, l'avantage de l'annotation manuelle est d'assurer une qualité d'annotation supérieure à celle qu'on peut proposer avec une méthode assistée.
Le tableau 6.4 récapitule les différentes durées sur un concept et 70 concepts sur les deux méthodes (manuelle et assistée).

6.3.1.2 Influence de la résolution

Ce test consiste à évaluer le temps nécessaire pour chacune des 3 tâches du groupe "extraction" en augmentant la résolution des images (figure 6.16).

Les tests ont été réalisés sur les bases suivantes :
- 25 plans de la base KTH correspondant à 6 actions avec une résolution de 160×120
- 25 plans de la base MARAT de résolution 300×300
- 25 plans d'une première base personnelle avec des actions similaires à la base KTH et une résolution 720×560
- 25 plans de la seconde base personnelle (résolution full-HD 1080p : 1920×1080)

Ainsi, nous disposons de 100 plans répartis équitablement sur les 4 résolutions. Ce test a été réalisé avec une seule machine (Y).

Nous obtenons les résultats suivants :

Globalement la taille des images change d'un facteur voisin de 4 entre chaque base. Comme nous le montre le tableau 6.5, l'évolution du temps de calcul pour la tâche A n'est pas linéaire par rapport au nombre de pixels à l'image. Dans un premier temps, l'augmentation n'est que de 33% (passage de 54 à 81 secondes) puis de 65% (passage de 81 à 229 secondes) puis enfin de 63% (passage de 229 à 605 secondes). Le passage de la plus faible résolution à la plus forte résolution multiplie le nombre de pixels par 64 alors que le temps de calcul a été multiplié par 10. Le temps de calcul n'est pas directement lié à la résolution de manière linéaire car, si certains extracteurs balayent l'ensemble de l'image ainsi leur temps de traitement est proportionnel à la résolution, d'autres ne sont pas

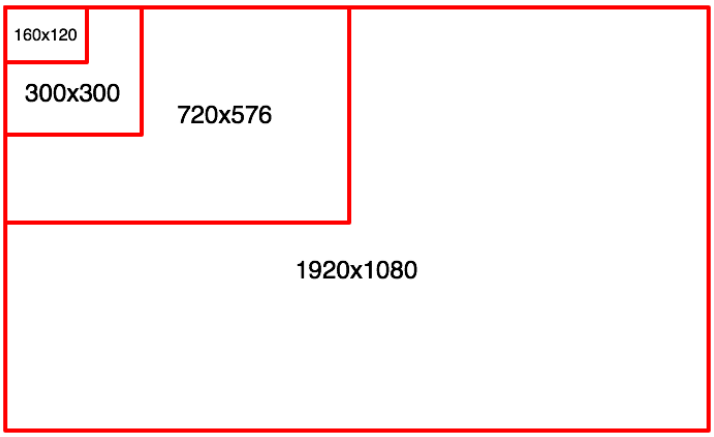


Figure 6.16 — Les différentes résolutions utilisées dans le test résolution.

Tâche	160 × 120	300 × 300	720 × 560	1920 × 1080
	19200 pixels	90000 pixels	403200 pixels	2073600
A	54 secondes	81 secondes	3 minutes et 49 sec.	10 minutes 05 sec.
	46 img/sec	30,8 img/sec	10,9 img/sec	4,16 img/sec
B	1,4 sec.	1,5 sec.	1,5 sec.	1,8 sec.
C	1 minute 55 sec.	1 minutes 54 sec.	1 minute 58 sec.	2 minutes 19 sec.

Tableau 6.5 — Les différents résultats obtenus sur chacune des tâches pour le test résolution pour 25 plans.

directement liés à la résolution de l'image (détecteur de Hough par exemple).

En ce qui concerne les tâches B et C, la différence entre les différents tests est logiquement à peine perceptible puisque ces tâches ne sont liées qu'aux données extraites à traiter dont la quantité reste sensiblement la même pour les différentes résolutions.

6.3.1.3 Conclusions

La méthode d'annotation assistée présente un réel avantage par rapport à l'annotation manuelle. Elle permet la diminution du temps de travail de l'Expert Applicatif et améliore la qualité de ce travail. En effet, il est plus motivant de répondre à un certain nombre de questions que de visionner un nombre important de vidéos à la recherche d'un concept. La durée de la tâche n'est plus proportionnelle à la quantité de vidéo comme c'est le cas avec de l'annotation manuelle mais est proportionnelle au nombre de concepts à définir.

La résolution des images est un point non négligeable sur la durée des traitements. La plupart des bases utilisées actuellement dans les évaluations de processus d'annotation ou d'indexation dispose de vidéos ayant des résolutions autour de 160 × 120 alors que sur

internet notamment, ce sont des formats haute-résolution qui sont massivement partagés (720p voir 1080p depuis quelques mois). Dans ces conditions, les performances s'écroulent rapidement devant la quantité d'informations à traiter comme on l'a vu avec le test sur la résolution. On comprend alors pourquoi la plupart des travaux d'indexation de séquences d'images ne travaillent pas au-delà de résolution de 520×400.

6.3.2 Évaluation de la qualité d'une définition

Ce test a pour objectif d'évaluer la qualité de la définition d'un concept c'est à dire d'étudier si cette définition permet de retrouver des prototypes puis des séquences qui correspondent effectivement à ce concept.

Le système de questions/réponses permet d'établir la définition d'un concept. Puis on recherche les prototypes correspondant à ce concept dans la première base (base dénotée AVANT dans la suite) de documents. A l'aide des prototypes validés, la définition du concept est corrigée ce qui permet de rechercher de nouveaux documents sur la seconde base (base dénotée APRÈS dans la suite). Les deux bases utilisées sont constituées chacune par la moitié de la base UCF-50 (1000 plans / 20 activités soit 50 plans par activité). Nous disposons donc de deux bases de 500 plans avec 25 plans par activité. On définit 5 concepts : 'marcher', 'courir', 'plonger', 'sauter', 'skier'.

La validation consiste à estimer la qualité des prototypes choisis (en précision/rappel) extraits de chacune des bases, avant et après correction du modèle de concept.

Il est important de noter que les performances dépendent étroitement des paramètres extraits, du jeu de questions/réponses et des réponses exprimées. Toutefois, ce test permet d'avoir une évaluation de la qualité d'une définition sur un jeu de questions/réponses donné, et de paramètres donnés.

Les questions et les briques utilisées

Pour la définition des 5 concepts, l'ensemble des questions a été utilisé sauf les questions du groupe "communication" comme "Est-ce un dialogue, une manifestation, une réunion ?" qui comporte 8 questions. Ceci représente donc 24 questions. Concernant les briques, l'ensemble des briques était accessible, soit 75 briques, mais seules 14 d'entre elles ont réellement servi dans les définitions ainsi que 6 opérateurs sur 18.

Les définitions de concepts obtenues

On dispose de deux définitions : celles obtenues avec le système de questions/réponses et celles corrigées en utilisant les informations issues de la validation des prototypes. Voici les définitions obtenues avec le système de questions/réponses :

– **marcher** :

$$\{c-f \oplus c-m \oplus c-F\,,\mathbf{m}\}\ \mathbf{d}\ no-1\ \mathbf{d}\ fo-io-lent\ \mathbf{d}\ \{fo-*\}^1 \tag{6.1}$$

Ce qui peut être traduit par :

> présence des briques de compacité faible, moyenne ou forte (c-f/c-m/c-F) de manière séquentielle (opérateur '**m**')
>
> ET un seul objet en mouvement (no-1)
>
> ET une intensité du flot optique plutôt faible (fo-io-lent)
>
> ET une orientation du flot optique quelconque (fo-$*$) mais plutôt homogène (translation - opérateur ^{1}).

– **courir** :

$$\{c-f \oplus c-m \oplus c-F\,,\mathbf{m}\}\ \mathbf{d}\ no-1\ \mathbf{d}\ fo-io-moyen\ \mathbf{d}\ \{fo-*\}^1 \tag{6.2}$$

Définition très similaire à celle de marcher. La seule brique différente est celle concernant l'intensité du flot optique (fo-io-moyen). On note ici l'intérêt d'avoir la possibilité de pouvoir créer un lien entre des concepts proches.

– **plonger** :

$$(fo-oo-bas\ \mathbf{d}\ env-eau)\mathbf{d}\ no-1\mathbf{d}\ fo-oi-fort \tag{6.3}$$

Ce qui correspond à la recherche de prototypes se déroulant dans l'environnement aquatique et concernant un seul objet en mouvement, mouvement plutôt orienté vers le bas et plutôt rapide.

– **sauter** :

$$\begin{aligned}((fo-oo-haut\ \mathbf{m}\ fo-oo-bas)\oplus(fo-oo-bas\ \mathbf{m}\ fo-oo-haut))\\ \mathbf{d}\ no-1\ \mathbf{d}\ stip-qi-moyen\end{aligned} \tag{6.4}$$

Cette définition correspond à un objet en mouvement générant des points d'intérêt spatio-temporels de manière modérée et s'effectuant en 2 phases (soit mouvement haut puis mouvement bas soit mouvement bas puis mouvement haut). En fait, on sait bien que sauter est d'abord un mouvement vers le haut mais le système ne permet via les questions/réponses de définir cette 'subtilité. L'intérêt des prototypes est également d'ajouter ce genre d'informations.

– **skier** :

$$((fo-*\mathbf{m}\ fo-oo-bas\ )\mathbf{d}\ env-neige\ \mathbf{m}\ no-1) \tag{6.5}$$

Dans un environnement de neige, cela correspond à un objet en mouvement dont le mouvement est plutôt orienté vers le bas à des vitesses variables. Pour cette définition, plutôt imprécise, c'est l'environnement qui nous aide beaucoup à retrouver des prototypes valides.

Voici les définitions finales obtenues avec le système de questions/réponses et après validation des prototypes. Il reste la définition commune à l'ensemble des prototypes validés :

– **marcher** :

$$\{c-f\ \mathbf{m}\ c-m\ \mathbf{m}\ c-F\ \mathbf{m}\ c-m\ \}^{*}\ \mathbf{d}\ no-1\ \mathbf{d}\ fo-io-lent\mathbf{d}\ \{fo-*\}^{1} \tag{6.6}$$

avec $c-f > c-m$ et $c-F > c-m$
On remarque que dans le premier terme, la première définition (équation 6.1) n'indiquait pas d'ordre entre $c-f$, $c-m$ et $c-F$. En étudiant les prototypes validés, un ordre d'alternance apparaît explicitement. De plus, on dispose également d'une information de durée concernant les briques de compacité : les briques 'compacité faible' et 'compacité forte' sont plus longues que les briques 'compacité moyenne'. Enfin, on remarque que certaines briques ont été ajoutées.
De la même manière, on peut observer des différences pour les autres concepts.

– **courir** :

$$\begin{aligned}(c-f\ \mathbf{m}\ c-m\ \mathbf{m}\ c-F\ \mathbf{m}\ c-m)^{*}\ \mathbf{d}\ \{no-1\ \mathbf{d}\\ fo-io-rapide\ \mathbf{d}\ stip-qi-fort\}\ \mathbf{d}\ \{fo-*\}^{1}\end{aligned} \tag{6.7}$$

avec $(c-f/c-F) > c-m$

– **plonger** : (fo-oo-bas **d** env-eau) **d** no-1
– **sauter** : (fo-oo-haut **m** fo-oo-bas) **d** no-1
– **skier** : ((fo-* **m** fo-oo-bas) **d** env-neige **m** no-1) **d** fo-ii-rapide

On remarque qu'entre les définitions initiales et les définitions finales, certaines informations ont été affinées notamment la position des briques dans la séquentialité des évènement (opérateur m) comme c'est le cas pour courir et marcher par exemple.

Les résultats

	marcher	courir	plonger	sauter	skier	MOYENNE
nombre prototypes détectés par le système	22	18	31	33	19	25
nombre prototypes validés	11	8	11	14	14	12
nombre prototypes présents dans la base	25	25	25	25	25	25
précision	0,50	0,44	0,35	0,42	0,74	0,49
rappel	0,44	0,32	0,44	0,56	0,56	0,46

Tableau 6.6 **—** Résultats sur la qualité des prototypes extraits de la base AVANT sur 5 activités.

Le premier test (tableau 6.6) permet la sélection de 18 à 33 plans (sur un total de 500) correspondant à la définition effectuée avec une précision et un rappel de l'ordre de 50%. On note le bon résultat de la précision du concept 'ski'.
Le second test (tableau 6.7) permet la sélection d'une vingtaine de plans (sur un total de 500) correspondant au concept défini avec une précision et un rappel de l'ordre de 86%. On note que les meilleures performances sont obtenues pour le concept 'ski'. Le concept

	marcher	courir	plonger	sauter	skier	MOYENNE
nombre prototypes détectés par le système	28	29	21	23	26	25
nombre prototypes validés	21	22	19	18	24	21
nombre prototypes présents dans la base	25	25	25	25	25	25
précision	0,88	0,86	0,90	0,78	0,92	0,87
rappel	0,84	0,88	0,76	0,72	0,96	0,83

Tableau 6.7 — Résultats de la qualité des plans de la base APRÈS retrouvés avec les définitions validées sur 5 activités.

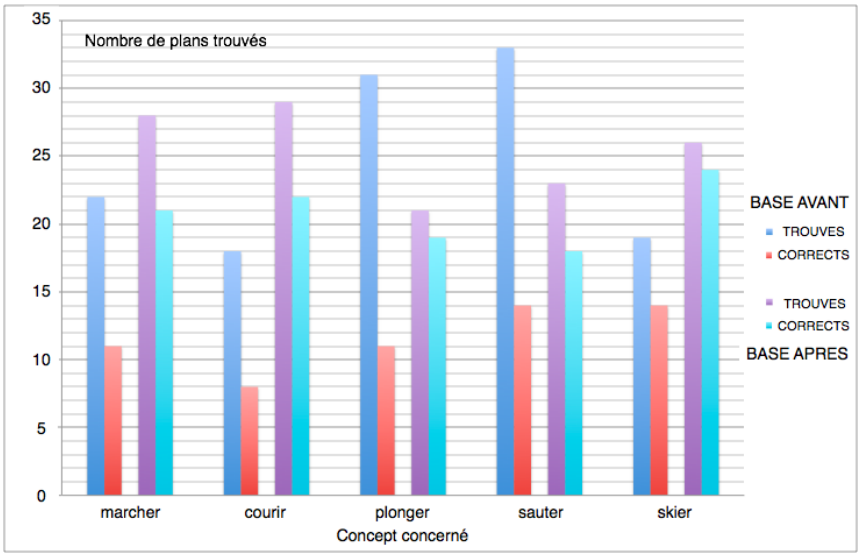


Figure 6.17 — Comparaison des différents résultats sur la base AVANT et la base APRÈS : proportions.

'sauter' montre des résultats plutôt faible par rapport aux autres et le concept 'plonger' présente une bonne précision.

Le cas du ski est particulier. Celui-ci bénéficie de l'information sur la couleur de l'environnement (blanc) qui ne se retrouve pas dans d'autres plans et permet une grande précision.
Dans les autres cas, la validation des prototypes permet l'amélioration de la précision et du rappel de l'ordre de 55% (figure 6.17). Le concept 'sauter' présente des résultats plus faibles car la définition effectuée engendre la sélection de mouvement présentant des similarités avec les gestes de saut. Par exemple, certains plans sélectionnés sont de l'haltérophilie. Pour le concept 'plonger', le problème est similaire au concept 'sauter' mais il est atténué par l'information sur la couleur de l'environnement.

En conclusion, on peut dire que le système de validation de prototypes permet l'amé-

lioration très nette de la précision et du rappel c'est à dire l'amélioration de la définition initialement effectuée à partir du système de questions/réponses. Bien que ce ne soit pas l'objectif, avec une précision et un rappel de l'ordre de 85%, le système établit une annotation qui n'est pas parfaite au contraire de celle effectuée manuellement qui même si elle est dépendante de l'expert, peut être considérée comme relativement bonne.

6.3.3 Évaluation comparative avec ceux de la base KTH

Ce test consiste à définir les 6 actions de la base KTH qui sont 'marcher', 'trottiner', 'courir', 'boxer', 'battre des bras', et 'applaudir' à l'aide du système de questions-réponses en utilisant une première partie de la base puis de rechercher ensuite ces actions en utilisant les définitions précédemment établies dans une seconde partie de la base. Ce test correspond au test de la qualité d'une définition mais sur une base plus restreinte. L'objectif est de pouvoir comparer notre système avec les autres méthodes qui ont été évaluées dans [Laptev *et al.*, 2007b], en utilisant la même base (la base KTH).

Dans [Laptev *et al.*, 2007b], quatre méthodes sont évaluées (**l'ensemble de ces méthodes utilisent un classifieur SVM**) : OF-PDHIST, Spatial-4Jets, Global-STG-HIST-MS et Global-STG-HIST-ZI. OF-PDHIST est la méthode proposée de détection d'événements spatio-temporels basée sur un lot de descripteurs. Spatial-4Jets utilise les points d'intérêt spatiaux ainsi qu'un descripteur spatial JETS de quatrième ordre. Global-STG-HIST-MS est l'histogramme global des gradients spatio-temporels normalisés calculés à plusieurs échelles spatiales et temporelles et Global-STG-HIST-ZI utilise l'histogramme global des gradients spatio-temporels normalisés comme présentés dans [Zelnik-Manor et Irani, 2001].

Pour réaliser ce test, nous avons utilisé 1000 plans de la base KTH pour définir les 6 actions. Puis nous avons cherché tous les plans correspondants à chacune des actions dans la seconde base de 1000 plans.

Les résultats obtenus sont les suivants :

	courir	trottiner	marcher	boxer	applaudir	battre
courir	0.95	0.04	0.01	0.00	0.00	0.00
trottiner	0.05	0.85	0.10	0.00	0.00	0.00
marcher	0.00	0.03	0.97	0.00	0.00	0.00
boxer	0.00	0.00	0.00	0.97	0.00	0.03
applaudir	0.00	0.00	0.00	0.00	1.00	0.00
battre	0.00	0.00	0.00	0.03	0.04	0.93

Tableau 6.8 — Matrice de confusion de la reconnaissance des 6 activités avec la méthode proposée.

Le tableau 6.8 présente la matrice de confusion entre le concept recherché et les concepts sortis. On note que 'applaudir' est très bien détecté et que globalement la reconnaissance est autour de 95% sauf sur le concept 'trottiner' qui est plus difficilement dissocié de 'marcher'

	courir	trottiner	marcher	boxer	applaudir	battre
OF-PDHIST	1.00	0.91	1.00	0.87	1.00	1.00
Spatial-4Jets	0.19	0.66	0.12	0.38	0.22	0.31
Global-STG-HIST-MS	0.97	0.75	0.97	0.78	0.91	1.00
Global-STG-HIST-ZI	1.00	0.75	1.00	0.69	0.72	0.94
BRIK-QR	0.95	0.85	0.97	0.97	1.00	0.93

Tableau 6.9 — Comparaison des reconnaissances des 6 activités pour les différentes méthodes.

et de 'courir'.

Le tableau 6.9 compare les reconnaissances effectives en terme de rappel des 6 différentes activités des différentes méthodes présentées dans [Laptev *et al.*, 2007b] et de notre méthode BRIK/QR. Globalement la méthode proposée indique des taux de reconnaissance variant entre 93% et 100% excepté pour le concept 'trottiner' à 85%. Elle n'obtient le meilleur résultat sur aucune des 6 activités mais obtient des résultats plus homogènes que Spatial-4Jets, Global-STG-HIST-MS et Global-STG-HIST-ZI. Elle reste néanmoins en deçà des performances de OF-PDHIST (sauf pour l'activité 'boxer').

Ce test met en avant les bonnes facultés à différencier les concepts définis. Le plus faible taux (85%) est obtenu par le concept 'trottiner' qui est plus difficilement reconnu du fait de sa ressemblance avec les concepts 'marcher' et 'courir'. De la même façon, le concept 'battre' est proche de 'boxer' ou d'applaudir' et amène quelques erreurs de reconnaissance.
Globalement, la méthode marche relativement bien. On remarque que les taux sont supérieurs à ceux du test précédent sur la qualité de la définition mais cette fois, chaque vidéo appartient obligatoirement à l'un des 6 concepts alors que dans le test précédent certains vidéos (une grande partie même) ne présentaient pas forcément le concept défini, la diversité de la base de données étant beaucoup plus importante.

En conclusion, la comparaison avec les autres méthodes montre que la méthode BRIK-QR donne des résultats satisfaisants qui ne sont pas les meilleurs mais qui sont homogènes. Il est important de noter que les méthodes auxquelles nous avons comparé BRIK-QR sont toutes des méthodes basées sur de l'apprentissage par utilisation de classifieur SVM. Enfin, les résultats semblent fortement liés à la définition issue des questions/réponses. Quelques tests ont été faits pour étudier l'influence des questions/réponses, mais le manque de temps n'a pas permis d'analyser plus précisément ce point important.

6.4 Conclusion

Les tests effectués ont permis de mettre en évidence plusieurs points :
– le temps du processus dépend quasiment uniquement de la tâche d'extraction de caractéristiques ;

– la définition du concept produite est utilisable et plutôt de bonne qualité (85% de taux de reconnaissance).

Au final, la méthode proposée semble être une alternative intéressante à l'annotation manuelle d'une base d'apprentissage. Plus la base d'apprentissage utilisée sera importante et plus l'intérêt de la méthode sera grand puisque c'est dans ces conditions qu'elle permet le gain de temps le plus appréciable.
Il est intéressant de noter que plus la base est spécialisée et plus les performances sont importantes. Ainsi, l'utilisation de la méthode proposée pour une base de séquences spécifique contenant quelques dizaines de concepts permet un gain de temps appréciable et des performances satisfaisantes.
Les améliorations proposées concernant l'apprentissage (perspectives des chapitres 4 et 5) pourraient encore améliorer ces résultats mais il sera nécessaire de vérifier que l'ajout de ces améliorations ne nuit pas aux performances temporelles actuelles pour un gain pas ou peu perceptible concernant la qualité des définitions.
Enfin, le premier test a mis en avant une faiblesse actuelle du système : la définition des questions-réponses et des liens entre réponses et briques. L'apprentissage de ces relations permettra sans doute d'améliorer le système alors que la collaboration avec des spécialistes de la théorie du langage pourrait ouvrir la voie à une nouvelle liste de questions-réponses.

Conclusion

« Les gens heureux ont une tendance à préférer les accomplissements de performance aux accomplissements de finalité. Le plaisir en est peut-être plus durable. »

Robert Blondin – Le bonheur possible

La définition de concepts signifiants, particulièrement temporels, pour l'annotation d'une base de séquences d'images passe généralement par une phase d'apprentissage à partir d'une base, dite d'apprentissage, déjà annotée. Le travail effectué lors de cette thèse concerne le développement d'un système d'aide à l'annotation de cette première base d'apprentissage. Il consiste à définir un modèle du concept spatio-temporel choisi par l'expert applicatif, à partir d'attributs extraits des séquences d'images et plus particulièrement d'attributs attachés au mouvement.

Contributions

Les principales contributions proposées dans ce travail de thèse (figure 6.18) portent sur :

- **L'extraction d'informations** : nous avons mis en place un certain nombre d'extracteurs et nous nous sommes particulièrement intéressés aux points d'intérêt spatio-temporels. L'étude des caractéristiques et du comportement de ces points nous a permis de mettre en lumière plusieurs utilisations intéressantes comme la détection de transitions, la détection d'objets en mouvement et surtout la détection de changements dans les mouvements qui a été intégrée comme une donnée principale dans le système proposé. L'étude de ces points particuliers a été complétée par une comparaison avec le système de vision humaine au travers les modèles de saillance et a permis de déterminer que ceux-ci étaient particulièrement pertinents en terme d'intérêt visuel pour un utilisateur.
- **La représentation des connaissances** : la définition d'un modèle de représentation (le modèle de briques) et de manipulation de données (combinaison par opérateurs) permettant la prise en compte de l'aspect spatio-temporel des informations est une des deux contributions principales de ces travaux. Ce modèle est particulièrement souple quant à l'utilisation des briques. On peut facilement ajouter ou retirer des modèles et

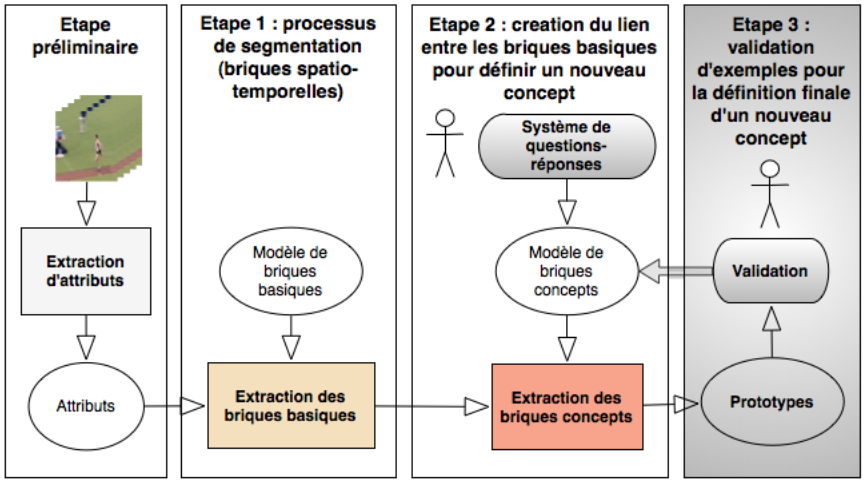


Figure 6.18 — Système global proposé : phase préliminaire d'extraction, représentation des connaissances sous forme de briques puis extraction des briques, et enfin système d'assistance.

les opérateurs sont assez génériques pour être directement utilisables.

– **Le système d'assistance** : à partir d'un système de questions/réponses, nous avons construit un système d'assistance à l'expert applicatif dans la définition de concept. Il facilite l'extraction des connaissances de l'utilisateur et la définition d'un concept de manière experte (règles) tout en permettant de relier directement ses connaissances avec les informations extraites et modélisées sous forme de briques. Ce modèle, bien qu'évalué sur des bases spécifiques et sur des questions un peu spécialisées, a montré des performances intéressantes.

Conclusions

Le modèle de représentation proposé, intitulé modèle de briques, permet de tenir compte de l'aspect spatio-temporel des données et de représenter des informations plus complexes en les combinant par utilisation d'opérateurs de combinaison et de liaison. Ainsi, à partir de données extraites, on construit des informations où la sémantique est de plus en plus importante dans le but de réduire le fossé sémantique.

Le système d'assistance présenté, construit sur un modèle de définition par questions/réponses, à l'instar de certains systèmes d'intelligence artificielle des années 80, permet d'apporter une réponse à la difficulté qu'il y a à réaliser la liaison entre les données modélisées sous forme de briques et les concepts à définir. Ce système permet l'extraction de la connaissance de l'expert applicatif et la création de la liaison entre cette connaissance et les modèles de briques.

L'ensemble proposé permet de disposer d'un premier système d'annotation assistée

d'une base de données vidéo. Il a montré certaines capacités quant aux performances puisqu'il permet de réduire de manière significative le temps nécessaire à l'annotation (de l'ordre de 75% sur nos évaluations) mais également sur la qualité des définitions produites puisqu'il permet la récupération de prototypes avec une précision de 75% environ et la reconnaissance des concepts à plus de 90%. Cependant, les performances du système sont liées à un certain nombre de d'éléments qui sont, pour l'instant, encore très primaires. En effet, la définition des modèles de concepts est étroitement liée au jeu de questions/réponses disponible qui a été défini par expertise, ainsi qu'à la qualité des réponses fournies par l'expert applicatif. Le passage du fossé sémantique est réalisé en traduisant les réponses en choix de briques ainsi qu'en relations entre briques. Nous avons développé des outils qui facilitent ce passage, mais ils sont certainement encore trop réduits pour couvrir un large spectre de définitions. Dans ces travaux de thèse, nous nous sommes concentrés sur la validation d'une démarche, qui n'est pas suffisamment riche actuellement pour répondre à des applications diversifiées, mais qui devrait être amenée à s'enrichir au fil des utilisations.

Au final, la conclusion la plus importante de ces travaux est qu'une intervention limitée de l'expert applicatif permet d'atteindre des performances d'indexation d'un niveau proche de celles obtenues avec une annotation manuelle classique.

Perspectives et améliorations futures

Au cours du développement des différentes étapes présentées dans ce manuscrit et de l'évaluation, un certain nombre de difficultés et de limitations ont été mises à jour. Elles permettent de envisager quelques pistes de réflexion pour les améliorations futures du processus ainsi que des perspectives intéressantes pour ces travaux de recherche.

A court terme

Les premières améliorations envisagées correspondent aux informations extraites. Le système présenté comporte une douzaine d'extracteurs d'informations qui ont été mis en place au début de ces travaux. Un certain nombre de caractéristiques seraient susceptibles d'apporter des informations intéressantes tels que HoG [12], HoF [13], SIFT [14], SURF [15], 'Bag of features'. De plus, certains extracteurs tels que détecteurs de visages, de silhouettes, de cercles pourraient également être utilisés dans la construction des briques. Enfin, à partir de ces différents extracteurs, le nombre d'attributs créés peut être considérablement augmenté. Dans un premier temps, à partir de notre douzaine d'extracteurs, nous avons construit par combinaison de données, une vingtaine d'attributs. L'augmentation du nombre d'attributs aura comme conséquence directe l'augmentation du nombre de briques et donc l'augmentation de la diversité du système. Mais cela accroîtrait également la difficulté à choisir un

12. HoG - Histogram of Gradient - [Dalal et Triggs, 2005]
13. HoF - Histogram of Flow - [Dalal *et al.*, 2006]
14. SIFT - Scale Invariant Feature Transform [Lowe, 2004]
15. SURF - Speeded Up Robust Features - [Bay *et al.*, 2006]

concept.
Cependant, l'augmentation du nombre d'attributs pose également le problème du risque d'explosion combinatoire. L'étape d'extraction 'offline', déjà relativement longue, sera considérablement allongée avec l'ajout de nouveaux extracteurs. Il conviendra alors, sans doute, d'effectuer des choix d'attributs en supprimant ceux inutilisés ou peu utilisés de manière rétroactive ou adaptative, afin de garder un nombre satisfaisant d'attributs.
Enfin, l'ajout ou la modification d'attributs passe également par la modification des liaisons entre questions/réponses et briques. La liaison peut être réalisée par l'expert en traitement d'images ou par apprentissage en associant les nouvelles données extraites à des réponses données en utilisant les prototypes.

A moyen terme

Les mécanismes proposés dans les perspectives des chapitres 4 et 5 constituent l'essentiel des perspectives à moyen terme : elles correspondent aux améliorations des modèles de briques et du système de questions/réponses.

Le modèle de briques peut être amélioré de nombreuses façons comme évoqué dans le chapitre 4. La plus simple est l'augmentation du nombre de modèles de briques par la création de nouvelles informations extraites mais également par des travaux sur la définition des différents intervalles de définition notamment par l'utilisation de techniques d'apprentissage permettant de se servir des définitions effectuées pour corriger les intervalles de définition existants, pour en ajouter ou en supprimer. Le processus de définition peut également être modifié notamment en utilisant la libération de contraintes sur la recherche, sur les briques ou sur les opérateurs, c'est à dire en sélectionnant en plus des briques "proches", des définitions présentant une correspondance approchée.

Les questions du système questions/réponses peuvent être retravaillées avec des linguistes afin de les rendre plus génériques. Deux possibilités sont à envisager pour retravailler les questions : soit les rendre génériques mais peut-être au détriment de la capacité d'adaptation du système à toutes les applications, soit les retravailler ponctuellement pour chaque application et les rendre les plus pertinentes possible pour une application donnée et donc moins génériques.
La liaison entre les différentes réponses et les briques est un point d'amélioration important. Ces liaisons sont actuellement créées par l'expert en traitement d'images et n'évoluent pas dans le temps. L'utilisation d'un processus d'adaptation consistant à créer, modifier et supprimer les liaisons en fonction des réponses données et des prototypes sélectionnés aura un impact intéressant sur la qualité des définitions produites.
L'ensemble des questions/réponses peut également être modifié en ajoutant, en modifiant ou en supprimant des questions. La création des liaisons entre ces nouvelles questions (ajoutées ou modifiées) et les données extraites pourra être réalisée par un expert en traitement d'images ou par apprentissage en insérant les nouvelles questions ponctuellement dans le processus et en utilisant les prototypes validés.

A long terme

A long terme, la perspective la plus intéressante est le passage à l'échelle et l'utilisation du système en mode collaboratif, objectif final qui, lui seul, garantira la pertinence de l'approche proposée. Chacune des utilisations du système de définition permet d'obtenir un ensemble de données (réponses, sélection de briques et d'opérateurs, prototypes validés et refusés, nom du concept). L'apprentissage utilisant ces données peut ainsi permettre l'amélioration du système.
De la même façon, l'utilisation de bases plus importantes et plus hétérogènes permettra de vérifier si le système est capable de devenir générique ou s'il restera un outil spécifique adaptable aux différents types de base de données et aux différentes applications.
Enfin, l'utilisation d'une ontologie pour modéliser l'ensemble des concepts entre eux permettrait également de compléter certains modèles de concepts et allégerait l'interaction nécessaire avec l'utilisateur.

Multi-modalité

Dans l'optique de fournir un système d'annotation complet et performant pour l'indexation, les récentes recherches sur l'analyse multimodale des vidéos montrent l'intérêt de la prise en compte des multimodalités d'un document. Pour les vidéos, la prise en compte du son est une perspective à court terme qui permettra d'apporter des informations très intéressantes tout comme la prise en compte du texte lorsque celui-ci est disponible (synopsis, sous-titre, résumé...).

Conclusion finale

Le système proposé est une première étape dans la création d'un système à grande échelle, collaboratif et adaptable à l'application. Les améliorations qui sont susceptibles d'être apportées sont nombreuses. Les recherches à effectuer sur les différents points clés du processus sont multiples mais semblent tout à fait intéressantes et prometteuses. Il sera toutefois nécessaire d'insérer également le son et le texte dans le traitement. Encore inexploités, ils sont le vecteur d'un nombre important d'informations. Le résultat qui pourrait en résulter à terme est un système d'annotation rapide et performant car chaque jour, le nombre de vidéos disponibles est en augmentation et où le besoin d'indexation est grandissant.

Quatrième partie

Annexes

A Liste des caractéristiques extraites

compacité	objet	rapport entre hauteur et largeur
type caméra	séquence	fixe ou mobile
orientation caméra	image	orientation cardinale
zoom	image	avant/arrière
nombre de lignes	image	lignes par hough
taille objet	objet	rapport l'objet par rapport à l'image
position verticale objet	objet	par rapport à l'image
position horizontale objet	objet	par rapport à l'image
nombre de SIP par objet	objet	nombre de points
nombre de SIP par image	image	nombre de points
nombre de STIP par objet	objet	nombre de points
nombre de STIP par image	image	nombre de points
intensité du flot optique par objet	objet	intensité du vecteur principal
intensité du flot optique par image	image	intensité du vecteur principal
orientation du flot optique	image	orientation cardinale
position verticale	objet	par rapport à l'image
position horizontale	objet	par rapport à l'image
nombre d'objets	objet	comptage automatique

Tableau A.1 — Liste des caractéristiques extraites des images.

B Liste des modèles de briques basiques définis

Nom du modèle	Caractéristiques	Modèle	Type	Définition
c-faible	compacité	faible	intervalle	[0 ; 0.4]
c-moyenne	compacité	moyen	intervalle	[0.4 ; 0.65]
c-fort	compacité	fort	intervalle	[0.65 ; 1.0]
cam-fixe	camera	fixe	valeur	0
cam-mobile	camera	mobile	valeur	1
camera-ohh	orientation camera	haut	intervalle	[0 ; 0.4]
camera-ohm	orientation camera	milieu	intervalle	[0.4 ; 0.6]
camera-ohb	orientation camera	bas	intervalle	[0.6 ; 1.0]
camera-ovg	orientation camera	haut	intervalle	[0 ; 0.4]
camera-ovm	orientation camera	milieu	intervalle	[0.4 ; 0.6]
camera-ovd	orientation camera	bas	intervalle	[0.6 ; 1.0]

Tableau B.1 — Liste des différents modèles de briques définis et leurs caractéristiques : n 1 à 11.

Dans le tableau B.2, l'intervalle sur la taille de l'objet est une proportion par rapport à la taille de l'image. Il est impossible d'évaluer la taille d'un objet de manière absolue, cette donnée est donc relative à la taille de l'image. Ainsi, 0,25 correspond à une taille d'un quart de l'image.

Il faut noter que certains ensembles de définition de caractéristiques n'ont pas été découpés en utilisant la méthode de répartition : l'orientation du flot optique (par 8 directions cardinales), les positions (par cadran de taille égale) et le sens du mouvement.

zoom-av	zoom	avant	valeur	0
zoom-ar	zoom	arrière	valeur	1
l-faible	nombre de lignes	faible	intervalle	[0 ; 10]
l-moyen	nombre de lignes	moyen	intervalle	[10 ; 100]
l-fort	nombre de lignes	fort	intervalle	[100 ;[
taille-petit	taille objet (/image)	petit	intervalle	[0 ; 0.25]
taille-moyen	taille objet (/image)	moyen	intervalle	[0.25 ; 0.4]
taille-grand	taille objet (/image)	grand	intervalle	[0.4 ; 1.0]
ov-haut	position verticale objet	haut	intervalle	[0 ; 40%]
ov-milieu	position verticale objet	milieu	intervalle	[40% ; 60%]
ov-bas	position verticale objet	bas	intervalle	[60% ;100%]
oh-gauche	position horizontale objet	gauche	intervalle	[0 ; 40%]
oh-milieu	position horizontale objet	milieu	intervalle	[40% ; 60%]
oh-droite	position horizontale objet	droite	intervalle	[60% ;100%]

Tableau B.2 **—** Liste des différents modèles de briques définis et leurs caractéristiques : n 12 à 25.

sip-qo-faible	quantité SIP (objet)	faible	intervalle	[0.0 ; 0.01]
sip-qo-moyen	quantité SIP (objet)	moyen	intervalle	[0.01 ; 0.05]
sip-qo-fort	quantité SIP (objet)	fort	intervalle	[0.05 ; 1.0]
sip-qi-faible	quantité SIP (image)	faible	intervalle	[0.0 ; 0.01]
sip-qi-moyen	quantité SIP (image)	moyen	intervalle	[0.01 ; 0.05]
sip-qi-fort	quantité SIP (image)	fort	intervalle	[0.05 ; 1.0]
stip-qo-faible	quantité STIP (objet)	faible	intervalle	[0.0 ; 0.01]
stip-qo-moyen	quantité STIP (objet)	moyen	intervalle	[0.01 ; 0.05]
stip-qo-fort	quantité STIP (objet)	fort	intervalle	[0.05 ; 1.0]
stip-qi-faible	quantité STIP (image)	faible	intervalle	[0.0 ; 0.01]
stip-qi-moyen	quantité STIP (image)	moyen	intervalle	[0.01 ; 0.05]
stip-qi-fort	quantité STIP (image)	fort	intervalle	[0.05 ; 1.0]

Tableau B.3 **—** Liste des différents modèles de briques définis et leurs caractéristiques : n 26 à 37.

fo-io-faible	intensité du FO (objet)	faible	intervalle	[0.0 ; 0.2]
fo-io-moyen	intensité du FO (objet)	moyen	intervalle	[0.2 ; 0.4]
fo-io-fort	intensité du FO (objet)	fort	intervalle	[0.4 ; 1.0]
fo-ii-faible	intensité du FO (image)	faible	intervalle	[0.0 ; 0.2]
fo-ii-moyen	intensité du FO (image)	moyen	intervalle	[0.2 ; 0.4]
fo-ii-fort	intensité du FO (image)	fort	intervalle	[0.4 ; 1.0]
fo-n	orientation du FO	nord	intervalle	$[\frac{3\pi}{8} ; \frac{5\pi}{8}]$
fo-no	orientation du FO	nord-ouest	intervalle	$[\frac{5\pi}{8} ; \frac{7\pi}{8}]$
fo-o	orientation du FO	ouest	intervalle	$[\frac{7\pi}{8} ; -\frac{7\pi}{8}]$
fo-so	orientation du FO	sud-ouest	intervalle	$[-\frac{7\pi}{8} ; -\frac{5\pi}{8}]$
fo-s	orientation du FO	sud	intervalle	$[-\frac{5\pi}{8} ; -\frac{3\pi}{8}]$
fo-se	orientation du FO	sud-est	intervalle	$[\frac{-3\pi}{8} ; -\frac{1\pi}{8}]$
fo-e	orientation du FO	est	intervalle	$[-\frac{1\pi}{8} ; \frac{1\pi}{8}]$
fo-ne	orientation du FO	nord-est	intervalle	$[\frac{1\pi}{8} ; \frac{3\pi}{8}]$

Tableau B.4 — Liste des différents modèles de briques définis et leurs caractéristiques : n 38 à 51.

smv-haut	position verticale objet	haut	intervalle	[0 ; 45%]
smv-nul	position verticale objet	milieu	intervalle	[45% ; 55%]
smv-bas	position verticale objet	bas	intervalle	[55% ; 100%]
smh-gauche	position horizontale objet	gauche	intervalle	[0 ; 45%]
smh-nul	position horizontale objet	milieu	intervalle	[45% ; 55%]
smh-droite	position horizontale objet	droite	intervalle	[55% ; 100%]
no-1	nombre d'objets	1	intervalle	[1 ;1]
no-2	nombre d'objets	2	intervalle	[2 ;2]
no-3+	nombre d'objets	3+	intervalle	[3 ;[

Tableau B.5 — Liste des différents modèles de briques définis et leurs caractéristiques : n 52 à 60.

col-blanc	couleur dominante	blanc	valeur	blanc
col-bleu	couleur dominante	bleu	valeur	bleu
col-vert	couleur dominante	vert	valeur	vert
col-gris	couleur dominante	gris	valeur	gris
col-marron	couleur dominante	marron	valeur	marron
lum-clair	luminance	clair	intervalle	[0 ;0,4]
lum-moyen	luminance	moyen	intervalle	[0,4 ;0,6]
lum-fonce	luminance	fonce	intervalle	[0,6 ;1]

Tableau B.6 — Liste des différents modèles de briques définis et leurs caractéristiques : n 61 à 68.

nbp-1	nombre de phases	1	valeur	1
nbp-2	nombre de phases	2	valeur	2
nbp-3	nombre de phases	3	valeur	3
nbp-n	nombre de phases	n	valeur	>3
dur-court	durée	court	intervalle	[0 ;75]
dur-moyen	durée	moyen	intervalle	[75 ;300]
dur-long	durée	long	intervalle	[300 ;[

Tableau B.7 — Liste des différents modèles de briques définis et leurs caractéristiques : n 69 à 75.

C Liste des liens entre réponses et briques/opérateurs

Numéro	Questions
1	Quel est le type d'activité
2	Combien de personnages sont concernés ?
3	Y a-t-il des spectateurs ? des observateurs ?
4	Y a-t-il un environnement particulier ?
5	Est-ce en intérieur ou en extérieur ?
6	Est-ce plutôt le jour ou la nuit ?
7	Est-ce une activité sportive ?
8	Est-ce plutôt une activité nautique ? aérienne ?
9	Est-ce plutôt au vert (campagne, nature, gazon, herbe) ?
10	Est-ce une activité de montagne ? à la neige ?
11	Est-ce plutôt en ville ? en stade ? équipement sportif ?
12	Est-ce une activité courte ou longue ?
13	Est-ce que l'activité peut être décomposée en plusieurs actions ?
14	En combien peut-on la décomposer ?
15	Peut-on prédire la suite du mouvement ?
16	Peut-on savoir quand le mouvement va se terminer (prédiction) ?
17	Peut-on savoir quand le mouvement est terminé (constat) ?
18	Quelle est la rapidité/l'intensité de l'action ?
19	Est-ce un mouvement régulier, ponctuel, unique, séquentiel ?
20	Est-ce un dialogue, une manifestation, une réunion ?
21	Y a-t-il besoin d'un objet spécifique ?
22	Utilise-t-on un moyen de locomotion ?
23	Le mouvement est-il localisé à une partie du corps en particulier ?
24	Plutôt antérieure ? postérieure ?
25	L'objet principal est-il un personnage ?
26	Y a-t-il un déplacement dans l'environnement ?
27	Y a-t-il un déplacement par rapport au sol ?
28	Le concept concerne-t-il un (ou des) objet(s) d'intérêt ?
29	Concerne-t-il un objet principal ?
30	Y a-t-il déformation pendant le mouvement ?
31	Y a-t-il des changements dans le mouvement ?
32	Le personnage se lève-t-il ?

Tableau C.1 — Liste des questions et leurs liaisons avec les briques.

Questions	Réponses	Liaisons
1	communication	ouverture thème communication
1	mouvement	ouverture thème mouvement
1	déplacement	ouverture thème déplacement
1	indéterminé	ouverture complète
2	un seul	no-1
2	deux	no-2
2	quelques	no-*
2	groupe	no-*
2	beaucoup	no-*
3	toujours	stip-qi-fort
3	souvent	stip-qi-fort
3	parfois	stip-qi-moyen ou stip-qi-fort
3	jamais	stip-qi-faible
3	ne sais pas	/
4	toujours	ouverture thème environnement
4	parfois	ouverture thème environnement
4	jamais	blocage thème environnement
4	ne sais pas	ouverture thème environnement
5	intérieur	lum-clair
5	extérieur	lum-foncé
5	les 2	lum-clair lum-foncé
5	ne sais pas	toutes lum
6	le jour	lum-clair
6	la nuit	lum-foncé
6	les 2	lum-clair lum-foncé
6	ne sais pas	toutes lum
7	oui	ouverture de questions
7	non	fermeture de questions
7	ne sais pas	ouverture de questions
8	aérien	col-bleu, col-blanc sip-faible
8	nautique	col-bleu sip-moyen sip-fort
8	aucun des 2	ouverture autre environnement
9	oui	col-vert
9	variable	col-vert
9	non	ouverture autre environnement
10	montagne	col-vert col-marron
10	neige	col-blanc
10	aucun des 2	ouverture autre environnement

Tableau C.2 — Liste des liaisons des questions 1 à 10.

Questions	Réponses	Liaisons
11	ville	col-gris l-moyen sip-fort
11	stade	col-marron col-bleu col-vert l-faible l-moyen sip-fort
11	équipement	col-marron col-bleu col-vert l-faible l-moyen sip-fort
11	aucun des trois	ouverture autre environnement
12	très courte	dur-court
12	courte	dur-court dur-moyen
12	plutôt long	dur-moyen dur-long
12	long	dur-long
12	variable	/
13	toujours	opérateur m
13	souvent	opérateur m
13	parfois	opérateur m (optionnel)
13	jamais	/
14	2	nbp-2 dur
14	3	nbp-3 dur
14	4 et plus	nb-n dur
15	toujours	séquence fixe
15	souvent	séquence fixe
15	parfois	séquence variable
15	jamais	pas de séquence fixe
15	ne sais pas	/
16	toujours	dur stabilité
16	souvent	dur stabilité
16	parfois	dur stabilité (facultative)
16	jamais	dur
16	ne sais pas	/
17	lente	fo-io-lent stip-qi-faible
17	plutôt lente	fo-io-lent stip-qi-faible stip-qi-moyen
17	moyenne	fo-io-moyen stip-qi-moyen
17	plutôt rapide	fo-io-rapide stip-qi-fort
17	rapide	fo-io-rapidestip-qi-fort
18	peu intense	stip-qi-faible
18	assez intense	stip-qi-moyen
18	plutôt intense	stip-qi-fort
18	très intense	stip-qi-fort

Tableau C.3 — Liste des liaisons des questions 11 à 18.

Questions	Réponses	Liaisons
19	régulier	dur opérateur(m) fixe
19	ponctuel	nbp dur opérateur(m)
19	unique	nbp-1 dur
19	séquentiel	opérateur(m) fixe
19	ne sais pas	/
20	dialogue	no-2 stip-qi-faible
20	réunion	no-* stip-qi-moyen
20	manifestation	stip-qi-fort
21	toujours	ouverture questions
21	souvent	ouverture questions
21	parfois	ouverture questions
21	jamais	fermeture questions
21	ne sais pas	/
22	toujours	fo-ii-rapide stip-qi-fort
22	souvent	fo-ii-rapide stip-qi-fort
22	parfois	fo-ii-rapide stip-qi-fort (facultatif)
22	jamais	/
22	ne sais pas	/
23	toujours	ouverture questions
23	souvent	ouverture questions
23	parfois	ouverture questions
23	jamais	fermeture questions
23	ne sais pas	/
24	antérieure	stip-oc-haut
24	postérieure	stip-oc-bas
24	les 2	stip-oc-haut stip-oc-bas
24	ne sais pas	/
25	toujours	taille-moyen taille-grand
25	souvent	taille-moyen taille-grand
25	parfois	/
25	jamais	/
25	ne sais pas	/
26	toujours	cam-mobile
26	souvent	cam-mobile
26	parfois	cam-mobile (facultatif)
26	jamais	/
26	ne sais pas	/

Tableau C.4 — Liste des liaisons des questions 19 à 27.

Questions	Réponses	Liaisons
27	toujours	fo-oo-* camera-o*
27	souvent	fo-oo-* camera-o*
27	parfois	fo-oo-* camera-o* (facultatif)
27	jamais	/
27	ne sais pas	/
28	aucun	no-0
28	un	no-1
28	deux	no-2
28	plusieurs	no-n
28	ne sais pas	/
29	toujours	no-1
29	souvent	no-1
29	parfois	no-1 (facultatif)
29	jamais	/
29	ne sais pas	/
30	toujours	c-f c-m c-F
30	souvent	c-f c-m c-F
30	parfois	c-f c-m c-F (facultatif)
30	jamais	/
30	ne sais pas	/
31	toujours	chg-n nbp-n
31	souvent	chg-n nbp-n
31	parfois	chg-n nbp-n (facultatif)
31	jamais	/
31	ne sais pas	
32	toujours	fo-oo-haut
32	souvent	fo-oo-haut
32	parfois	fo-oo-haut (facultatif)
32	jamais	/
32	ne sais pas	/

Tableau C.5 — Liste des liaisons des questions 27 à 32.

D Paramétrage des applications

Extracteurs	Paramètres	Valeurs
Caméra	type	mobile/fixe
SIP	σ_S	1.5
SIP	seuil	150
STIP	σ_S	1.5
STIP	σ_T	1.5
STIP	seuil	150
FO	nombre de points	400
FO	qualité minimale	0.01
FO	nombre pyramide	5
FO	distance minimale	0.01
FO	taille fenêtre	3
Détection d'objets (Motion2D)	modèle	AC
Détection d'objets (Motion2D)	itérations	300000
Détection d'objets (Background)	Taille minimale	10%
Détection d'objets (Background)	Filtrage	15
Hough lignes	rho	1
Hough lignes	theta	100
Hough lignes	seuillage	180
Hough cercles	distance minimale	1
Hough cercles	radius minimal	100
Hough cercles	radius maximal	200
Hough cercles	dp	2

Tableau D.1 — Liste des paramètres utilisés.

L'appel au module Motion 2D est effectué par la commande D.1.

```
motionconsapp_v2.exe -m AC -p ./ -f 0000 -e .png -i 300000 -b ./tmp/ -w
    ./support/ -r ./estimation.txt -v'
```

Listing D.1 — Appel du module Motion2D pour une estimation par un modèle à 6 paramètres

Bibliographie

P.E. AGRE : *Computation and Human Experience*. Cambridge University Press, 1997.

J.F. ALLEN : An interval-based representation of temporal knowledge. *Proceedings of IJCAL*, pages 221–226, 1981.

J.F. ALLEN : Maintaining knowledge about temporal intervals. *Communications of the ACM*, pages 832–843, November 1983.

A. AMIR, J. ARGILLANDER, M. CAMPBELL, A. HAUBOLD, G. IYENGAR, S. EBADOLLAHI, F. KANG, M.R. NAPHADE, A.P. NATSEV, J.R. SMITH, J. TESIC et T. VOLKMER : Ibm research trecvid-2005 video retrieval system. *Proceedings of TRECVID Workshop*, 2005.

M.A. ARBIB : *The Handbook of Brain Theory and Neural Networks*. The MIT Press, 1995.

L. AUDIBERT : *Bases de données de la modélisation au SQL*. 2009.

S. AYACHE : *Indexation de documents vidéos par concepts par fusion de caractéristiques audio, image et texte*. Thèse de doctorat, Institut National Polytechnique de Grenoble, 2007.

S. AYACHE et G. QUÉNOT : Evaluation of active learning strategies for video indexing. *Signal Processing : Image Communication*, 22:692–704, 2007.

S. AYACHE et G. QUÉNOT : Trecvid 2007 : Collaborative annotation using active learning. *Dans TRECVID 2007*, 2008a.

S. AYACHE et G. QUÉNOT : Video corpus annotation using active learning. *Dans In 30th European Conference on Information Retrieval (ECIR'08*, pages 187–198, 2008b.

J. AZÉ : *Extraction de Connaissances dans des Données Numériques et Textuelles*. Thèse de doctorat, Université de Paris-Sud 11, 16 Décembre 2003.

T.M. BAE, C.S. KIM, S.H. JIN, K.H. KIM, et Y.M. RO : Semantic event detection in structured video using hy- brid hmm/svm. *CIVR : Conference of Image and Video Retrieval*, pages 113–122, 2005.

R. BAEZA-YATES et B.RIBEIRO-NETO : *Modern Information Retrieval*. Addison Wesley Longman, 1999.

J. L. BARRON, D. J. FLEET et S. S. BEAUCHEMIN : Performance of optical flow techniques. *International Journal of Computer Vision*, pages 236–242, 1994.

H. BAY, A. E. T. TUYTELAARS et L. Van GOOL : Surf : Speeded up robust features. *ECCV*, pages 404–417, 2006.

N.J. BELKIN : Anomalous states of knowledge as a basis for information retrieval. *Canadian Journal of Information Science*, (5):133–143, 1980.

A. BENOIT, A. CAPLIER, B. DURETTE et J. HERAULT : Using human visual system modeling for bio-inspired low level image processing. *Computer Vision and Image Understanding*, 114:758–773, 2010.

H. BENOÎT : *La télévision numérique*. Dunod, 2002.

F. Le BER, J. LIEBER et A. NAPOLI : Utilisation d'une algèbre temporelle pour la représentation et l'adaptation de recettes de cuisine. *17ème Séminaire Raisonnement à partir de Cas*, pages 141–149, 2009.

P. BOUTHEMY, M. GELGON et F. GANANSIA : A unified approach to shot change detection and camera motion characterization. *Publication interne IRISA*, (1148), 1997.

Y. BOYKOV, O. VEKSLER et R. ZABIH : Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1222–1239, 2001.

F. BRÉMOND et M. THONNAT : Issues of representing context illustrated by video-surveillance applications. *Int. J. Hum.-Comput. Stud.*, 48(3):375–391, 1998.

S. BÜTTCHER, C.L.A. CLARKE et G.V.CORMACK. : *Information Retrieval : Implementing and Evaluating Search Engines*. MIT Press, 2010.

J.E. BYUN et T. NAGATA : Determining the 3-d pose of a flexible object by stereo matching of curvature representations. *Pattern Recognition*, 29(8):1297–1307, 1996.

A. CARBONARO : Ontology-based video retrieval in a semantic- based learning environment. *Journal of e-Learning and Knowledge Society*, 4(3):203–212, 2008.

R. CARMI et L. ITTI : Visual causes versus correlates of attentional selection in dynamic scenes. *Vision Research*, 46:4333–4345, 2006.

L. CARMINATI et J. BENOIS-PINEAU : Gaussian mixture classification for moving object detection in video surveillance environment. *Dans ICIP (3)*, pages 113–116, 2005.

C. CARSON, M. THOMAS, S. BELONGIE, J. M. HELLERSTEIN et J. MALIK : Blobworld : a system for region-based image indexing and retrieval. *Third Int. Conf. on Visual Information Systems*, 1614:509–516, 1999.

M. CERF, J. HAREL, W. EINHAUSER et C. KOCH : Predicting gaze using low-level saliency combined with face detection. *Neural Information Processing System*, 2007.

M. CHARHAD : *Modèles de Documents Vidéo basés sur le Formalisme des Graphes Conceptuels pour l'Indexation et la Recherche par le Contenu Sémantique*. Thèse de doctorat, Université de Grenoble, 2005.

M. CHEIN et M.-L. MUGNIER : Conceptual graphs : fundamental notions. *Revue d'intelligence artificielle*, 6:365–406, 1992.

R.J. CLARKE : Digital compression of still images and video. *London : Academic press*, pages 285–299, 1995.

R. CLOUARD : An ontology-based model for representing image processing objectives. *International Journal of Pattern Recognition and Artificial Intelligence*, 24(8):1181–1208, 2010.

H. COMER et B. DRAPER : Interest point stability prediction. 5815:315–324, 2009.

N. CORREIRA et T. CHAMBEL : Active video watching using annotation. *Dans ACM International Conference on Intelligent User Interfaces*, pages 151–154, 2000.

T.M. COVER et P.E. HART : Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

N. DALAL et B. TRIGGS : Histograms of oriented gradients for human detection. *CVPR*, pages 886–893, 2005.

N. DALAL, B. TRIGGS et C.SCHMID : Human detection using oriented histograms of flow and appearance. *CVPR*, 3954:428–441, 2006.

K. DARLINGTON : *The Essence of Expert Systems*. Pearson Education, 2000.

E. DIDAY, J. LEMAIRE, J.and POUGET et F. TESTU : *Éléments d'analyse de données*. 1982.

P. DOLLAR, V. RABAUD, G. COTTRELL et S. BELONGIE : Behaviour recognition via sparse spatio-temporal interest point detector. *VSPETS*, pages 65–72, 2005.

R. O. DUDA et P. E. HART : Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972.

E. ETIEVENT : *Assistance à l'indexation vidéo par analyse du mouvement*. Thèse de doctorat, Institut National des Sciences Appliquées de Lyon, 2004.

F. FAILLE : A fast method to improve the stability of interest point detection under illumination changes. *International Conference on Image Processing*, 4:2673–2676, 2004.

N. FATEMI et O. Abou KHALED : Coala : Content oriented audiovisual library access. *8th International Conference on Multimedia Modeling*, pages 59–71, 2001.

J. FERBER : *Objets et Agents : une étude des structures de représentation et de communications en Intelligence Artificielle*. Thèse de doctorat, Université Paris VI, 1989.

W.A.C. FERNANDO, C.N. CANAGARAJAH et D.R. BULL : Fade and dissolve detection in uncompressed and compressed video sequence. *IEEE International Conference on Image Processing*, 3:299–303, 1999.

W.A.C. FERNANDO et K.K. LOO : Abrupt and gradual scene transition detection in mpeg-4 compressed video sequences using texture and macroblock information. *Dans Computer Vision and Image Understanding*, 2004.

J. FOURNIER, M. CORD et S. PHILIPP-FOLIGUET : Retin : A content- based image indexing and retrieval system. *Pattern Analysis and Applications*, 4(2-3):153–173, 2001.

B. FRANCOIS, T. MONIQUE et Z. MARCOS : Video understanding framework for automatic behavior recognition. *Behavior Research Methods*, 3(38):416–426, 2006.

E. GALMAR et B. HUET : Analysis of vector space model and spatiotemporal segmentation for video indexing and retrieval. *Dans CIVR 2007, ACM International Conference on Image and Video Retrieved, July 9-11 2007, Amsterdam, The Netherlands*, pages 433–440, 2007.

J.-P. GAMBOTTO : Segmentation spatio-temporelle de séquences d'images. *12ème Colloque sur le traitement du signal et des images*, 1989.

B. GIAI-CHECA, P. BOUTHEMY et T. VIEVILLE : Détection d'objets en mouvement. Rapport technique INRIA-RR - 1906, INRIA, 1993.

J.C. GIARRATANO et G.RILEY : *Expert Systems, Principles and Programming*. Course Technology, 1998.

P. GOSSELIN : *Méthodes d'apprentissage pour la recherche de catégories dans des bases d'images*. Thèse de doctorat, Université de Cergy-Pontoise, 2005.

H. GOËAU : *Structuration de collections d'images par apprentissage actif crédibiliste*. Thèse de doctorat, Université de Grenoble, 2009.

M. GRABISCH et AL. : Évaluation subjective. *Méthodes, Applications et Enjeux. Les Cahiers des Clubs CRIN*, 1997.

A. GUÉRIN-DUGUÉ, C. BIERNACKI et J. HÉRAULT : Statistical modelling for image retrieval using a biological model of the perceptive colour space. *Proceedings of ICIP*, 1:209–212, 2001.

M. GUIRONNET : *Méthodes de résumé de vidéo à partir d'informations bas niveau, du mouement de caméra et de l'attention visuelle*. Thèse de doctorat, Université de Grenoble, 2006.

C. HARRIS et M.J. STEPHENS : A combined corner and edge detector. *Alvey Vision Conf.*, pages 147–151, 1988.

W.J. HENG et K.N. NGAN : The implementation of object-based shot boundary detection using edge tracing and tracking. *IEEE International Symposium on Circuits and Systems*, 1999.

W.J. HENG et K.N. NGAN : High accuracy flashlight scene determination for shot boundary detection. *Signal Processing : Image Communication*, 18(3):203–219, 2003.

J.-M. HOC : L'extraction des connaissances et l'aide à l'activité humaine. *Intellectica*, 12 (33–64), 1995.

J. HUANG, Z. LIU, Y. WANG, Y. CHEN et E. WONG : Integration of multimodal features for video scene classification based on hmm. *Proceedings of IEEE Workshop on Multimedia Signal Processin*, pages 53–58, 1999.

J. IGNIZIO : *Introduction to Expert Systems*. Mcgraw-Hill College, 1991.

B. IONESCU : *Caractérisation symbolique de séquences d'images - application aux films d'animation*. Thèse de doctorat, Université de Savoie, 2007.

L. ITTI, C. KOCH et E. NIEBUR : A model of salincy-based visual attention for rapid scene analysis. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20:1254–1259, 1998.

P. JACKSON : *Introduction to Expert Systems*. Addison Wesley, 1998.

A. JAIN, P. DUIN et J. MAO : Statistical pattern recognition : A review. *IEEE Transactions on PAMI*, 22:4–37, 2000.

N.H. KIM et A.C. BOVIK : A contour-based stereo matching algorithm using disparity continuity. *Pattern Recognition*, 21(5):505–514, 1988.

L. KITCHEN et R. ROSENFELD : Gray-level corner detection. *Pattern Recognition letters*, 1:95–102, 1982.

A. KLASER, M. MARSZALEK et C. SCHMID : A Spatio-Temporal Descriptor Based on 3D-Gradients. *Dans British Machine Vision Conference*, Leeds Royaume-Uni, 09 2008. URL `http://hal.inria.fr/inria-00514853/PDF/KlaserMarszalekSchmid-BMVC08-3DGradientDescriptor.pdf`.

F. KOKKORAS, H. JIANG, I. VLAHAVAS, A.K. ELMAGARMID, E.N. HOUSTIS et W.G. AREF : Smart videotext : a video data model based on conceptual graphs. *Multimedia Systems*, 8:328–338, 2002.

H.C. KWAN : *Fast Motion Estimation Techniques for Video Compression*. Thèse de doctorat, City University of Hong Kong, 1998.

R. LAGANIÈRE, R. BACCO, A. HOCEVAR, P. LAMBERT, G. PAÏS et B.E. IONESCU : Video summarization from spatio-temporal features. *ACM*, pages 144–148, 2008.

L.T. LAN, B. ALAIN, T. MONIQUE et B. FRANCOIS : A framework for surveillance video indexing and retrieval. *In International Workshop on Content Based Multimedia Indexing*, pages 338–345, 2008.

I. LAPTEV : On space-time interest points. *International Journal of Computer Vision*, pages 432–439, 2005.

I. LAPTEV, B. CAPUTO, C. SCHULDT et T. LINDEBERG : Local velocity-adapted motion events for spatio-temporal recognition. *Computer Vision and Image Understanding*, (108):207–229, 2007a.

I. LAPTEV, B. CAPUTO, C. SCHULDT et T. LINDEBERG : Local velocity-adapted motion events for spatio-temporal recognition. *Computer Vision and Image Understanding*, (107):207–229, 2007b.

I. LAPTEV et T. LINDEBERG : Space-time interest points. *ICCV'03*, pages 432–439, 2003.

D. LARLUS : *Création et utilisation de vocabulaires visuels pour la catégorisation d'images et la segmentation de classes d'objets*. Thèse de doctorat, INPG, 2008.

F. LEBER, G. LIGOZAT et O. PAPINI : *Raisonnements sur l'espace et le temps : des modèles aux applications*. 2007.

R. LIENHART : Reliable transition detection in videos : A survey and practitiner's guide. *MRL, Intel Corporation*, 2001.

C.-Y. LIN, L. TSENG et J.R. SMITH : Videoannex annotation tool.

D. G. LOWE : Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, pages 91–110, 2004.

B.D. LUCAS et T. KANADE : An iterative image registration technique. *IJCAI'81*, pages 674–679, 1981a.

B.D. LUCAS et T. KANADE : An iterative image registration technique with an application to stereo vision. *Proceedings of Imaging understanding workshop*, pages 121–130, 1981b.

M. C. MAMLOUK, A. A. YOUNES, H. AKDAG et I. TRUCK : Extraction des couleurs dominantes d'une image. *SETIT*, pages 574–588, 2007.

S. MARAT, T. Ho PHUOC, L. GRANJON, N. GUYADER, D. PELLERIN et A. GURIN-DUGUÉ : Modelling spatio-temporal saliency to predict gaze direction for short videos. *International Journal of Computer Vision*, 82(3):231–243, 2009.

N. MARTEL-BRISSON et A. ZACCARIN : Moving cast shadow detection from a gaussian mixture shadow model. *Dans CVPR '05 : Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 643–648, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2.

C.M. MASCIOCCHI, S. MIHALAS, D. PARKHURST et E. NIEBUR : Everyone knows what is interesting : Salient locations which should be fixated. *Journal of Vision*, 9(11), 2009.

A. M. MCIVOR : Background subtraction techniques. *In Proc. of Image and Vision Computing*, 4:3099–3104, 2000.

A. MICHARD : *XML, Langage et Applications*. Eyrolles, Paris, 1998.

B. MOGHADDAM, H. BIERMANN et D. MARGARITIS : Image retrieval with local and spatial queries. *International Conference on Image Processing*, 2:542–545, 2000.

F. MOKHTARIAN et A. MACKWORTH : Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):34–43, 1986.

H.P. MORAVEC : *Obstacle avoidance and navigation in the real world by a seeing robot rover*. Thèse de doctorat, Stanford artificial intelligence laboratory, 1980.

P. MULLER et D.R. INSUA : Issues in bayesian analysis of neural network models. *Neural Computation*, 10(571–592), 1995.

V.S. MURTHY et AL. : Content based image retrieval using hierarchical and k-means clustering techniques. *International Journal of Engineering Science and Technology*, 2010.

V.S. MURTHY, E.Vamsidhar J.N. SWARUP, V.R. KUMAR et P.SANKARA : Randomized locality sensitive vocabularies for bag-of-features model. *ECCV*, 2010.

J-M. ODOBEZ et P. BOUTHEMY : Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348–365, 1995.

O.K. OTSUJI, T. YOSHINOBU et O. YUJI : Video browsing using brightness data. *Visual Communications and Image Processing '91 : Image Processing*, 1606:980–989, 1991.

J. OUYANG, J. LI et Y. ZHANG : Ontology based sports video annotation and summary. *Advanced Workshop on Content Computing*, 3309:499–508, 2004.

T. PAWAR, N.S. ANANTAKRISHNAN, S. CHAUDHURI et S.P. DUTUUAGUPTA : Transition detection in body movement activities for wearable ecg. *IEEE transactions on biomedical engineering*, 54(2)(6):1149–1152, 2007.

G. PAÏS : *Analyse conjointe texte et image pour la caractérisation de films d'animation*. Thèse de doctorat, Université de Savoie, 2009.

G. PAÏS, F. DELOULE, D. BEAUCHÊNE et P. LAMBERT : Analyse texte et image pour la caractérisation de l'activité dans les films d'animation. *INFORSID*, (243–258), 2009.

R.J. PETERS et L. ITTI : Applying computational tools to predict gaze direction in interactive visual environments. *ACM Trans. Appl. Percept.*, 5(2):1–19, 2008. ISSN 1544-3558.

R.J. PETERS, A. LYER, L. ITTI et C. KOCH : Components of bottom up gaze allocation in natural images. *Vision Research*, 45:2397–2416, 2005.

C. M. PRIVITERA et L. W. STARK : Algorithms for defining visual regions-of-interest : Comparison with eye fixations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:970–982, 2000.

A.K. PUJARI, G.V. KUMARI et A. SATTAR : Indu : an interval and duration network. *Australian Joint Conf. on Artificial Intelligence*, pages 291–303, 1999.

E. RAMASSO : *Reconnaissance de séquences d'états par le Modèle des Croyances Transférables et application à l'analyse de vidéos d'athlétisme*. Thèse de doctorat, University Joseph Fourier of Grenoble, 2007.

F. RANCHIN et F. DIBOS : Segmentation d'objets en mouvement par utilisation du flot optique. *ORASIS*, 2005.

K. RAVISHANKAR, B. PRASAD, S. GUPTA et K. BISWAS : Dominant color region based indexing for cbir. *Proceedings of ICIAP*, pages 887–894, 1999.

H. REHATSCHAK et R. MÜLLER : A generic annotation for video databases. *VISUAL*, pages 383–390, 1999.

B.J. ROBBINS et R. OWENS : 2d feature detection via local energy. *Image and Vision Computing*, 15:353–368, 1997.

C. SCHMID : Constructing models for content-based image retrieval. *Computer Vision and Pattern Recognition*, 2:39–45, 2001.

C. SCHMID, R. MOHR et C. BAUCKHAGE : Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

J. SERRA : *Image Analysis and Mathematical Morphology Vol. I*. Ac. Press, London, 1982.

J. SERRA : *Image Analysis and Mathematical Morphology Vol. II*. Ac. Press, London, 1988.

J. SHI et C. TOMASI : Good features to track. *9th IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

A. SIMAC-LEJEUNE, S. MARAT, D. PELLERIN, P. LAMBERT, M. ROMBAIT et N. GUYADER : Relevance of interest points for eye position prediction on videos. *ICVS*, 5815:325–334, 2009.

A. SIMAC-LEJEUNE, M. ROMBAUT et P. LAMBERT : Points d'intérêt spatio-temporels pour la détection de mouvements dans les vidéos. *MajecSTIC*, 2010a.

A. SIMAC-LEJEUNE, M. ROMBAUT et P. LAMBERT : Spatio-temporal block model for video indexation assistance. *KDIR*, 2010b.

A.W. M. SMEULDERS, M. WORRING, S. SANTINI, A. GUPTA et R. JAIN : Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000.

J. R. SMITH et B. LUGEON : A visual annotation tool for multimedia content description. *Proc. SPIE Photonics East, Internet Multimedia Management Systems*, 4210:49–59, 2000.

C.G.M. SNOEK, M. WORRING, J.-M.G EUSEBROEK, D.C. KOELMA, F.J. SEINSTRA et A.W.M. SMEULDERS : The semantic pathfinder for generic news video indexing. *Proceedings of ICME*, pages 1469–1472, 2006.

J.F. SOWA : *Conceptual structures : information processing in mind and machine*. 1984.

N. SUVONVORN : *Mise en Correspondance d'images pour l'analyse du Mouvement et la stéréorvision*. Thèse de doctorat, Université de Paris-Sud-Orsay, 1999.

C. TOMASI et T. KANADE : Detection and tracking of point features. *Technical report CMU-CS-91-132*, 1991.

M.M. ULLAH, S.N. PARIZI et I. LAPTEV : Improving bag-of-features action recognition with non-local cues. *BMVC'10*, pages 95.1–95.11, 2010.

L. VALET : *Un système flou de fusion coopérative : application au traitement d'images naturelles*. Thèse de doctorat, Université de Savoie, 2001.

P. VANNOORENBERGHE : Bouclage de pertinence par arbres de décision crédibilistes en indexation d'images par le contenu. *Dans Actes Compression et Représentation des Signaux Audiovisuels (CORESA)*, 2004.

H.S. VARANDAS : *Compressed domain H.264/AVC shot detection*. Thèse de doctorat, Universidade Técnica de Lisboà, 2008.

M. VILAIN et H. KAUTZ : Constraint propagation algorithms for temporal reasoning. *AAAI86 Proceedings*, pages 377–382, 1986.

A. WALKER et AL. : *Knowledge Systems and Prolog*. Addison-Wesley, 1990.

X. WAN et C.-C. J. KUO : A new approach to image retrieval with hierarchical color clustering. *IEEE Trans. Circuits Systems Video Technology*, 8(5):628–643, 1998.

P.D. WASSERMAN : *Neural computing theory and practice*. Van Nostrand Reinhold, 1989.

H.H. YU et W. WOLF : A hierarchical multiresolution video shot transition detection scheme. *Computer Vision and Image Understanding*, 75(1-2):196–213, 1999.

L. ZELNIK-MANOR et M. IRANI : Event-based analysis of video. *Computer Vision and Pattern Recognition*, pages 123–130, 2001.

H. ZHONG, S. JIANBO et M. VISONTAI : Detecting unusual activity in video. *Computer Vision and Pattern Recognition*, 2:819–826, 2004.

Y. ZHOU, Y. CAO, L. ZHANG et H.ZHANG : An svm-based soccer video shot classification. *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, 9:5398–5403, 2005.

Liste des figures

Liste des tableaux

Liste des publications

Conférences internationales

2010

Spatio-temporal block model for video indexation assistance

Alain Simac-Lejeune, Michèle Rombaut et Patrick Lambert

International Conference on Knowledge Discovery and Information Retrieval (KDIR), Valence, Espagne

Résumé : In the video indexing framework, we have developed an assistance system for the user to define a new concept as semantic index according to the features automatically extracted from the video. Because the manual indexing is a long and tedious task, we propose to focus the attention of the user on pre-selected prototypes that a priori correspond to the concept. The proposed system is decomposed in three steps. In the first one, some basic spatio-temporal blocks are extracted from the video, each associated to a particular property of one feature. In the second step, a Question - Answer system allows the user to define links between basic blocks in order to define concept blocks. And finally, some concept blocks are extracted and proposed as prototypes of the concepts. In this paper, we present this assistance system, the block structure and the global software architecture, illustrated by an example of video indexing that corresponds to the "running" concept in athletic videos.

2009

Relevance of Interest points for eye position prediction on videos

Alain Simac-Lejeune, Sophie Marat, Denis Pellerin, Patrick Lambert, Michèle Rombaut et Nathalie Guyader

International Conference of Vision System (ICVS), Liège, Belgique

Résumé : This paper tests the relevance of interest points to predict eye movements of subject when freely viewing video sequences. Moreover the paper compared the eye positions of subjects with interest maps obtained using two classical interest point detectors : one spatial and one space-time. We found that in function of the video sequence, and more especially in function of the motion inside the sequence, the spatial or the space-time interest point detector is more or less relevant to predict eye movements.

Conférences nationales

2010

Points d'intérêt spatio-temporels pour la détection de mouvements dans les vidéos
Alain Simac-Lejeune, Michèle Rombaut et Patrick Lambert
MAnisfestation des JEunes Chercheurs en Sciences et Technologies de l'Information et de la Communication (MajecSTIC), Bordeaux, France
Résumé : Among all the features which can be extracted from videos, we propose to use Space-Time Interest Points (STIP). STIP are particularly interesting because they are simple and robust low-level features providing an efficient characterization of moving objects within videos. In this paper, after defining STIP and after giving some of their properties, we will use STIP to detect moving objects and to characterize specific changes in the movements of these objects. Proposed results are obtained from very different types of videos, namely athletic videos and animation movies.