



HAL
open science

Indexation et recherche de vidéo pour la vidéosurveillance

Thi Lan Le

► **To cite this version:**

Thi Lan Le. Indexation et recherche de vidéo pour la vidéosurveillance. Interface homme-machine [cs.HC]. Université Nice Sophia Antipolis, 2009. Français. NNT: . tel-00393866

HAL Id: tel-00393866

<https://theses.hal.science/tel-00393866>

Submitted on 10 Jun 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE DE NICE-SOPHIA ANTIPOLIS

ECOLE DOCTORALE STIC
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THESE EN COTUTELLE INTERNATIONALE

pour obtenir le titre de

Docteur en Sciences

de l'Université de Nice-Sophia Antipolis
et
de l'Institut Polytechnique de Hanoi

Mention : Automatique, traitement du signal et des images

présentée et soutenue par

Thi-Lan LE

INDEXATION ET RECHERCHE DE VIDÉO POUR LA VIDÉOSURVEILLANCE

Thèse dirigée par M^{me} Monique THONNAT et M. Alain BOUCHER

au sein du projet PULSAR, INRIA Sophia Antipolis
et du centre de recherche international MICA
(Multimedia, Informations, Communication et Application), Hanoi

le 3 février 2009

Jury :

M. Bernard Merialdo	Pr. Eurecom, France	Président
M. Philippe Joly	Pr. IRIT, Toulouse, France	Rapporteur
M. Patrick Gros	DR, INRIA Rennes, France	Rapporteur
M ^{me} Monique Thonnat	DR, INRIA Sophia Antipolis, France	Directrice de thèse
M. Alain Boucher	Pr. IFI-AUF Hanoi, VietNam	Co-directeur de thèse
M. Eric Castelli	Dr, MICA, Hanoi, VietNam	Examineur

Remerciement

Je souhaiterais tout d'abord remercier les membres du centre MICA (présente et passée) qui m'ont accueilli depuis mon master. Parmi eux, je tiens tout particulièrement à remercier Nguyen Trong Giang, Pham Thi Ngoc Yen, Eric Castelli d'avoir pris le pari de m'embaucher en tant que membre permanent du centre MICA.

Je tiens à exprimer tout ma gratitude à mes deux directeurs de thèse, Monique Thonnat et Alain Boucher. Merci à vous deux de m'avoir proposé ce sujet et de m'avoir fait confiance pendant la thèse. Merci également pour toutes ces discussions, questions, conseils, critiques nombreuses et constructives, vos soutiens moraux dans les périodes difficiles.

Un grand merci à Patrick Gros et ainsi qu'à Philippe Joly d'avoir bien voulu lire les quelques 200 pages de ce manuscrit et rapporté sur cette thèse. Merci pour vos remarques et questions constructives qui m'ouvrent de nouvelles pistes pour compléter et poursuivre ce travail. Merci également à Bernard Merialdo d'avoir accepté la charge de président de mon jury.

Je remercie aussi l'ensemble de l'équipe PULSAR (présente et passée) pour avoir fait de ces trois ans une expérience inoubliable dans une ambiance toujours stimulante et joyeuse. Je tiens tout particulièrement à remercier François Brémond pour les discussions très constructives. Merci Catherine Martin, assistante du projet et aide au combien importante pour tous les doctorants. Merci à Guillaume, Bernard, Valéry, Nadia, Ikhlef, Vincent pour les discussions à la fois scientifiques mais aussi toutes autres que l'on a pu avoir. Grand merci à Luis et Étienne pour le travail de préparation des données.

Je tiens à remercier tous mes amis vietnamiens à Nice et à Sophia Antipolis : Hau, Anh-Tuan, Chi-Anh, Nghia-Thuy, Sy Hung, Van, Phu. Merci Hau pour tout ce que tu as fait pour moi pendant 3 ans. Je suis très heureuse de te faire reconnaître. Tu étais toujours à mon côté, les difficiles moments ainsi qu'aux joyeux moments. Merci Sophie et Paul pour les cours de français et pour les repas chez vous. Grand merci Thomas Martin, mon grand frère, pour les discussions et les moments qu'on a partagé pendant des années.

J'adresse toute ma gratitude à mes parents, Duc, mes soeurs Hien et Huong, mon neveu Chien, mes nièces Thao, Quynh et Linh. Merci pour les amours et les soutiens que vous m'avez donné. Sans lesquels, cette thèse n'aurait pas été aussi loin.

Enfin, je ne pourrai clore ces remerciements sans exprimer ma gratitude à Hai, Yen, "mon fils" Hoang-Anh et Quyen, Su, Manh (mes amis au lycée Luong Dac Bang) pour leurs soutiens.

Résumé

L'objectif de cette thèse est de proposer une approche générale pour l'indexation et la recherche de vidéos pour la vidéosurveillance. En se basant sur l'hypothèse que les vidéos sont prétraitées par un module d'analyse vidéo, l'approche proposée comprend deux phases : la phase d'indexation et celle de recherche.

Afin d'utiliser les résultats de différents modules d'analyse vidéo, un modèle de données comprenant deux concepts, objets et événements, est proposé. La phase d'indexation visant à préparer des données déterminées dans ce modèle de données effectue trois tâches. Premièrement, deux nouvelles méthodes de détection des blocs représentatifs de la tâche représentation d'objets déterminent un ensemble de blocs associés à leurs poids pour chaque objet. Deuxièmement, la tâche extraction de descripteurs consiste à analyser des descripteurs d'apparence et aussi temporels sur les objets indexés. Finalement, la tâche indexation calcule les attributs des deux concepts et les stocke dans une base de données.

La phase de recherche commence avec une requête de l'utilisateur et comprend quatre tâches. Dans la tâche formulation de requêtes, afin de permettre à l'utilisateur d'exprimer ses requêtes, un nouveau langage est proposé. La requête est traitée par la tâche analyse syntaxique. Une nouvelle méthode dans la tâche mise en correspondance permet de retrouver efficacement les résultats pertinents. Deux méthodes dans la tâche retour de pertinence permettent d'interagir avec l'utilisateur afin d'améliorer les résultats de recherche.

Dans le but d'évaluer la performance de l'approche proposée, nous utilisons deux bases de vidéos dont l'une provenant du projet CARETAKER et l'autre provenant du projet CAVIAR. Les vidéos du projet CARETAKER sont analysées en utilisant la plate-forme VSIP de l'équipe PULSAR alors que les vidéos du projet CAVIAR sont manuellement annotées. La méthode de détection des blocs représentatifs améliore la performance d'une méthode dans l'état de l'art. L'utilisation du langage de requêtes montre qu'il permet d'exprimer de nombreuses requêtes à différents niveaux. La méthode de mise en correspondance obtient de meilleurs résultats en comparaison avec deux méthodes de l'état de l'art. Les résultats expérimentaux montrent que l'approche proposée retrouve efficacement les objets d'intérêt et les événements complexes.

Mots clefs : indexation de vidéos, recherche de vidéos, langage de requêtes, mise en correspondance, vidéosurveillance

Abstract

The goal of this work is to propose a general approach for surveillance video indexing and retrieval. Based on the hypothesis that videos are preprocessed by an external video analysis module, this approach is composed of two phases : indexing phase and retrieval phase.

In order to profit from the output of various video analysis modules, a general data model consisting of two main concepts, objects and events, is proposed. The indexing phase that aims at preparing data defined in the data model performs three tasks. Firstly, two new key blob detection methods in the object representation task choose for each detected object a set of key blobs associated with a weight. Secondly, the feature extraction task analyzes a number of visual and temporal features on detected objects. Finally, the indexing task computes attributes of the two concepts and stores them in the database.

The retrieval phase starts with a user query and is composed of 4 tasks. In the formulation task, user expresses his query in a new rich query language. This query is then analyzed by the syntax parsing task. A new matching method in the matching task aims at retrieving effectively relevant results. Two proposed methods in the relevance feedback task allow to interact with the user in order to improve retrieved results.

The key blob detection method has improved results of one method in the state of the art. The analysis of query language usage shows that many queries at different abstraction levels can be expressed. The matching method has proved its performance in comparison with two other methods in the state of the art. The complete approach has been validated on two video databases coming from two projects : CARETAKER and CAVIAR. Videos of the CARETAKER project are analyzed by the VSIP platform of the Pulsar team while videos coming from CAVIAR project are manually annotated. Experiments have shown how the proposed approach is efficient and robust to retrieve the objects of interest and the complex events from surveillance videos.

Keywords : video indexing, video retrieval, query language, matching, videosurveillance

Table des matières

Liste des tableaux	ix
Table des figures	xviii
1 Introduction	1
1.1 Motivations et objectifs	1
1.1.1 Contexte	2
1.1.2 Applications	3
1.1.3 Problèmes et Objectifs	3
1.2 Approche proposée	6
1.2.1 Hypothèses	6
1.2.2 Questions ouvertes	6
1.2.3 Contributions	7
1.3 Structure du manuscrit	8
2 État de l’art	11
2.1 Indexation et recherche d’images	11
2.1.1 Recherche locale vs recherche globale	11
2.1.2 Ontologie	17
2.1.3 Retour de pertinence	19
2.1.4 Discussions	25
2.2 Indexation et recherche de vidéos	25
2.2.1 Deux types de vidéos et terminologies	25
2.2.2 Indexation et recherche de vidéos scénarisées	26
2.2.3 Indexation et recherche de vidéos non scénarisées	32
2.2.4 Discussions	38
2.3 Indexation et recherche de vidéos pour la vidéosurveillance	38
2.3.1 Introduction	38
2.3.2 Approches sans retour de pertinence	39
2.3.3 Approches basées sur le retour de pertinence	54
2.3.4 Discussions	57
2.4 Conclusion et problèmes ouverts	57
2.4.1 Problèmes ouverts	58
2.4.2 Relation entre l’indexation et la recherche d’images et notre travail	60
2.4.3 Relation entre l’indexation et la recherche de vidéos scénarisées et notre travail	61

3	Approche proposée	63
3.1	Description générale de l'approche proposée	63
3.2	Interface analyse vidéo/indexation	65
3.3	La phase d'indexation	65
3.3.1	Modèle de données	65
3.3.2	Représentation d'objets	66
3.3.3	Extraction de descripteurs	66
3.3.4	Indexation	67
3.4	La phase de recherche	67
3.4.1	Formulation des requêtes	67
3.4.2	Sélection d'exemples	68
3.4.3	Analyse syntaxique	68
3.4.4	Extraction de descripteurs	68
3.4.5	Mise en correspondance	68
3.4.6	Affichage des résultats	69
3.4.7	Retour de pertinence	69
3.5	Conclusion	70
4	Indexation de vidéos de vidéosurveillance	71
4.1	Introduction	71
4.2	Modèle de données	71
4.2.1	Objets	71
4.2.2	Événements	75
4.2.3	Discussions	76
4.3	Extraction de descripteurs d'apparence	76
4.3.1	Analyse de la couleur	77
4.3.2	Analyse du contour	79
4.3.3	Analyse des points d'intérêt	82
4.3.4	Discussions	86
4.4	Extraction de descripteurs temporels	88
4.4.1	Représentation des trajectoires	88
4.4.2	Discussions	91
4.5	Représentation d'objets mobiles	92
4.5.1	Introduction	92
4.5.2	Détection des blobs représentatifs	93
4.5.3	Discussions	98
4.6	Conclusion	99
5	Recherche de vidéos de vidéosurveillance	101
5.1	Introduction	101
5.2	Scénarios de recherche	101
5.3	Mise en correspondance des éléments indexés	102
5.4	Mise en correspondance entre des objets basée sur les descripteurs d'apparence	102

5.4.1	Mise en correspondance entre des blobs basée sur les descripteurs d'apparence	103
5.4.2	Mise en correspondance entre des objets par les descripteurs d'apparence	105
5.4.3	Discussions	109
5.5	Mise en correspondance des objets basée sur les descripteurs temporels	110
5.5.1	La distance d'édition	110
5.5.2	Mise en correspondance entre des trajectoires au niveau numérique	111
5.5.3	Mise en correspondance entre des trajectoires au niveau symbolique	112
5.5.4	Discussions	112
5.6	Relations temporelles	113
5.7	Langage de requêtes	114
5.7.1	Introduction	114
5.7.2	Syntaxe	115
5.7.3	Exemples de requêtes exprimées par le langage proposé	116
5.7.4	Vers une comparaison avec l'approche de Ghanem et al.	117
5.7.5	Discussions	119
5.8	Recherche interactive et Retour de pertinence	120
5.8.1	Introduction	120
5.8.2	Retour de pertinence basé sur plusieurs images d'exemple . . .	120
5.8.3	Retour de pertinence par les machines à vecteurs de support . .	121
5.8.4	Discussions	123
5.9	Conclusion	123
6	Implémentation et évaluation	125
6.1	Implémentation	125
6.2	Modules d'analyse vidéo	125
6.2.1	Plate-forme VSIP (Video Surveillance Interpretation Platform)	126
6.2.2	Annotation manuelle	127
6.2.3	Mesures d'évaluation des performances des modules d'analyse vidéo	128
6.3	Bases des données	129
6.3.1	Bases des trajectoires	129
6.3.2	Bases des vidéos	129
6.4	Mesures d'évaluation des performances de l'approche d'indexation et de recherche d'information	134
6.4.1	Rappel et précision	136
6.4.2	Rang normalisé moyen	136
6.4.3	Matrice de confusion	137
6.4.4	Discussions	137
6.5	Analyse de performance de l'approche proposée	138
6.6	Utilisation du langage de requêtes	139

6.6.1	Analyse d'effort demandé à l'utilisateur et expressivité du langage	139
6.6.2	Requêtes concernant des images d'exemple et des objets (C1)	139
6.6.3	Requêtes concernant des objets (C2)	141
6.6.4	Requêtes concernant des événements (C3)	142
6.6.5	Requêtes concernant des objets et des événements (C4)	142
6.6.6	Requêtes concernant des images d'exemple, des objet et des événement (C5)	143
6.6.7	Facilité du langage de requêtes	144
6.6.8	Discussions	144
6.7	Évaluation de la détection des blobs représentatifs	144
6.7.1	Résultats de la détection des blobs représentatifs	145
6.7.2	Comparaison de la méthode de détection des blobs représentatifs basée sur le regroupement des blobs et celle de Ma et al.	145
6.7.3	Discussions	153
6.8	Évaluation de la recherche d'objets	153
6.8.1	Analyse de difficultés rencontrées dans la recherche d'objets	155
6.8.2	Comparaison de notre méthode de mise en correspondance avec celle de Ma et al.	155
6.8.3	Comparaison de notre méthode avec celle de Calderara et al.	163
6.8.4	Évaluation des descripteurs visuels pour la recherche d'objets	170
6.8.5	Recherche d'objets basée sur les trajectoires	179
6.8.6	Discussions	179
6.9	Évaluation de recherche d'objets et d'événements	180
6.9.1	Résultats de recherche avec les requêtes de la catégorie 1	181
6.9.2	Résultats de recherche avec les requêtes de la catégorie 2	181
6.9.3	Résultats de recherche avec les requêtes de la catégorie 3	181
6.9.4	Résultats de recherche avec les requêtes de la catégorie 4	182
6.9.5	Résultats de recherche avec les requêtes de la catégorie 5	183
6.9.6	Discussions	184
6.10	Évaluation du retour de pertinence	184
6.10.1	Retour de pertinence basé sur plusieurs images d'exemple	185
6.10.2	Retour de pertinence basé sur les SVM à une classe	191
6.10.3	Discussions	192
6.11	Conclusion	192
7	Conclusions et perspectives	195
7.1	Résumé des contributions	195
7.2	Limitations	196
7.3	Discussions	196
7.3.1	Relation entre l'indexation et la recherche de vidéos et les modules d'analyse vidéo	196
7.3.2	Descripteurs visuels	197

7.3.3	Mise en correspondance entre des objets	197
7.4	Perspectives	198
7.4.1	Perspectives à court terme	198
7.4.2	Perspectives à long terme	199
A	Langage de requêtes	205
B	Implémentation de l'approche proposée	207
B.1	Logiciels utilisés	207
B.1.1	Librairie ltilib	207
B.1.2	Détecteurs des points d'intérêt	209
B.1.3	Regroupement agglomératif	209
B.1.4	Librairie du projet PULSAR	209
B.1.5	Librairie de SVM	209
B.2	Phase d'indexation	209
B.2.1	Extraction de descripteurs	209
B.2.2	Détection des blobs représentatifs	209
B.3	Phase de recherche	209
B.3.1	Langage de requêtes	209
B.3.2	Calcul de distance EMD	210
B.3.3	Retour de pertinence	210
B.4	Algorithme de Calderara et al. et celui de Ma et al.	210
	Bibliographie	211

Liste des tableaux

2.1	Origine et nature des informations qui peuvent être exploitées par le retour de pertinence [Crucianu 2004].	21
2.2	7 descripteurs d'une activité [Hu 2007].	48
2.3	4 descripteurs d'un modèle d'activité [Hu 2007].	48
2.4	Résumé des approches pour l'indexation et la recherche de vidéos pour la vidéosurveillance selon les niveaux auxquels l'indexation et la recherche effectuent : niveau objets (Objets) ou/et niveau événements (Événements), la présence du langage de requête (Langage) et la présence du retour de pertinence (RP).	58
4.1	9 attributs des objets mobiles.	72
4.2	2 attributs des images d'exemple.	73
4.3	6 attributs des événements.	76
5.1	Distances entre des paires de blobs en utilisant les matrices de covariance, les colonnes sont les blobs de l'objet 1064, les lignes sont les blobs de l'objet 1065. La distance de Hausdorff entre deux objets est déterminée par la distance entre le blob 1 de l'objet 1064 et le blob 4 de l'objet 1065.	109
5.2	7 relations temporelles entre deux intervalles de temps I_1 et I_2	113
6.1	Vidéos provenant du projet CARETAKER.	130
6.2	Résultats de l'analyse des vidéos provenant du projet CARETAKER au niveau objets.	131
6.3	Résultats de l'analyse de la vidéo CARE_1 provenant du projet CARETAKER au niveau événements.	132
6.4	Résultats de l'analyse de la vidéo CARE_2 provenant du projet CARETAKER au niveau événements.	133
6.5	Valeurs des mesures d'évaluation de la plate-forme VSIP pour la vidéo CARE_6. La valeur de persistance montre qu'en moyenne deux objets détectés correspondent à un objet réel alors que la valeur de confusion indique que certains objets détectés correspondent à plus d'un objet réel.	134
6.6	Dix vidéos provenant du projet CAVIAR.	135

6.7	Analyse de l'effort demandé pour chaque catégorie de requête correspondant à 3 niveaux d'analyse vidéo. Le premier niveau (N1) contient la détection et le suivi d'objets. Le deuxième niveau (N2) rajoute la classification d'objets. Le troisième niveau (N3) contient toutes les analyses (la détection, le suivi, la classification d'objets et la reconnaissance d'événements). Le symbole \vee montre la présence d'un niveau d'analyse vidéo. Si un niveau d'analyse n'est pas disponible, nous utilisons le symbole \emptyset tandis que si ce niveau n'est pas concerné, nous employons le symbole $-$. L'effort demandé est divisé en trois niveaux : peu (la requête est facilement exprimée), beaucoup (il est faisable mais difficile d'exprimer la requête) et indéfini (c'est impossible à exprimer la requête). Le symbole \times indique le niveau de l'effort demandé.	140
6.8	Résultats obtenus de la détection des blobs représentatifs, les méthodes 1 et 2 sont la méthode basée sur le changement d'apparence et celle basée sur le regroupement des blobs respectivement.	146
6.9	Valeurs moyennes des mesures F et P obtenues par les deux méthodes dans les deux expérimentations.	153
6.10	Moyennes des rangs moyens obtenus par les deux méthodes de deux expérimentations. Les rangs moyens obtenus par notre méthode sont plus petits que ceux de la méthode de Ma et al.	158
6.11	Comparaison de deux méthodes (notre méthode et celle de Ma et al.) basée sur les mesures TP (true positive) et FP (false positive). La valeur de TP mesure le nombre de résultats pertinents alors que la valeur de FP montre le nombre de résultats non pertinents dans les m premiers résultats. m a quatre valeurs (de 1 à 4). Le nombre de requêtes est 247 et 54 respectivement. Nous calculons les valeurs de TP et FP sur toutes les requêtes. N_m, G_m, TP_m, FP_m sont le nombre total des premiers résultats, le nombre total des résultats pertinents dans la vérité terrain, le nombre total des résultats pertinents et celui des résultats non pertinents dans les m premiers résultats de toutes les requêtes.	160
6.12	Moyennes des rangs moyens obtenus par les deux méthodes (notre méthode et celle de Calderara et al.) dans les trois expérimentations. Plus les valeurs de rang normalisé moyen sont petites, plus la méthode est efficace.	167

6.13	Comparaison de deux méthodes (notre méthode et celle de Calderara et al.) basée sur les mesures TP (true positive) et FP (false positive). La valeur de TP mesure le nombre de résultats pertinents alors que la valeur de FP montre le nombre de résultats non pertinents dans les m premiers résultats. m a quatre valeurs (de 1 à 4). Le nombre de requêtes est 16, 247, et 54 respectivement. Nous calculons les valeurs de TP et FP sur toutes les requêtes. N_m, G_m, TP_m, FP_m sont le nombre total des premiers résultats, le nombre total des résultats pertinents dans la vérité terrain, le nombre total des résultats pertinents et celui des résultats non pertinents dans les m premiers résultats de toutes les requêtes.	169
6.14	Comparaison de la performance des 4 descripteurs d'apparence pour les premiers résultats sur les vidéos provenant du projet CAVIAR : les couleurs dominantes (DC), les histogrammes de contours (HC), les matrices de covariance (MC) et les points intérêt DoG+SIFT basée sur les mesures TP (true positive) et FP (false positive). La valeur de TP mesure le nombre de résultats pertinents alors que la valeur de FP montre le nombre de résultats non pertinents dans les m premiers résultats. m a quatre valeurs (de 1 à 4). Le nombre de requête est 15. Nous calculons les valeurs de TP et FP sur toutes les 15 requêtes. N_m, G_m, TP_m, FP_m sont le nombre total des premiers résultats, le nombre total des résultats pertinents dans la vérité terrain, le nombre total des résultats pertinents et celui des résultats non pertinents dans les m premiers résultats de toutes les 15 requêtes.	178
6.15	Rangs obtenus par les descripteurs pour les personnes observées à différents moments ou par différentes caméras. Les matrices de covariance obtiennent les meilleurs résultats.	178
A.1	Fonctions d'accès et opérateurs d'apparence et ceux d'appartenance définis dans le langage de requêtes	206
B.1	Environnement de travail	207

Table des figures

1.1	Architecture générale de l'indexation et de la recherche de vidéos de vidéosurveillance. Les vidéos acquises sont tout d'abord prétraitées par un module d'analyse vidéo. Les sorties de ce module sont ensuite les entrées de l'indexation et de la recherche.	4
2.1	Transformation des points d'intérêt détectés dans une image à une phase de mots : après avoir enlevé des points redondants, un axe est déterminé pour tous les points. Les points sont projetés sur l'axe déterminé ([Tirilly 2008]).	16
2.2	Indexation et recherche d'images avec une ontologie ([Mezaris 2004]).	17
2.3	Correspondance entre des descripteurs à bas niveau et des termes à intermédiaire niveau pour la luminance ([Mezaris 2004]).	18
2.4	Graphe conceptuel construit pour l'image de piscine. Trois noeuds qui sont des concepts : "image", "foliage", "water pool". Les autres noeuds sont des relations ([Lim 2003b]).	19
2.5	Graphe conceptuel déterminé pour le modèle d'événement "piscine". Deux noeuds qui sont des concepts : "water pool" et "object". Le concept "object" a deux sous-concepts : "nature" et "man made". Deux noeuds sont des relations : "on top" et "touches" ([Lim 2003b]).	20
2.6	Illustration de retour de pertinence basé (a) sur la modification de requête (b) et sur le rôle de chaque descripteur dans le cas où deux descripteurs f_1 , f_2 sont utilisés ([Yin 2005]).	23
2.7	Représentation et analyse de la structure des vidéos scénarisées ([Xiong 2006]).	27
2.8	Indexation et recherche de journaux télévisés en se basant sur l'interaction avec l'utilisateur ([Zhai 2006]).	30
2.9	Reconnaissance de personnes dans les journaux télévisés basé sur leurs vêtements quand la reconnaissance de visages est disponible ([Jaffré 2004]).	31
2.10	Représentation de vidéos non scénarisées à différents degrés de granularité (<i>break and play</i> , <i>marqueur visuels et auditifs</i> , <i>highlight candidate</i> et <i>highlight group</i>) et analyse des vidéos non scénarisées ([Xiong 2006]).	33
2.11	Détection de <i>break</i> et <i>play</i> dans une vidéo ([Ekin 2003b]).	35
2.12	Détection de la présence d'un arbitre : (a) l'arbitre dans un frame ; (b) la projection horizontale des couleurs dominantes des pixels ; (c) la projection verticale des couleurs dominantes des pixels ; (d) la région déterminée pour l'arbitre ([Ekin 2003a]).	36

2.13	Détection de la surface de réparation : (a) le frame étudié; (b) le masque du champ; (c) les champs avec herbes ou sans herbe; (d) le frame après avoir appliqué le Laplacien; (e) le frame après le traitement; (f) les trois lignes parallèles détectées ([Ekin 2003a]).	36
2.14	Machine à états finis correspond à un but gagné est construite en se basant sur la connaissance du domaine ([Assfalg 2003]).	37
2.15	Architecture générale du module d'analyse vidéo pour la vidéosurveillance proposée par le projet PULSAR. Ce module est constitué de 4 tâches : la détection d'objets, la classification d'objets, le suivi d'objets et la reconnaissance d'événements. Les connaissances a priori comprenant l'information de contexte et la connaissances du domaine peuvent être utilisées ([Avanzi 2005]).	39
2.16	Indexation et la recherche de vidéos pour la vidéosurveillance au niveau objets avec la fusion précoce. Les données provenant de différentes caméras sont fusionnées dans la détection et le suivi d'objets. L'indexation et la recherche s'effectuent sur les données fusionnées.	40
2.17	Indexation et la recherche de vidéos pour la vidéosurveillance au niveau objets avec la fusion tardive. L'indexation et la recherche s'effectuent sur la donnée de chaque caméra. Les résultats de recherche sont fusionnés.	42
2.18	Processus de génération PA, SCAT et MCAT pour chacun des objets observés par un ensemble de caméras ([Calderara 2006]).	45
2.19	Appariement des objets est constitué de deux étapes : la première étape (Best PA selection) permettant de déterminer le PA d'un objet à travers de PAs créés à partir des vidéos provenant de toutes les caméras, la deuxième étape consistant de comparer le PA choisi avec les MCATs de tous les objets détectés ([Calderara 2006]).	46
2.20	Objets sont détectés et suivis par un module vision (<i>Intermediate Vision Modules</i>). Les événements primitifs reconnus par <i>Primitives Detection</i> sont représentés par des réseaux de Petri. Les requêtes qui définissent un événement complexe à partir des événements primitifs, sont également représentées par un réseau de Petri ([Ghanem 2004]).	47
2.21	Approche proposée par Hu et al. : après avoir suivi d'objets, les trajectoires des objets et les modèles d'activités appris sont stockées dans la base de données ([Hu 2007]).	49
2.22	Approche proposée par Stringa et al. Une fois l'abandon d'un objet reconnu (une alarme est déclenchée), un plan de vidéo constitué de 24 frames est créé. L'approche se limite à retrouver des plans vidéo concernant l'événement d'abandon d'un objet ([Stringa 2000]).	50
2.23	Construction d'un plan vidéo concernant l'événement d'abandon d'un objet. Le plan est constitué de 24 frames du X-8 à X+16 où X est le moment auquel l'alarme est déclenchée ([Stringa 1998]).	50

2.24	Exemple de la détection du moment de fin t_{fin} de l'événement d'abandon d'un objet : (a) le graphe temporel correspondant aux objets détectés et suivis (Object Tracking Layer - OTL) et à l'objet abandonnée (Abandoned Object Layer-AOL). Le frame de fin correspond au moment où un noeud est divisé en deux dans le graphe temporel ; (b) les frames correspondant à ce graphe temporel ([Foresti 2002]).	52
2.25	Exemple de la détection du moment de début t_{in} de l'événement d'abandon d'un objet. Le moment de début t_{in} est déterminé par le chemin simple connecté au noeud correspondant à l'objet abandonné dans le graphe ([Foresti 2002]).	53
2.26	Approche de Greenhill et al. consiste à (1) détecter et suivre les objets, (2) extraire les descripteurs et faire les annotations (3) et répondre aux requêtes des utilisateurs ([Greenhill 2002]).	53
2.27	Architecture de l'approche interactive pour la détection des accidents dans les enregistrements de routes. Les véhicules sont détectés et suivis. Les événements d'intérêt sont modélisés. Un réseau de neurones est créé et entraîné en utilisant les retours des utilisateurs ([Chen 2006]).	56
3.1	Tâches principales de la phase d'indexation et de celle de recherche de l'approche proposée. Les parties en bleue sont nos contributions dans cette thèse.	64
4.1	Attributs d'un objet dont ID = 57. Cet objet appartient à la classe Person. Il est détecté et suivi pendant 143 frames (du frame #2017 au frame #2160). Cinq blobs représentatifs associés au poids sont déterminés pour cet objet. Pour chacun des blobs, un vecteur de 5 éléments de l'histogramme de contours est calculé pour l'attribut R_{ap} .	74
4.2	Attributs d'un objet de contexte (Gates) dont ID = 1.	74
4.3	Une image d'exemple, l'attribut R_{ap} est un vecteur de 5 éléments de l'histogramme de contours.	75
4.4	Un événement "inside_zone_Platforme" dont ID=50 est représenté dans le modèle de données. L'objet impliqué dans cette événement est montré dans la figure 4.1.	76
4.5	(a) blob d'une personne détectée ; (b) et projection de 3 couleurs dominantes sur le blob.	79
4.6	Cinq types de contours : vertical, horizontal, 45 degré, 135 degré et non directionnel [Park 2000].	80
4.7	Décomposition d'une image en 16 sous-images pour calculer l'histogramme local et l'identification de 13 sous-histogrammes semi-locaux [Won 2002].	80
4.8	(a) blob d'une personne détectée ; (b) points d'intérêt de Harris Affine ; (c) points d'intérêt de MSER.	85
4.9	Identification des orientations pour chacun des points d'intérêts se base sur les maximums de l'histogramme de l'orientation ([Lowe 2004]).	87

4.10	Région divisée en 16×16 sous-régions. Pour chaque bloc de 4×4 sous-régions, un histogramme de 8 éléments, correspondant à 8 orientations est créé ([Lowe 2004]).	87
4.11	Détection des points de contrôle dans une trajectoire T : (a) un point de contrôle (p), deux points p^- et p^+ reliant à p sont satisfaits l'équation 4.18 ; (b) p_1 et p_2 are deux points détectés qui sont proches l'un de l'autre. Le point ayant l'angle le plus petit est gardé comme le point de contrôle ([Hsieh 2006]).	90
4.12	Exemple de l'espace défini par la direction de mouvement et l'incrément relatif de distance parcourue entre deux positions consécutives. Cet espace est divisé en 64 sous-régions, chacune des sous-régions est assignée à un symbole distinctif ([Chen 2004]).	92
4.13	Exemple d'une personne détectée pendant 24 frames. 24 blobs dont 12 blobs pertinents (en bleu) et 12 blobs non pertinents sont déterminés.	94
4.14	26 blobs d'une personne détectée et les points d'intérêt de MSER appariés pour 25 frames consécutifs.	96
4.15	Blobs représentatifs associés à leurs poids détectés par la méthode basée sur le changement d'apparence.	97
5.1	(a) blobs représentatifs de l'objet #1064 ; (b) et ceux de l'objet #1065. L'objet 1064 n'est pas bien détecté et suivi pendant certains frames. Le blob représentatif 1 n'est pas pertinent.	109
5.2	Mise en correspondance entre des objets en utilisant la distance EMD (en bleu) et la distance de Hausdorff (en rouge). La distance EMD tient compte de la contribution de chaque blob en fonction de son poids. Le chiffre associé à chaque paire de blobs montre la participation de ces blobs dans la mise en correspondance basée sur la distance EMD (en bleu) Plus le poids du blob est élevé, plus la contribution du blob est impacte. Le poids du blob 3 de l'objet 1064 est le plus élevé (0.842), ce blob a un rôle important dans la mise en correspondance de cette objet alors que le rôle du blob 1 (non pertinent) n'est pas considérable.	110
5.3	Interface du système d'indexation et de recherche d'images proposé par le projet IMEDIA, INRIA. Trois types de requêtes (en vert) sont possibles : une image fournie par l'utilisateur, une image choisie par l'utilisateur en naviguant dans la base de données, quelques mots clés (si l'annotation est disponible).	114
5.4	Interface du système d'indexation et de recherche de vidéos de vidéo-surveillance. Un ensemble limité de requêtes est à gauche. La requête "Find Cars" (en vert) est activé, des petites vidéos à droite sont des résultats de cette requête ([Tian 2008]).	115
5.5	Réseau de Pétri pour la requête "compter le nombre de voitures qui sont garées dans un zone pendant une période" ([Ghanem 2004]). . .	118

5.6	Réseau de Pétri pour la requête “trouver une personne qui passe d’une voiture à l’autre” ([Ghanem 2004]).	119
6.1	Trajectoires d’exemple dans la base des trajectoires.	129
6.2	Quelques frames dans les vidéos provenant du projet CARETAKER.	130
6.3	Supermarché dans le projet CAVIAR est observé par deux caméras : l’une est dans le couloir et l’autre est en face du magasin.	135
6.4	Une personne détectée et suivie pendant 905 frames de la vidéo CARE_5, 4 blobs représentatifs sont identifiés pour cette personne. Les images en haut sont les images de la scène avec la présence de la personne. Les images en bas sont les blobs représentatifs de la personne. Tous les quatre blobs représentatifs sont pertinents. Ils représentent des aspects différents de la personne.	146
6.5	Une personne détectée et suivie pendant 95 frames de la vidéo CARE_5, 3 blobs représentatifs sont identifiés pour cette personne. Les images en haut sont les images de la scène avec la présence de la personne. Les images en bas sont les blobs représentatifs de la personne. Tous les trois blobs représentatifs sont pertinents. Les blobs représentent des aspects différents (avec ou sans présence d’autres personnes).	147
6.6	Une personne détectée et suivie pendant 175 frames de la vidéo CARE_6, 4 blobs représentatifs sont identifiés pour cette personne. Les images en haut sont les images de la scène avec la présence de la personne alors que les images en bas sont les blobs représentatifs de la personne. Tous les quatre blobs représentatifs sont pertinents. Ils représentent les aspects différents de la personne. Cependant, la détection de la personne n’est pas bonne, la personne est entièrement présente dans un seul blob parmi 4 blobs représentatifs.	147
6.7	Valeurs de la mesure F obtenues par les deux méthodes dans la première expérimentation. Cette mesure exprime la capacité de réduire les informations à stocker. Une méthode est efficace si elle obtient une petite valeur de F. La valeur de F peut être de 0% à 100% (respectivement de 0 à 1 dans cette figure). Parmi 675 objets, la valeur de F obtenue par notre méthode est plus petite que celle obtenue par la méthode de Ma et al. sur 96 objets alors que la méthode de Ma et al. a de meilleurs résultats sur 47 objets.	149
6.8	Valeurs de la mesure P obtenues par les deux méthodes dans la première expérimentation. La mesure P montre la capacité à corriger les erreurs produites par la détection et le suivi d’objets. Une méthode est efficace si elle obtient une grande valeur de P. La valeur de P peut être de 0% à 100% (respectivement de 0 à 1 dans cette figure). Parmi 675 objets, la valeur de P obtenue par notre méthode est plus élevée que celle obtenue par la méthode de Ma et al. sur 157 objets alors que la méthode de Ma et al. a de meilleurs résultats sur 49 objets.	150

6.9	Valeurs de la mesure F obtenues par les deux méthodes dans la deuxième expérimentation. Cette mesure exprime la capacité de réduire les informations à stocker. Une méthode est efficace si elle obtient une petite valeur de F. La valeur de F peut être de 0% à 100% (respectivement de 0 à 1 dans cette figure). Parmi 661 objets, la valeur de F obtenue par notre méthode est plus petite que celle obtenue par la méthode de Ma et al. sur 131 objets alors que la méthode de Ma et al. a de meilleurs résultats sur 41 objets.	151
6.10	Valeurs de la mesure P obtenues par les deux méthodes dans la deuxième expérimentation. Cette mesure montre la capacité à corriger les erreurs produites par la détection et le suivi d'objets. Une méthode est efficace si elle obtient une grande valeur de P. La valeur de P peut être de 0% à 100% (respectivement de 0 à 1 dans cette figure). Parmi 661 objets, notre méthode obtient la valeur de P plus élevée que celle de la méthode de Ma et al. sur 142 objets alors que la méthode de Ma et al. a de meilleurs résultats sur 75 objets.	152
6.11	Deux personnes avec leurs blobs représentatifs détectés par la méthode de Ma et al. . Les blobs en rouge qui ne sont pas appropriés sont enlevés par notre méthode.	154
6.12	(a) personne détectée n'est pas présente dans le blob ; (b) personne détectée est partiellement présente dans le blob ; (c) et (d) deux personnes sont présentes dans un seul blob.	156
6.13	(a) un exemple de confusion d'étiquette : trois personnes réelles sont détectées et suivies comme une seule personne ; (b) un exemple de persistance d'étiquette : une personne est détectée et suivie comme deux personnes différentes.	156
6.14	Rangs normalisés moyens obtenus par les deux méthodes sur 247 personnes recherchées. Plus les valeurs de rang normalisé moyen sont petites, plus la méthode est efficace. Notre méthode est plus performante que celle de Ma et al. sur 187 requêtes sur les 247.	157
6.15	Rangs normalisés moyens obtenus par les deux méthodes sur 54 requêtes. Plus les valeurs de rang normalisé moyen sont petites, plus la méthode est efficace. Notre méthode obtient de meilleurs résultats sur 32 requêtes.	158
6.16	4 blobs représentatifs et leurs poids pour la personne #2160	160
6.17	4 blobs représentatifs et leurs poids pour la personne #1518	161
6.18	5 blobs représentatifs et leurs poids pour la personne #1763	161
6.19	Mise en correspondance : (a) entre les deux personnes #2160 et #1518 ; (b) et entre les deux personnes #2160 et # 1763. Les valeurs correspondantes montrent des parties que ce blob participe à la mise en correspondance dans notre méthode (en bleu). La méthode de Ma et al. détermine la distance entre deux ensembles de blobs par la distance de deux blobs (en rouge)	162

6.20	Rangs normalisés moyens obtenus par notre approche et celle de Calderara et al. avec 16 requêtes sur 810 personnes indexées de la vidéo CARE_6.	164
6.21	Rangs normalisés moyens obtenus par notre approche et celle de Calderara et al. dans la deuxième expérimentation avec 247 personnes recherchées sur 810 personnes indexées.	165
6.22	Rangs normalisés moyens obtenus par la troisième expérimentation avec 54 personnes recherchées de CARE_5 sur 810 personnes indexées de CARE_6	166
6.23	Trois blobs représentatifs pour une personne recherchée. Les blobs représentatifs ne sont pas toujours pertinents.	166
6.24	Résultats de la requête #6 de la troisième expérimentation. Les images en haut sont les blobs représentatifs de la personne recherchée et les trois premiers résultats. Les images en bas sont les blobs de la personne recherchée et les premiers résultats obtenus par la méthode de Calderara et al. Les résultats en rouge sont des résultats pertinents.	167
6.25	Résultats de la requête #40 de la troisième expérimentation. Les images en haut sont les blobs représentatifs de la personne recherchée et les trois premiers résultats. Les images en bas sont les blobs de la personne recherchée et les premiers résultats obtenus par la méthode de Calderara et al. Les résultats en rouge sont des résultats pertinents.	168
6.26	Rangs normalisés obtenus avec 4 types d'histogrammes des contours sur 2311 objets de la vidéo CARE_4 : l'histogramme local, semi-local, global et composé	171
6.27	Requête #1 et deux objets retrouvés en utilisant l'histogramme global qui ne sont pas pertinents	171
6.28	Requête #21 avec un objet qui n'est pas pertinent mais retrouvé (#2727) en utilisant l'histogramme local tandis que l'objet pertinent (#219) n'est pas retrouvé.	172
6.29	Rangs normalisés obtenus avec les couleurs dominantes et les matrices de covariance	173
6.30	Les couleurs dominantes déterminées pour un blob avec les valeurs 10, 15, 25 de T_d respectivement.	173
6.31	Rangs normalisés obtenus de 7 requêtes sur 53 objets provenant du projet CAVIAR en utilisant les points d'intérêt (DoG, MSER, HarrisAffine) associés au descripteur SIFT	175
6.32	Résultats de recherche avec 23 requêtes sur 2311 objets de la vidéo CARE_4 avec les points d'intérêt de MSER et SIFT, les histogrammes des contours, les couleurs dominantes et les matrices de covariance.	176
6.33	Résultats de recherche avec 15 requêtes sur 53 objets provenant du projet CAVIAR en utilisant les descripteurs : les points d'intérêt de DoG et SIFT, les histogrammes des contours, les couleurs dominantes et les matrices de covariance	177

6.34	Courbes de rappel et précision obtenus pour EDR et EDM en utilisant tous les points de trajectoires et les points de contrôle	180
6.35	Image d'exemple est à gauche, trois objets retrouvés dont les étiquettes sont 176, 162, 111. Ces trois objets impliqués dans l'événement "close_to_VendingMachine1" aux frames 5940, 5895, et 3825 respectivement.	182
6.36	Rangs normalisés moyens obtenus de 15 requêtes sur 19 événements de <i>close_to_Gate1</i> pour la vidéo CARE_2 du projet CATETAKER.	183
6.37	Rangs normalisés moyens obtenus pour 19 événements de <i>close_to_Gate1</i> et 29 événements de <i>inside_zone_Platform</i>	184
6.38	Rangs normalisés moyens pour la première méthode de retour de pertinence sur 145 objets de CARE_1 avec 15 requêtes. Les valeurs de M et le nombre d'itérations maximal autorisé sont 16 et 5 respectivement.	187
6.39	(a) requête #14 est initialisée par une image d'exemple ; (b) images positives dans la requête #14 lors de la première itération	188
6.40	Blobs représentatifs de l'objet #1.	188
6.41	Rangs normalisés moyens pour la première méthode de retour de pertinence sur 810 objets de CARE_6 avec 50 requêtes provenant de la vidéo CARE_5. M et le nombre d'itérations maximal autorisé sont 100 et 5 respectivement.	189
6.42	(a) résultats obtenus avec la requête #20 sans retour de pertinence ; (b) lors de la première itération ; (c) et de la deuxième itération ; (d) les requêtes en bleu, les objets non pertinents en rouge.	190
6.43	Rangs normalisés moyens pour le retour de pertinence basé sur les SVM à une classe sur 810 objets de CARE_6 avec 50 requêtes provenant de la vidéo CARE_5. M et le nombre d'itérations sont 100 et 5 respectivement.	191
B.1	Relation entre les 5 modules de notre prototype et les logiciels utilisés. Les 5 modules sont implémentés dans 3 librairies (libdescriptors, libutilities, libqlanguage).	208

Introduction

1.1 Motivations et objectifs

Trois tâches sont essentielles pour la gestion de documents multimédias [Rowe 2005] : les numériser, les stocker et les retrouver. Les avancées technologiques ont permis aux professionnels et aux particuliers de numériser et stocker de nombreux documents qui ne se limitent plus au texte mais qui incluent à présent la photo et la vidéo. Le moindre accessoire (appareils de photos, caméras, téléphones portables, etc) est maintenant capable d'acquérir de petites séquences d'images de notre quotidien. Et le tout peut facilement être partagé au travers du réseau Internet. Cependant retrouver un document voulu dans une multitude de documents multimédias n'est pas toujours efficace et parfois impossible.

Pour retrouver efficacement un document, deux phases sont importantes : la phase d'indexation et la phase de recherche. La phase d'indexation vise à calculer des descripteurs pertinents des documents multimédias et à créer des index à partir des descripteurs extraits. Une fois l'indexation effectuée, la phase de recherche consiste à mettre en correspondance la requête et les informations indexées et à retourner les résultats retrouvés à l'utilisateur. En général, le terme de "recherche d'information" contient lui-même implicitement les deux phases. La définition générale de la recherche d'information se trouve dans [Lew 2006] :

"Multimedia information retrieval (MIR) is about the search for knowledge in all its forms, everywhere."

Une approche de recherche d'information est efficace si elle est capable de retrouver des informations appropriées à la requête de l'utilisateur dans un temps acceptable.

La recherche d'information a naturellement commencé avec le texte, puis continué avec l'image et la vidéo. Les recherches de textes et d'images deviennent matures avec certains résultats obtenus [Smeulders 2000], [Sebe 2003], [Datta 2008]. La recherche de vidéos cependant devient active depuis seulement 10 ans. Les approches proposées au tout début étaient simplement de considérer une vidéo comme une séquence d'images. C'est pourquoi les approches dédiées à la recherche d'images ont été appliquées à celle de vidéos. Ces approches montrent évidemment des inconvénients parce qu'elles ne prennent pas en compte deux caractéristiques spécifiques des vidéos qui sont les suivantes :

- La vidéo elle-même, contient des informations riches telles que l'information visuelle, auditive et textuelle ;
- L'information noyée dans une vidéo peut être spatiale ou temporelle.

La considération d'une vidéo comme une séquence d'images ne prend que l'information visuelle et spatiale. La recherche de vidéo peut être classifiée par le type de vidéos : les vidéos professionnelles télédiffusées (films, émissions), les vidéos de sport, les vidéos de vidéosurveillance et les vidéos domestiques. La recherche de vidéo a commencé avec les vidéos professionnelles et continué avec les vidéos de sport et les vidéos de vidéosurveillance. L'augmentation du nombre de chaînes de télévision donne de grandes bases d'émissions stockées. La recherche de vidéos télédiffusées a été développée pour pouvoir répondre au besoin de retrouver des informations d'intérêt dans les émissions. L'analyse de vidéos de sport associant deux applications, le résumé de vidéos et l'indexation et la recherche de vidéos, a reçu beaucoup d'attention. Les recherches de vidéos domestiques et de vidéos de vidéosurveillance ont émergé ces dernières années.

1.1.1 Contexte

Les caméras se trouvent de plus en plus dans des environnements où l'on s'intéresse à savoir ce qui va se passer. Elles sont installées dans les lieux publics (routes, stations de métro, hôpitaux, supermarchés), ainsi que dans les lieux avec accès restreint (campus universitaires, bâtiments des entreprises, banques) ou dans les lieux personnels (à la maison). L'objectif de la vidéosurveillance est à la fois de détecter des événements prédéfinis (p. ex. une voiture roule en sens interdit) ou d'analyser à long terme des comportements humains (p. ex. l'analyse de la visite des clients dans les rayons dans un supermarché) dans les vidéos enregistrées. De nombreuses caméras et une capacité énorme de stockage permettent d'avoir de grandes bases de vidéos de vidéosurveillance. La valeur de ces bases de données dépend de la capacité de retrouver des informations voulues dans ces bases. De nombreuses approches proposées cherchent à répondre au besoin d'avoir des outils de recherche. Ce travail de thèse se place dans ce contexte.

De plus, le travail de cette thèse se trouve dans un contexte particulier. Car cette thèse se déroule au sein de l'équipe PULSAR¹ (Perception Understanding System for Activity Recognition) (anciennement ORION²). L'équipe PULSAR fait partie de l'Institut National de Recherche en Informatique et Automatique (INRIA), France. L'un des thèmes de cette équipe est l'interprétation automatique d'images et de vidéos. Depuis sa création, cette équipe a participé à plusieurs projets de vidéosurveillance tels que AVITRACK (Aircraft surroundings, categorised Vehicles & Individuals Tracking for apRon's Activity model interpretation & ChecK)³, CARETAKER (Content Analysis and RETrieval Technologies to Apply Extraction to massive Recording)⁴. Elle est partenaire de plusieurs projets en cours tels que GERHOME (GERrontology at HOME)⁵, CoFriend⁶. Grâce à ces projets, de grandes

¹<http://www-sop.inria.fr/pulsar/>

²<http://www-sop.inria.fr/orion/>

³<http://www.avitrack.net>

⁴<http://www.ist-caretaker.org/>

⁵<http://gerhome.cstb.fr>

⁶<http://co-friend.net/>

bases de vidéos ont été enregistrées et stockées. Les vidéos ont été également analysées par les algorithmes de vision proposés par l'équipe. La recherche d'information dans ces grandes bases, qui devient de plus en plus nécessaire, ne peut pas toujours être effectuée directement à partir des analyses.

1.1.2 Applications

L'indexation et la recherche de vidéos de vidéosurveillance trouve sa place dans des applications différentes avec plusieurs types d'utilisateurs. Nous décrivons ici deux grands types d'applications.

La première application est la sécurité. Dans un système de vidéosurveillance, une alarme est déclenchée si le système détecte un événement intéressant. Habituellement, le personnel de sécurité veut trouver des informations antérieures concernant le ou les objets impliqués dans cet événement. Par exemple, dans un parking, le système détecte une personne qui s'approche d'une voiture. Le personnel de sécurité peut s'intéresser à savoir ce qu'elle a fait avant de s'approcher de la voiture.

La deuxième application est l'étude statistique. Il est intéressant de savoir combien de fois par mois un événement aura lieu ou quel événement suit un événement particulier. Par exemple, dans les supermarchés, combien de fois le client visite le rayon A, et passe ensuite dans le rayon B.

1.1.3 Problèmes et Objectifs

Dans cette section, les caractéristiques que nous souhaitons prendre en compte dans notre travail de thèse sont tout d'abord présentées. Les spécifications d'une nouvelle approche et nos objectifs sont ensuite introduits. La figure 1.1 montre l'architecture générale de l'indexation et de la recherche de vidéos de vidéosurveillance. Les vidéos acquises sont tout d'abord prétraitées par un module d'analyse vidéo. Les sorties de ce module sont ensuite les entrées de l'indexation et de la recherche. Nous séparons bien dans ce travail de thèse le module d'analyse vidéo et l'indexation et la recherche pour pouvoir identifier leurs propres caractéristiques. Notons que dans les travaux de l'état de l'art, le module d'analyse vidéo est considéré comme un composant interne des systèmes d'indexation et de recherche.

Nous distinguons trois types de caractéristiques : les caractéristiques de la nature des vidéos, celles du module d'analyse vidéo, et celles de l'indexation et de la recherche.

Les caractéristiques de la nature des vidéos de vidéosurveillance sont :

- La qualité des images qui peut être faible ou élevée en fonction de la résolution de l'image ;
- Le changement d'illumination, surtout dans les applications en milieu extérieur, est considérable ;
- Le contexte est plus ou moins complexe (p. ex. il existe plusieurs types d'objets) ;

- L'activité humaine est très variée, il existe par exemple des occultations entre les personnes et les objets du contexte, et des interactions entre les personnes.

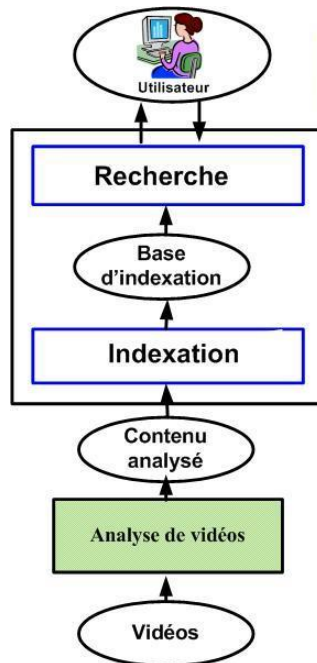


FIG. 1.1 – Architecture générale de l'indexation et de la recherche de vidéos de vidéosurveillance. Les vidéos acquises sont tout d'abord prétraitées par un module d'analyse vidéo. Les sorties de ce module sont ensuite les entrées de l'indexation et de la recherche.

Les caractéristiques du module d'analyse comprennent :

- En vidéosurveillance, une scène peut être observée par une ou plusieurs caméras. Dans le cas où les vidéos d'une scène sont enregistrées par plusieurs caméras, l'analyse peut être effectuée avant ou après la fusion des données ;
- Il existe deux types de connaissances a priori : celle du contexte et celle du domaine. La connaissance du contexte décrit tout ce qui est présent dans la scène vidéo. La connaissance du domaine définit des événements d'intérêt pour le domaine considéré ;
- L'analyse de vidéos est effectuée à différents degrés de granularité. Elle peut s'arrêter à la détection, au suivi, à la classification d'objets ou à la reconnaissance d'événements. Elle contient en général la détection des objets, le suivi des objets, la classification des objets et parfois la reconnaissance des événements ;
- Les résultats des modules d'analyse ne sont pas toujours parfaits. Cela a été montré dans les évaluations des modules d'analyse vidéo sur les bases

communes de vidéos telles que CAVIAR ⁷ (Context Aware Vision using Image-based Active Recognition) [Nascimento 2006], ETISEO ⁸ [Nghiem 2007].

Les caractéristiques de l’indexation et de la recherche de vidéos de vidéosurveillance sont :

- Dans le cas où une scène est observée par plusieurs caméras, l’indexation et la recherche de vidéos de vidéosurveillance peuvent travailler sur des données qui sont déjà fusionnées ou sur des données qui sont séparément traitées dans les modules d’analyse vidéo ;
- L’indexation et la recherche de vidéos de vidéosurveillance peuvent être effectuées à différents niveaux : images, objets et événements ;
- L’objectif de la recherche de vidéos de vidéosurveillance est de retrouver les frames contenant des objets d’intérêt, des événements intéressants selon des critères. Ces critères peuvent concerner des attributs des objets, et/ou certains des événements. La recherche de vidéos de vidéosurveillance se fait donc au niveau plus fin que celle des journaux télévisés. Retrouver des plans entiers de vidéo n’est pas l’objectif de la recherche de vidéos de vidéosurveillance.

Un des grands défis dans la recherche d’informations est le fossé sémantique entre la similarité calculée sur les informations indexées et la similarité attendue par l’utilisateur [Smeulders 2000]. Bien que le fossé sémantique soit causé par plusieurs facteurs, nous présentons les deux facteurs principaux. L’un des facteurs principaux est la distance entre le contenu de vidéos qui est riche en sémantique, et la méthode d’extraction automatique, qui essaie de représenter ce contenu par un ensemble de descripteurs de bas niveaux. L’autre facteur est le manque de conformité entre d’une part les informations indexées sous la forme de vecteurs de descripteurs à bas niveaux et d’autre part les requêtes exprimées par les utilisateurs.

Les performances des approches d’indexation et de recherche sont mesurées par la capacité de combler le fossé sémantique. Pour l’indexation et la recherche de vidéos de vidéosurveillance, une approche qui permet de combler le fossé sémantique doit :

- profiter des résultats des modules d’analyse vidéo car de nombreuses approches avec certains de résultats ont été proposées pour l’analyse de vidéos. Pour pouvoir utiliser des approches différentes, l’indexation et la recherche doivent être générales ;
- pourvoir d’une part corriger des erreurs produites par l’imperfection des modules d’analyse vidéo, d’autre part compléter l’indexation effectuée par les sorties des modules d’analyse vidéo ;
- être interactif : l’interaction avec l’utilisateur nous permet d’une part de comprendre mieux les requêtes des utilisateurs et d’autre part d’avoir les jugements des utilisateurs sur les résultats de recherche.

Notre travail de thèse a pour objectif de concevoir une telle approche.

⁷<http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/>

⁸<http://www-sop.inria.fr/orion/ETISEO/index.htm>

1.2 Approche proposée

Nous proposons dans cette thèse une approche pour l'indexation et la recherche de vidéos de vidéosurveillance. Nous présentons tout d'abord nos hypothèses dans la section 1.2.1. Ensuite, nous analysons dans la section 1.2.2 les questions posées en indexation et recherche de vidéos de vidéosurveillance. Enfin, nous présentons les contributions que nous apportons dans notre thèse dans la section 1.2.3.

1.2.1 Hypothèses

Notre approche se base sur les trois hypothèses suivantes :

- Hypothèse 1 : nous supposons que les vidéos doivent être prétraitées par un module d'analyse vidéo. Ce module d'analyse vidéo doit effectuer obligatoirement la détection et le suivi d'objets. La classification d'objets et la reconnaissance d'événements sont facultatives ;
- Hypothèse 2 : nous ne travaillons que sur des vidéos non éditées. La détection de transitions, la décomposition d'une vidéo en plans, le regroupement des plans en scènes et en groupes ne sont donc pas considérés dans notre travail de recherche ;
- Hypothèse 3 : bien que notre approche puisse être prolongée pour travailler avec des données visuelles et auditives, nous supposons que notre travail n'utilise que des données visuelles.

1.2.2 Questions ouvertes

Avant de présenter nos contributions, nous analysons six questions indispensables posées en indexation et recherche de vidéos de vidéosurveillance, dont deux (questions 1, 2) dans la phase d'indexation et quatre (questions 3, 4, 5, 6) dans celle de recherche.

Question 1 : Quelles sont les informations noyées dans une vidéo qu'il faut calculer et stocker pendant la phase d'indexation, pour que la phase de recherche soit capable de répondre à toutes les requêtes de l'utilisateur ? Cette question se pose quand on fait l'indexation. Les requêtes de l'utilisateur peuvent être très variées et il est impossible de toutes les prévoir dès la phase d'indexation ; plus on calcule d'informations, plus le système arrivera à répondre aux requêtes. Il existe un compromis entre les informations analysées et le coût de stockage et de calcul. De plus, en sachant que l'analyse manuelle des grandes bases de données est une tâche fastidieuse et coûteuse en temps, l'analyse automatique devient un choix judicieux.

Question 2 : Quels descripteurs peut-on extraire ? Sont-ils suffisamment expressifs ? Aucun descripteur n'est suffisant pour toutes les applications et toutes les requêtes. Les approches d'indexation et de recherche doivent extraire les descripteurs appropriés. Elles doivent permettre de rajouter de nouveaux descripteurs en fonction de l'application.

Question 3 : Que cherche l'utilisateur dans un enregistrement de vidéosurveillance ? Il peut chercher des informations qui sont déjà indexées, ou qui ne le sont pas encore. Par contre, elles peuvent être déduites à partir des informations indexées.

Question 4 : Comment l'utilisateur formule-t-il ses propres requêtes et que peut fournir l'utilisateur ?

Question 5 : Comment peut-on mettre en correspondance entre les informations indexées et la requête en se basant sur les descripteurs ?

Question 6 : Comment et à quel niveau l'utilisateur peut-il interagir avec le système ? Il existe deux types de retour de pertinence : l'un est à court terme et l'autre à long terme. Le retour de pertinence à court terme tente d'analyser les retours de l'utilisateur pour mieux répondre à sa question. Les interactions se font dans une seule session et le résultat n'est pas stocké pour être utilisé plus tard. Alors que celui à long terme le fait.

1.2.3 Contributions

Nous proposons une approche hybride pour l'indexation et la recherche de vidéos de vidéosurveillance qui combine vision par ordinateur, apprentissage automatique et interaction homme-machine afin de combler le fossé sémantique. Les techniques de vision par ordinateur et d'apprentissage automatique concernent la phase d'indexation alors que les techniques d'apprentissage et d'interaction homme-machine concernent la phase de recherche. La phase d'indexation tente de monter au niveau sémantique le plus haut possible à partir des descripteurs de bas niveaux tandis que la phase de recherche essaie de passer de la requête sémantique fournie par l'utilisateur aux niveaux intermédiaires ou aux niveaux plus bas afin de mettre en correspondance entre la requête et les informations indexées.

Cette approche permet de faire une interaction intelligente et flexible entre la phase d'indexation et celle de recherche dans deux directions. D'une part, elle permet de profiter et de réutiliser des résultats obtenus en vision par ordinateur et en apprentissage automatique pour avoir une indexation riche et sémantique. D'autre part, elle permet d'employer des techniques d'apprentissage automatique et d'interaction homme-machine soit pour compléter l'indexation, soit pour la corriger en prenant en compte la participation de l'utilisateur et les caractéristiques de la vidéosurveillance.

Nous apportons dans cette thèse cinq contributions :

La première contribution est un **modèle de données** pour l'indexation et la recherche de vidéos de vidéosurveillance. Le modèle de données proposé contient deux concepts abstraits : objets et événements. Le modèle de données est général. Il nous permet de travailler avec différents modules d'analyse vidéo qui sont plus ou moins sophistiqués.

La deuxième contribution est un nouveau **langage de requêtes**. En se basant sur le modèle de données, le langage de requêtes proposé permet de formuler les requêtes à trois niveaux : images, objets et événements. Des images d'exemple contenant des objets d'intérêt peuvent être associées aux requêtes formulées par ce

langage.

La troisième contribution concerne la représentation des objets. Un objet peut être détecté et suivi dans plusieurs frames. Pour chacun des frames, l'objet est en général déterminé par une région entourée par sa boîte englobante minimale. Désormais, nous utilisons le terme blob pour indiquer cette région. À cause de l'imperfection des modules d'analyse vidéo, l'utilisation de tous les blobs déterminés d'un objet est redondante et inefficace. Cela peut accumuler les erreurs d'un frame à d'autre frame. Nous proposons deux méthodes de **détection des blobs représentatifs** des objets qui nous permettent de choisir des blobs pertinents pour chacun des objets.

La quatrième contribution est une nouvelle méthode de **mise en correspondance** entre des objets. Un objet est représenté par un ensemble de blobs représentatifs. Plusieurs descripteurs peuvent être extraits à partir d'un blob. Pour une paire de blobs, plusieurs distances de similarité peuvent donc être calculées. Afin de mesurer la distance entre des objets à partir des distances entre leurs blobs, nous proposons une nouvelle méthode de mise en correspondance entre des objets basée sur la distance EMD (Earth Movers Distance).

La cinquième contribution concerne le retour de pertinence. **Deux méthodes de retour de pertinence à court terme** qui se basent sur les objets sont proposées.

1.3 Structure du manuscrit

Le manuscrit est divisé en sept chapitres. Nous décrivons le contenu de chacun des chapitres.

Chapitre 1 : Le but de ce chapitre est double : d'une part il annonce les problèmes posés dans le cadre de l'indexation et de la recherche de vidéos de vidéosurveillance que nous abordons dans notre travail de thèse, d'autre part, il présente nos contributions dans ce travail.

Chapitre 2 : Dans ce chapitre nous faisons un état de l'art des approches récentes d'indexation et de recherche d'images et de vidéos en général et de vidéos de vidéosurveillance en particulier. Cet état de l'art nous permet de placer notre approche dans l'ensemble des approches récemment proposées. Une comparaison des approches dédiées à l'indexation et la recherche de vidéos de vidéosurveillance est donnée. Cette comparaison montre les problèmes restants en indexation et recherche de vidéos de vidéosurveillance que nous énonçons dans le premier chapitre. Nous analysons les relations entre les travaux dédiés à l'indexation et à la recherche d'images et de vidéos structurées et notre travail.

Chapitre 3 : Le chapitre 3 décrit l'approche proposée. L'approche proposée est composée de deux phases qui font l'objet d'une description approfondie dans les chapitres 4 et 5 respectivement : la phase d'indexation et celle de recherche.

Chapitre 4 : Le chapitre 4 est dédié à la phase d'indexation de l'approche proposée. L'indexation se base sur un module d'analyse vidéo et un modèle de données. L'indexation consiste à représenter les objets et les événements en extrayant

des descripteurs.

Chapitre 5 : Ce chapitre détaille la phase de recherche de l'approche proposée. Cette phase est composée de trois tâches fondamentales : la formulation des requêtes basée sur un langage de requêtes, la mise en correspondance et le retour de pertinence.

Chapitre 6 : Ce chapitre est dédié à l'évaluation de l'approche proposée. Nous présentons les bases de données utilisées, les mesures d'évaluation, et les résultats obtenus.

Chapitre 7 : Dans ce chapitre, nous donnons les conclusions sur l'approche proposée. Nous discutons également des perspectives à court terme et à long terme.

État de l'art

Ce chapitre dresse un panorama des approches dédiées à l'indexation et à la recherche d'images et de vidéos. Nous présentons tout d'abord l'indexation et la recherche d'images et ensuite celles de vidéos. L'indexation et la recherche de vidéos sont divisées en deux grands types selon la caractéristique des vidéos traitées. Nous décrivons ces deux types de vidéos en donnant les définitions de terminologies utilisées et en présentant les approches dédiées à chaque type de vidéo. Ensuite, nous indiquons où notre travail de thèse se situe dans ce panorama. Une analyse détaillée des approches dédiées à l'indexation et à la recherche de vidéos pour la vidéosurveillance est donnée. Nous présentons les problèmes ouverts en indexation et recherche de vidéos pour la vidéosurveillance qui motivent notre travail de thèse.

2.1 Indexation et recherche d'images

L'indexation et la recherche d'images deviennent un domaine très actif depuis 1994. Selon [Datta 2008], le nombre de publications dans ce domaine est environ 1000 publications chaque année. Nous présentons dans cet état de l'art trois aspects importants : la recherche locale, l'ontologie et le retour de pertinence. Nous invitons des lecteurs à lire une analyse de plus 300 articles [Datta 2008] et un rapport de plus 43 systèmes proposés [Veltkamp 2000] dans ce domaine.

2.1.1 Recherche locale vs recherche globale

Les premières approches proposées pour l'indexation et la recherche d'images [Wayne 1993], [Pentland 1994] ont utilisé les descripteurs globaux d'images. Les descripteurs globaux tels que l'histogramme des couleurs sont calculées avec la participation de tous les pixels dans une image. La disposition des pixels n'est donc pas prise en compte. Évidemment, de telles approches ne sont pas appropriées pour la recherche d'images ayant une seule ou quelques parties similaires à la requête. La recherche d'images locale doit soit comprendre une segmentation d'images (la catégorie 1) soit utiliser des descripteurs locaux (la catégorie 2).

Nous appelons les approches de la première catégorie, les approches d'indexation et de recherche d'images au niveau régions. Pour ces approches, la méthode de segmentation peut être simple ou complexe. L'objectif de cette segmentation est de décomposer une image en quelques régions. Dans le cas idéal, chaque région correspond à un objet réel. Le noyau des approches d'indexation et de recherche d'images au niveau régions est la mise en correspondance entre images. Cette mise

en correspondance doit tenir compte de deux caractéristiques. (1) Le nombre de régions varie d'une image à l'autre. (2) La segmentation n'est pas parfaite, une région peut être appariée à plus d'une région.

Dans [Carson 2002], les auteurs ont présenté un système appelé Blobworld pour l'indexation et la recherche d'images au niveau régions. Dans ce système, la requête peut être une ou plusieurs régions. Pour des requêtes contenant une région, soit v_i le vecteur de descripteurs de la région recherchée, la mise en correspondance entre la région recherchée et une image cible s'effectue en suivant les trois étapes :

- pour chacune de régions v_j de l'image cible, la distance de Mahalanobis entre v_i et v_j est définie par : $d_{ij} = (v_i - v_j)^T \Sigma (v_i - v_j)$
- la similarité entre deux régions est calculée par : $\mu_{ij} = e^{-\frac{d_{ij}}{2}}$
- la similarité entre la région recherchée et l'image cible : $\mu_i = \max_j \mu_{ij}$

Pour des requêtes contenant plusieurs régions, les auteurs ont appliqué la logique floue pour calculer la similarité entre cette requête et une image de la base en se basant sur celle des requêtes comprenant une seule région. Un exemple d'une requête contenant plusieurs régions est "retrouver des images comportant (une région similaire à la région 1) et (une région similaire à la région 2 ou une région similaire à la région 3)". La similarité entre cette requête et une image cible est déterminée par : $\min\{\mu_1, \max\{\mu_2, \mu_3\}\}$ où μ_1 , μ_2 , et μ_3 sont les similarités entre cette image et trois requêtes comprenant les régions 1, 2 et 3. La méthode de Carson et al. [Carson 2002] tient compte de la première caractéristique (le nombre de régions varie d'une image à l'autre). Cependant, la deuxième caractéristique (la segmentation n'est pas parfaite, une région peut être appariée à plus d'une région) n'est pas prise en compte. Nous présentons par la suite deux méthodes pouvant tenir compte des deux caractéristiques : l'une de Wang et al. [Wang 2001] et l'autre de Rubner et al. [Rubner 1998].

Soit :

- $R_1 = \{r_1, r_2, \dots, r_m\}$: un ensemble de régions de l'image recherchée ;
- $R_2 = \{r'_1, r'_2, \dots, r'_n\}$: un ensemble de régions d'une image cible de la base d'images ;
- d_{ij} : distance entre deux régions r_i et r'_j ;
- p_i, p'_j : degrés d'importance des régions r_i et r'_j , $\sum_{i=0}^m p_i = \sum_{j=0}^n p'_j = 1$.

Wang et al. [Wang 2001] ont proposé une méthode de mise en correspondance nommée IRM (Integrated Region Matching) pour le système SIMPlicity (Semantics-Sensitive Integrated Matching for Picture Libraries).

S est une matrice :

$$S = \begin{bmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,n} \\ s_{2,1} & s_{2,2} & \dots & s_{2,n} \\ \dots & \dots & \dots & \dots \\ s_{m,1} & s_{m,2} & \dots & s_{m,n} \end{bmatrix} \quad (2.1)$$

où chaque élément $s_{i,j}$ exprime la participation d'une paire de r_i et r'_j dans la mise en correspondance. $s_{i,j}$ doit vérifier la contrainte définie dans l'équation 2.2.

$$\sum_{j=0}^n s_{ij} = p_i, \quad \sum_{i=0}^m s_{ij} = p'_j \quad (2.2)$$

La valeur de $s_{i,j}$ est calculée de la manière : plus les régions r_i et r'_j sont appariées, plus la valeur de $s_{i,j}$ est élevée. La distance entre deux ensembles R_1 et R_2 déterminée par :

$$d(R_1, R_2) = \sum_{i,j} s_{ij} d_{ij} \quad (2.3)$$

Le degré d'importance p est déterminé en fonction de la taille la de région.

Rubner et al. [Rubner 1998] ont présenté une mise en correspondance entre deux images en se basant sur la distance EMD (Earth Movers Distance).

Soit f_{ij} la participation de deux régions r_i et r'_j , le problème linéaire est :

$$\min_F \sum_{i=1}^m \sum_{j=1}^n f_{ij} d_{ij} \quad (2.4)$$

sous les contraintes :

$$f_{ij} \geq 0, \quad 1 \leq i \leq m, \quad 1 \leq j \leq n. \quad (2.5)$$

$$\sum_{j=1}^n f_{ij} \leq p_i, \quad 1 \leq i \leq m \quad (2.6)$$

$$\sum_{i=1}^m f_{ij} \leq p'_j, \quad 1 \leq j \leq n \quad (2.7)$$

$$\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min\left(\sum_{i=1}^m p_i, \sum_{j=1}^n p'_j\right) \quad (2.8)$$

Avec la solution optimale f^*_{ij} , la distance entre deux ensembles est :

$$d(R_1, R_2) = \frac{\sum_{i,j} f^*_{ij} d_{ij}}{\sum_{i,j} f^*_{ij}} \quad (2.9)$$

Les deux méthodes de Wang et al. [Wang 2001] et de Rubner et al. [Rubner 1998] prennent en compte les deux caractéristiques mentionnées. Cependant, la méthode de Wang et al. [Wang 2001] se base sur un algorithme glouton qui donne par fois une solution optimale locale tandis que celle de Rubner et al. fournit une solution optimale globale.

La plupart des approches d'indexation et de recherche d'images s'effectuent sur des images d'exemple fournies par l'utilisateur. Le travail de Fauqueur et al. [Fauqueur 2006] a abordé la recherche d'images où l'image d'exemple n'est pas disponible. Les auteurs ont proposé un nouveau terme : l'image mentale. Des images

sont segmentées en régions. La technique de regroupement CA (Competitive Agglomeration) classe des régions en catégories. Soit (C_1, \dots, C_P) les P catégories, (p_1, \dots, p_P) les P prototypes respectivement déterminés pour la base d'images, pour chaque catégorie une région représentative r_i est choisie. Les auteurs ont présenté la relation de voisinage entre deux catégories : deux catégories C_q, C_j sont voisines si $d(C_q, C_j) = \|p_q - p_j\|_{L^2} \leq \theta$.

Les requêtes sont sous la forme : "retrouver des images comprenant (ou pas) des régions appartenant aux types déterminés". Soit $PQC = \{C_{pq1}, C_{pq2}, \dots, C_{pqM}\}$ les M régions qui doivent être présentes dans les images de résultat, $NQC = \{C_{nq1}, C_{nq2}, \dots, C_{nqR}\}$ les R régions qui doivent ne pas être présentes dans les images de résultat. Les requêtes peuvent être exprimées par :

$Q = (C_{pq1} \text{ OR ses voisines}) \text{ AND } \dots (C_{pqM} \text{ or ses voisines}) \text{ AND NOT } (C_{nq1} \text{ or ses voisines}) \text{ AND NOT } \dots (C_{nqR} \text{ or ses voisines})$

Soit $S_N^\theta(C)$ l'ensemble d'images contenant une région de la catégorie C ou de ses catégories voisines qui sont déterminées par le seuil θ . En se basant sur $S_N^\theta(C)$, l'ensemble d'images contenant les catégories PQC et celui contenant les catégories NQC sont déterminés par $S_Q = \bigcap_{i=1}^M S_N^\theta(C_{pqi})$ et $S_{NQ} = \bigcap_{i=1}^R S_N^\theta(C_{nqi})$ respectivement. Enfin, l'ensemble de résultats pour la requête est $S_{result} = S_Q \setminus S_{NQ}$.

Pour les approches de la deuxième catégorie, les images ne doivent pas être segmentées. Les descripteurs locaux tels que les points d'intérêt sont extraits sur les images. La mise en correspondance entre images se base sur leurs points d'intérêt appariés. L'avantage des approches appartenant à cette catégorie est que les points d'intérêts permettent d'apparier des images prises à différentes conditions (p. ex. points de vue, lumière). Pour chacune des images, un ensemble de points d'intérêt est détecté. Il est à noter qu'une direction de recherche intéressante dans cette catégorie permet d'appliquer des techniques de recherche de textes à la recherche d'images. En considérant un point d'intérêt comme un terme, une image est équivalente à un document.

Pour cela, dans [Sivic 2008], les points d'intérêt sont regroupés, un terme est assigné à chaque groupe. Après avoir enlevé les termes qui sont trop fréquentes ou trop rares, la mise en correspondance deux documents (deux images) se base sur une technique dans la recherche de textes nommée le tf-idf (en anglais term frequency-inverse document frequency).

La fréquence du terme (term frequency) est simplement le nombre d'occurrences de ce terme dans le document considéré. Cette somme est en général normalisée pour éviter les biais liés à la longueur du document.

Soit d le document et t_i le terme, alors la fréquence du terme dans le document est :

$$tf_{t_i} = \frac{n_{id}}{n_d} \quad (2.10)$$

où n_{id} est le nombre d'occurrences du terme t_i dans le document d . Le dénominateur est le nombre d'occurrences de tous les termes dans le document d .

La fréquence inverse de document (inverse document frequency) est une mesure de l'importance du terme dans l'ensemble du corpus. Elle consiste à calculer le

logarithme de l'inverse de la proportion de documents du corpus qui contiennent le terme :

$$idf_{t_i} = \log \frac{N}{N_i} \quad (2.11)$$

où N est le nombre total de documents dans le corpus, N_i est le nombre de documents dans lesquels le terme t_i apparaît.

Finalement, le tf-idf est défini par :

$$tfidf_{t_i} = tf_{t_i} * idf_{t_i} \quad (2.12)$$

Soit V l'ensemble des termes déterminés pour une base d'images, l'image recherchée q est représentée par : $v_q = (tfidf_{t_1}, tfidf_{t_2}, \dots, tfidf_{t_V})$, une image p de la base est également représentée par : $v_p = (tfidf_{t_1}, tfidf_{t_2}, \dots, tfidf_{t_V})$. La similarité entre deux images :

$$sim(v_q, v_p) = \frac{v_q^T v_p}{\|v_q\|_2 \|v_p\|_2} \quad (2.13)$$

où $\|v\|_2$ est la norme L2 de v . En effet, le tf-idf est une méthode de pondération. Cette mesure permet d'évaluer l'importance d'un mot par rapport à un document extrait d'un corpus. Le poids augmente proportionnellement en fonction du nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence du mot dans le corpus. Cependant, le tf-idf ne prend pas en compte la position des termes. Cela est abordé par le travail de Tirilly et al. [Tirilly 2008]. Les auteurs ont transformé une image en une phase. Pour cela, les auteurs ont enlevé les points d'intérêt redondants, ont déterminé un axe pour tous les points et ont projeté les points d'intérêt sur l'axe déterminé. La figure 2.1 montre ce processus. Un modèle de langage a été employé afin de classer des phases. Soit w_n un terme, la possibilité d'avoir w_n si on connaît $n - 1$ termes précédents $Pr(w_n | w_1, \dots, w_{n-1})$. Étant donnée un modèle du langage L qui est calculé à partir d'un ensemble d'apprentissage T , la possibilité que les n termes $w_1 w_2 \dots w_n$ apparaissent ensemble est :

$$Pr_L(w_n | w_1, \dots, w_{n-1}) = \frac{C(w_1 w_2 \dots w_n)}{\sum_{w_i \in T} C(w_1, w_2 \dots w_i)} \quad (2.14)$$

où $C(w_1 w_2 \dots w_n)$ est le nombre d'occurrences de $w_1 w_2 \dots w_n$ dans T .

Avec le modèle du langage L , la possibilité de générer un document $d = w_1 w_2 \dots w_k$ est :

$$Pr_L(d) = \prod_{i=1}^k Pr_L(w_i | w_1, \dots, w_{i-1}) \quad (2.15)$$

En appliquant l'équation 2.14, cette possibilité devient :

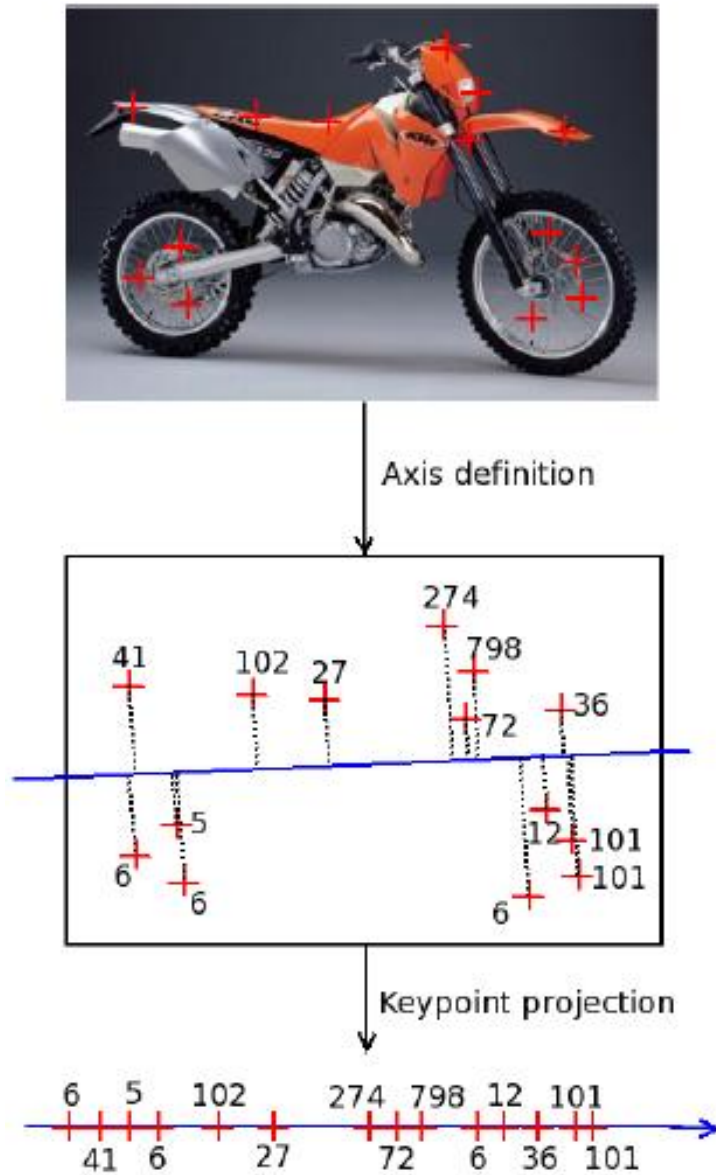


FIG. 2.1 – Transformation des points d'intérêt détectés dans une image à une phase de mots : après avoir enlevé des points redondants, un axe est déterminé pour tous les points. Les points sont projetés sur l'axe déterminé ([Tirilly 2008]).

$$Pr_L(d) \approx \prod_{i=1}^k Pr_L(w_i | w_{i-(n-1)}, \dots, w_{i-1}) \quad (2.16)$$

Soit C l'ensemble de classes apprises à partir l'ensemble d'apprentissage T , L_c le modèle du langage pour la classe c , pour un document quelconque d_{unk} , la classe du document est définie par :

$$c(d_{unk}) = \underset{c \in C}{\operatorname{argmax}} (Pr_{L_c}(d_{unk})) \quad (2.17)$$

En remplaçant un document par une image dans l'équation 2.17, une image est classée dans une de classes déterminées.

2.1.2 Ontologie

Des approches d'indexation et de recherche d'images basées sur l'ontologie cherchent tout d'abord à avoir une représentation formelle sous la forme d'une ontologie pour la connaissance du domaine. Des images sont ensuite indexées et retrouvées en utilisant des concepts de l'ontologie. Trois approches remarquables dans cet aspect sont l'approche de Mezaris et al. [Mezaris 2004], celle de Maillot et al. [Maillot 2005] et celle de Lim et al. [Lim 2003a], [Lim 2003b].

L'approche d'indexation et de recherche d'images à l'aide d'une ontologie de Mezaris et al. [Mezaris 2004] est montrée dans la figure 2.2.

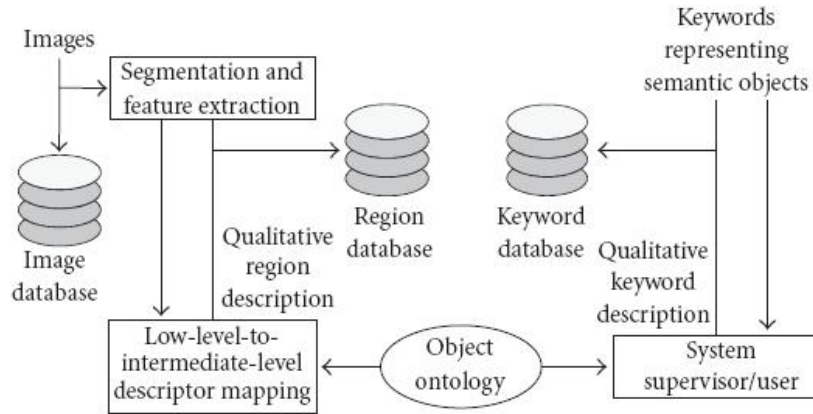


FIG. 2.2 – Indexation et recherche d'images avec une ontologie ([Mezaris 2004]).

Une ontologie qui définit la mise en correspondance entre des descripteurs à bas niveau extraits sur des régions et des termes à intermédiaire niveau est formulée par :

$$O := (\mathcal{D}, \leq_{\mathcal{D}}, \mathcal{R}, \sigma, \leq_{\mathcal{R}}) \quad (2.18)$$

où :

- \mathcal{D} : l'ensemble de concepts visuels (p. ex. la forme)
- \mathcal{R} : l'ensemble de concept de relations (p. ex. la position relative)
- $\leq_{\mathcal{D}}$: un ordre partiel qui décrit l'hierarchie de concepts (p. ex. la luminance est un sous-concept de l'intensité)
- σ : une fonction $\sigma : \mathcal{R} \rightarrow \mathcal{D}^+$, $\sigma(r) = (\sigma_{1,r}, \sigma_{2,r}, \dots, \sigma_{\sum,r})$ avec $\sigma_{i,r} \in \mathcal{D}$ (p. ex. $\sigma(r) = (\text{"position", position})$)
- $\leq_{\mathcal{R}}$: un ordre partiel qui décrit l'hierarchie de relations

La figure 2.3 montre la correspondance entre des descripteurs à bas niveau et des termes à intermédiaire niveau pour la luminance. Les requêtes sont exprimées en utilisant cette ontologie. La mise en correspondance entre des descripteurs à bas niveau et des termes à intermédiaire niveau dans [Mezaris 2004] consiste simplement à déterminer un intervalle de valeurs pour chaque terme.

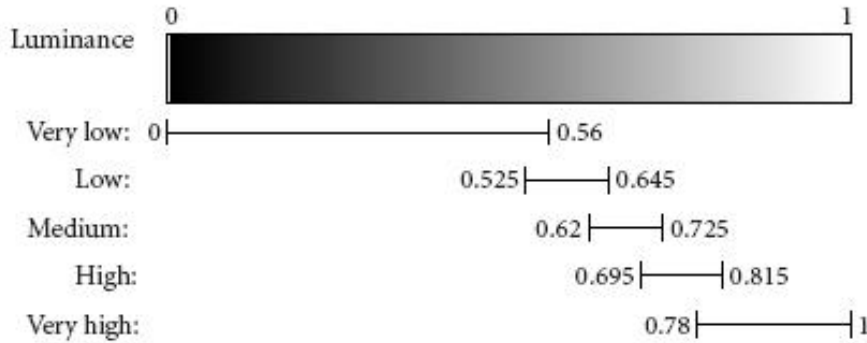


FIG. 2.3 – Correspondance entre des descripteurs à bas niveau et des termes à intermédiaire niveau pour la luminance ([Mezaris 2004]).

Le travail de Maillot et al. [Maillot 2005] a fourni deux contributions. Premièrement, une représentation formelle de l'ontologie est présentée. Deuxièmement, des techniques d'apprentissage faiblement supervisé sont utilisées afin de mettre en correspondance les descripteurs à bas niveau et les concepts de l'ontologie. Une ontologie comprenant 103 concepts visuels (p. ex. couleur, forme, taille, texture, relations spatiales) a été construite.

Lim et al. [Lim 2003a], [Lim 2003b] ont proposé le graphe conceptuel qui est équivalent à l'ontologie. Les auteurs ont défini le terme événement pour une image qui correspond à l'occasion pendant laquelle l'image est prise. Correspondant à un événement E_i , un modèle M_i comprenant deux facettes est construit. La première facette Mv_i est une représentation de ce modèle par les mots clés alors que la deuxième facette Mg_i correspond à un graphe conceptuel. L'ensemble de modèles M_i sont entraînés sur un ensemble d'images annotées. Pour un image x , la possibilité que x appartienne à M_i est $R(M_i, x)$:

$$R(M_i, x) = \alpha.S_v(Mv_i, x_{vk}) + (1 - \alpha).S_g(Mg_i, x_{gk}) \quad (2.19)$$

où α est un paramètre a priori, $S_v(Mv_i, x_{vk})$ et $S_g(Mg_i, x_{gk})$ sont les similarités entre l'image x et le modèle M_i selon le mot clé ou le graphe conceptuel. La similarité $S_v(Mv_i, x_{vk})$ est déterminée par la présence des mots clés dans l'image x et dans le modèle M_i . La similarité $S_g(Mg_i, x_{gk})$ est définie par la mise en correspondance entre le graphe conceptuel de l'image x et celui du modèle M_i . Cette mise en correspondance prend en compte la similarité entre des noeuds et entre des arcs de deux graphes. Un graphe conceptuel est un graphe orienté biparti fini comprenant des noeuds qui sont des concepts et des relations. Les relations incluent des relations de positions absolues, celles de positions relatives et celles de structure. Un concept est représenté comme : $[type : referent|w|ce]$ où $type$ est un type de concepts, $referent$ est une occurrence de concepts, w est le degré d'importance du concept dans l'image déterminé par la taille de région, ce est la valeur de confiance de la détection du concept. Les figures 2.4 et 2.5 montrent deux graphes conceptuels dont l'un pour une image de piscine et l'autre pour le modèle de l'événement "piscine". Il est à noter que la valeur de confiance de la détection des concepts pour les graphes conceptuels du modèle d'événement est 1.

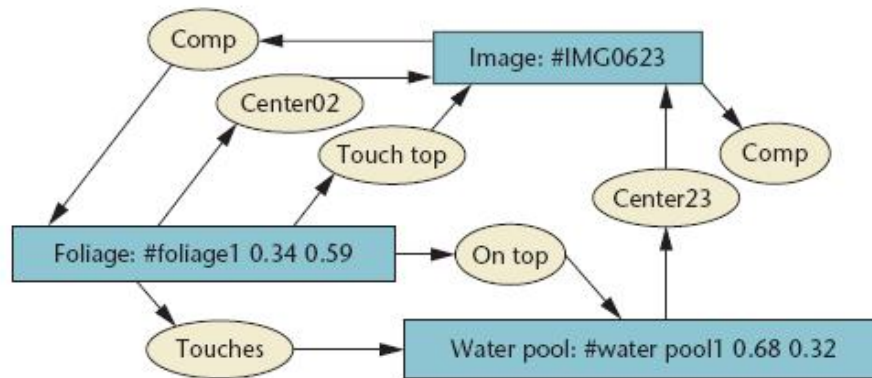


FIG. 2.4 – Graphe conceptuel construit pour l'image de piscine. Trois noeuds qui sont des concepts : "image", "foliage", "water pool". Les autres noeuds sont des relations ([Lim 2003b]).

2.1.3 Retour de pertinence

Les résultats de recherche d'information ne sont pas toujours parfaits en raison du fossé sémantique que nous expliquons dans le chapitre 1. La méthode pouvant combler ce fossé doit permettre de communiquer avec l'utilisateur. La recherche d'information interactive est une recherche d'information qui permet à l'utilisateur de faire un retour de pertinence. Le retour de pertinence est un processus qui consiste à apprendre à partir des retours de l'utilisateur et à trouver de nouveaux résultats pour répondre à l'utilisateur en se basant sur la connaissance apprise. L'approche

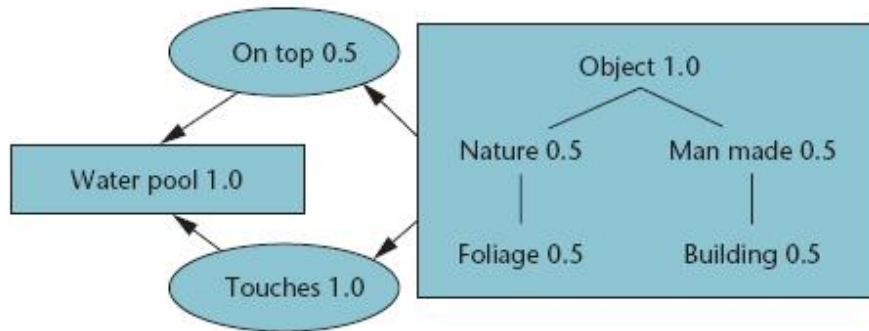


FIG. 2.5 – Graphe conceptuel déterminé pour le modèle d'événement "piscine". Deux noeuds qui sont des concepts : "water pool" et "object". Le concept "object" a deux sous-concepts : "nature" et "man made". Deux noeuds sont des relations : "on top" et "touches" ([Lim 2003b]).

permettant de faire un retour de pertinence comprend deux processus : un apprenant et un sélectionneur.

L'apprenant décide sur quelle partie de retours il va apprendre et comment il apprend. Le sélectionneur effectue la recherche en utilisant la connaissance apprise par l'apprenant et rend les nouveaux résultats à l'utilisateur.

L'utilisateur démarre le processus de recherche en exprimant sa requête. Le système effectue la mise en correspondance entre les éléments indexés et la requête. La liste des résultats ordonnés de manière décroissante par leurs similarités avec la requête est rendue à l'utilisateur. Si l'utilisateur est satisfait par la réponse, il finit le processus de recherche (sans avoir démarré le retour de pertinence). Sinon le retour de pertinence démarre. Il demande à l'utilisateur d'annoter les résultats obtenus comme exemples positifs et/ou exemples négatifs. L'apprenant est activé pour apprendre l'intention de l'utilisateur à partir de ses retours. Le sélectionneur vise à effectuer une nouvelle recherche et rendre de nouveaux résultats à l'utilisateur.

Nous présentons quelques définitions que nous utilisons pour le retour de pertinence :

- Une itération de recherche comporte un appel de l'apprenant et du sélectionneur.
- La session de recherche d'un utilisateur commence lorsque l'utilisateur formule sa requête et se termine au moment où il est satisfait par la réponse de recherche. Une session de recherche peut comporter plusieurs itérations de recherche ;
- Le retour de pertinence est à court terme si son sélectionneur n'emploie que ce que l'apprenant a appris lors de la session en cours pour trouver de nouveaux résultats ;

- Le retour de pertinence est à long terme si son sélectionneur emploie ce que l'apprenant a appris lors de la session en cours et d'autres sessions de recherche pour trouver de nouveaux résultats. Notons que les sessions de recherche peuvent être effectuées par des utilisateurs différents ;
- Le retour de pertinence est dit avec exemples positifs si son apprenant n'utilise que les exemples positifs ;
- Le retour de pertinence est dit avec exemples positifs et négatifs si son apprenant utilise les exemples positifs et négatifs.

Le retour de pertinence se base sur quatre hypothèses générales introduites par Crucianu et al. [Crucianu 2004] pour le retour de pertinence en indexation et recherche d'images :

- Il est possible de distinguer les objets pertinents et non pertinents en se basant sur leurs descripteurs ;
- Il existe un lien entre les espaces de descripteurs sur lesquels les objets sont représentés et les caractéristiques des objets que l'utilisateur veut chercher ;
- Le nombre d'objets pertinents d'une requête est relativement petit par rapport au nombre des objets dans la base ;
- L'utilisateur est volontaire pour juger les résultats.

Crucianu et al. [Crucianu 2004] ont précisé le lien sur l'origine et la nature des informations qu'on peut exploiter avec le retour de pertinence (voir tableaux 2.1).

TAB. 2.1 – Origine et nature des informations qui peuvent être exploitées par le retour de pertinence [Crucianu 2004].

Origine	Temps		
	autres sessions	session en cours	itération en cours
a priori	connaissance du domaine	contexte de la session	-
autres utilisateurs	corrélation des recherches	-	-
l'utilisateur	modèle de perception de similarité	retours des itérations antérieures	retours de l'itération en cours

Nous pouvons voir que pour une session de recherche d'un utilisateur, la connaissance peut provenir a priori, d'autres utilisateurs et de cet utilisateur. La connaissance a priori peut être la connaissance du domaine et le contexte de la session de recherche. La connaissance provenant des autres utilisateurs peut être la corrélation des recherches. La connaissance provenant de cet utilisateur peut être le modèle de perception de similarité, ses retours des itérations antérieures et de l'itération en cours.

Les techniques de retour de pertinence peuvent être classées en trois grandes familles : le retour de pertinence basée sur la modification de requête appelé QVM (Query Vector Modification), celui basé sur le rôle du descripteur appelé FRE (Feature Relevance Estimation), et celui basé sur la classification.

Les techniques de retour de pertinence basées sur la modification de requête consiste à reformuler des requêtes en utilisant des images positives et des images négatives de manière que des résultats de la prochaine itération comprennent plus de résultats pertinents. Soit $X_i^{(j)}$, $X_i^{(j+1)}$ les formulations de requête correspondant à une image recherchée i à l'itération j et $j + 1$, $X_i^{(j+1)}$ est déterminée par $X_i^{(j)}$ et des images positives et négatives selon la formule suivante :

$$X_i^{(j+1)} = \alpha X_i^{(j)} + \beta \sum_{Y_k \in R} \frac{Y_k}{|R|} - \gamma \sum_{Y_k \in N} \frac{Y_k}{|N|} \quad (2.20)$$

où R et N sont respectivement l'ensemble d'images positives et celui d'images négatives. Les paramètres α , β , γ décident la participation de chaque composant. La figure 2.6.a montre un exemple de retour de pertinence basée sur la modification de requête dans le cas où deux descripteurs f_1 , f_2 sont utilisés.

Les techniques de retour de pertinence basées sur le rôle du descripteur visent à déterminer l'importance de chaque descripteur pour la mise en correspondance entre des images. L'importance de chaque descripteur est initialisée par la même valeur.

Soit $X = (x_1, x_2, \dots, x_d)$, $Y = (y_1, y_2, \dots, y_d)$ deux vecteurs de descripteurs de deux images, la distance entre deux images :

$$Dist(X, Y) = \sum_{i=1}^d (x_i - y_i)^2 \quad (2.21)$$

L'importance de chaque descripteur w_i est mise à jour tout au long de la session de recherche. Plus le descripteur est pertinent, plus son importance est élevée. Lorsque l'importance de chaque descripteur est déterminée, la distance entre deux images (cf. équation 2.22) devient :

$$Dist(X, Y) = \sum_{i=1}^d w_i (x_i - y_i)^2 \quad (2.22)$$

La figure 2.6.b montre un exemple du retour de pertinence basé sur le rôle du descripteur dans le cas où deux descripteurs f_1 , f_2 sont utilisés. Au début, l'importance de chaque descripteur est égale. Après une itération, le descripteur f_1 devient plus important que le descripteur f_2 .

Les techniques de retour de pertinence basées sur la classification entraînent des classifieurs pour des images positives et/ou pour des images négatives. Des images de la base sont comparées avec ces classifieurs. Les machines à vecteurs de support (SVM) et les classifieurs bayésiens sont largement choisis.

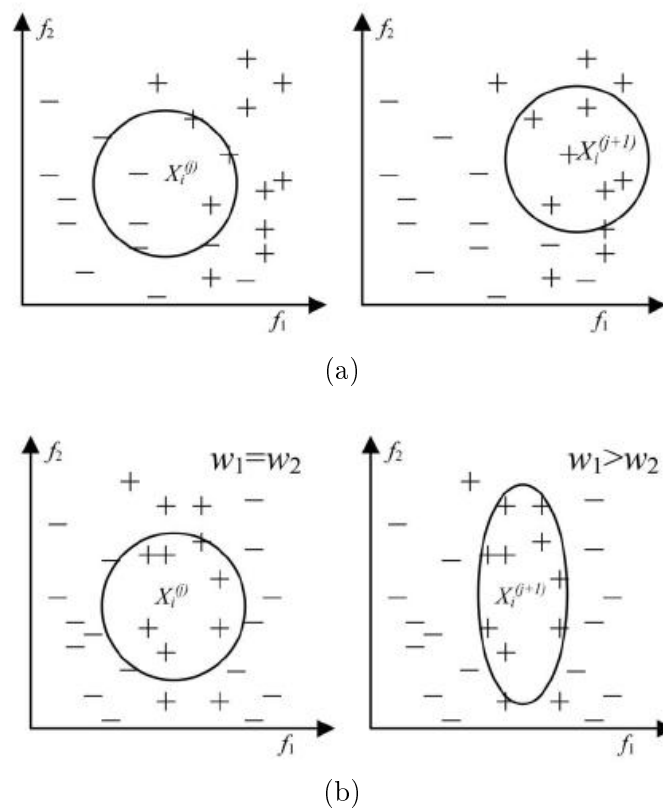


FIG. 2.6 – Illustration de retour de pertinence basé (a) sur la modification de requête (b) et sur le rôle de chaque descripteur dans le cas où deux descripteurs f_1, f_2 sont utilisés ([Yin 2005]).

Dans cette section, nous présentons quelques techniques de retour de pertinence. Des synthèses et des analyses de techniques de retour de pertinence en indexation et recherche d'images se trouvent dans les articles [Rui 1998], [Rui 2001] et [Crucianu 2004].

Il est à noter que des techniques de chaque famille ont des points forts et aussi des points faibles. Des techniques basées sur la classification ont besoin de plusieurs itérations alors que celles basées sur la modification de requête et le rôle du descripteur n'ont pas de contrainte pour le nombre d'itérations. En profitant des points forts de chaque famille, Yin et al. [Yin 2005] ont proposé plusieurs stratégies permettant de combiner des techniques des trois familles.

Les techniques proposées par Goldmann et al. [Goldmann 2006] font partie de la troisième famille. Des retours de l'utilisateur sont faits sur les images globales. Un modèle $M = (\mu, \sigma)$ est entraîné en utilisant des images positives jugées par l'utilisateur $j \in J$. Pour une image cible de la base d'images, $k \in K$, la distance entre cette image et le modèle entraîné sera calculée. Les images dont la distance est la plus faible sont retournées. La distance entre l'image x et le modèle M est définie par :

$$d(x) = \frac{1}{2} \sum_i \left(\frac{x_i - \mu_i}{\sigma_i} \right)^2 \quad (2.23)$$

La différence entre le retour de pertinence à court terme et à long terme est la façon d'entraînement du modèle M . Pour le retour de pertinence à court terme, le modèle $M = (\mu, \sigma)$ est construit à partir d'un ensemble d'images positives J à l'itération en cours :

$$\mu_i = \frac{1}{|J|} \sum_j x_{ji}; \sigma_i^2 = \frac{1}{|J|} \sum_j (x_{ji} - \mu_i)^2 \quad (2.24)$$

Des retours d'autres itérations et d'autres sessions ne sont pas pris en compte.

Pour le retour de pertinence à long terme, le modèle $M_j = (\mu_j, \sigma_j, n_j)$ est construit pour chaque image j en utilisant des retours de la même session et d'autres sessions. Ce modèle est mis à jour comme suit :

$$n'_j = n_j + |J| \quad (2.25)$$

$$\mu'_{ji} = \frac{1}{n'_j} (n_j \mu_{ji} + n \mu_i) \quad (2.26)$$

$$\sigma'^2_{ij} = \frac{1}{n'_j} ((n_j - 1) \sigma_{ji}^2 + n_j \mu_{ij}^2 - n'_j \mu'^2_{ji} + \sum_j x_{ji}^2) \quad (2.27)$$

Le modèle combiné $M = (\bar{\mu}, \bar{\sigma})$ est déterminé en utilisant les modèles entraînés pour toutes les images positives :

$$\bar{\mu}_i = \frac{\sum_j n'_j \mu'_{ji}}{\sum_j n'_{ji}}; \bar{\sigma}_i = \frac{\sum_j n'_j \sigma'_{ji}}{\sum_j n'_{ji}} \quad (2.28)$$

Dans [Jing 2003], [Jing 2004], les auteurs ont présenté une méthode de retour de pertinence de la troisième famille pour la recherche d'images au niveau régions. Les machines à vecteurs de support (SVM) sont choisies. Puisque la mise en correspondance entre deux images se base sur la distance EMD (cf. équation 2.9) entre deux ensembles de régions, dans le noyau gaussienne des SVM, la distance d est remplacée par la distance EMD :

$$k_{EMD}(x, y) = \exp\left(\frac{-EMD(x, y)}{2\sigma^2}\right) \quad (2.29)$$

où EMD est définie dans l'équation 2.3. Les SVM sont entraînés tout au long de la session de recherche.

2.1.4 Discussions

Les trois aspects importants (la recherche locale, l'ontologie, le retour de pertinence) en indexation et recherche d'images sont présentés. La recherche locale permet de tenir compte de la disposition des descripteurs. L'ontologie et le retour de pertinence essaient d'enrichir la connaissance des systèmes. La connaissance dans l'ontologie est la connaissance du domaine, a priori alors que celle dans le retour de pertinence provient des utilisateurs. Les recherches en cours en indexation et recherche d'images consistent à trouver de nouveaux descripteurs ou de nouvelles méthodes de mise en correspondance ou de nouvelles techniques de retour de pertinence. Les techniques d'indexation qui permettent de travailler avec de grandes bases d'images telles que la technique de Berrani et al. [Berrani 2002] sont également étudiées. Nous analysons à la fin de ce chapitre la relation entre l'indexation et la recherche d'images et notre travail ce qui est dédié à l'indexation et à la recherche de vidéos pour la vidéosurveillance. Pour le retour de pertinence, nous utilisons dans notre travail de thèse les mêmes termes définis qu'en indexation et recherche d'images.

2.2 Indexation et recherche de vidéos

2.2.1 Deux types de vidéos et terminologies

Afin de donner une analyse globale des approches proposées pour l'indexation et la recherche de vidéos et d'indiquer où se situe notre approche, nous classons les approches en deux grandes familles selon le type de vidéo traitée. Il existe deux types de vidéos [Xiong 2006] : la vidéo dont le contenu est scénarisé (vidéo scénarisée) et non scénarisé (vidéo non scénarisée).

La vidéo scénarisée : une vidéo est scénarisée si elle est produite selon un script.

Les journaux télévisés, les films sont des exemples de vidéos scénarisées.

La vidéo non scénarisée : une vidéo est non scénarisée si elle n'est pas produite selon un script.

Les enregistrements de réunion, de vidéosurveillance sont des vidéos non scénarisées. Dans les vidéos non scénarisées, les événements sont spontanés.

2.2.2 Indexation et recherche de vidéos scénarisées

2.2.2.1 Terminologies associées à la représentation des vidéos scénarisées

Plan de vidéo (video shot) : un plan vidéo est une séquence de frames qui sont acquis de manière continue par une seule caméra.

Le plan de vidéo est une unité de base de vidéo. Une vidéo peut être représentée par un ensemble de ses plans.

Image clé (keyframe) : une image clé est une image qui représente significativement le contenu visuel d'un plan de vidéo.

Selon la complexité du plan, une ou plusieurs images clés peuvent être choisies.

Scène : une scène est un ou plusieurs plans qui partagent le même contenu en terme d'actions, de lieux et de temps. Ces plans sont souvent consécutifs.

[Corridoni 1998]

Groupe : un groupe est une représentation intermédiaire entre la scène et le plan.

Un groupe est défini comme un ensemble de plans consécutifs ou semblables.

2.2.2.2 Indexation et recherche de vidéos scénarisées

La phase d'indexation consiste à analyser la structure et le contenu des vidéos scénarisées. L'analyse de la structure des vidéos scénarisées consiste à représenter une vidéo par ses composantes (plans, images clés, groupes, scènes) tandis que l'analyse du contenu vise à extraire l'information de ces composantes. Nous présentons les approches proposées pour l'indexation et la recherche de deux types de vidéos scénarisées : les journaux télévisés et les films. L'analyse de la structure des vidéos est composée de la segmentation temporelle de vidéos, de la détection des images clés et du regroupement des plans en groupes et en scènes. La vidéo est tout d'abord décomposée en plans (par la segmentation temporelle de vidéos). Les images clés sont ensuite détectées pour chacun des plans (par la détection des images clés). Les plans peuvent être regroupés en groupe et en scène (par le regroupement des plans en groupes et en scènes).

La segmentation temporelle de vidéos en plans et la détection des images clés sont le plus souvent présentées dans les approches d'indexation et de recherche de vidéos, néanmoins le regroupement des plans en groupes et en scènes est rarement effectué. La figure 2.7 illustre la représentation et l'analyse de la structure des vidéos scénarisées.

La segmentation temporelle de vidéos en plans détecte la transition entre les plans. C'est pourquoi, elle est également connue sous le nom de détection de transitions. Les transitions peuvent être soit brusques, soit progressives. Dans le premier cas, on passe directement d'un plan à l'autre, alors que dans le second, un effet (p.

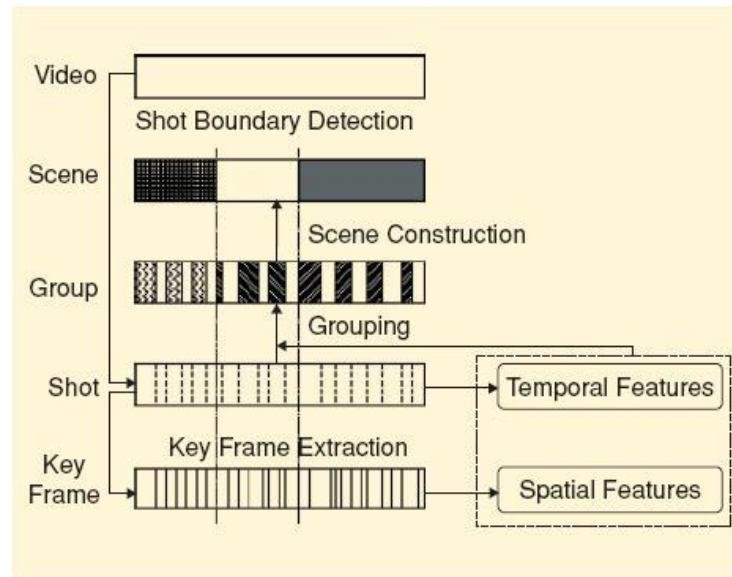


FIG. 2.7 – Représentation et analyse de la structure des vidéos scénarisées ([Xiong 2006]).

ex. volet, fondu, etc) est ajouté entre les deux plans. La segmentation a bénéficié de beaucoup d'effort et a obtenu de bon résultats. Deux états de l'art sur les approches de segmentation de vidéos ont été introduits [Koprinska 2001], [Cotsaces 2006].

La détection des images clés consiste à déterminer dans un plan un ou plusieurs frames qui représentent significativement le contenu du plan. En général, le premier frame et le dernier sont pris comme images clés. Un ou plusieurs frames qui sont largement différents d'autres frames selon une mesure de similarité sont considérés comme des images clés. Les approches de détection des images clés se différencient par leur descripteurs, leur mesures de similarité et leur stratégies de choix. Une évaluation des approches de détection des images clés a été donnée dans [Pickering 2003].

Le regroupement des plans en groupe et en scène consiste à retrouver le script sur lequel la production de la vidéo s'appuie. En effet, les scènes sont définies comme un groupe de plans cohérents qui présente un sens pour l'utilisateur. Le problème principal repose sur la définition de la cohérence des plans dans une scène. Est-ce une même unité de lieu, de personne, de thématique ? Contrairement aux plans, la définition des scènes repose sur une corrélation sémantique subjective.

De nombreuses approches sont proposées pour les journaux télévisés, il y a donc un besoin de comparer leurs performances. Afin de comparer ces approches, il est nécessaire de tester ces approches sur des bases communes de journaux télévisés. Une de ces bases nommée TRECVID a été proposée par NIST (National Institute of Standards and Technology) en 2001. TRECVID est en effet un grand corpus de journaux télévisés dont le contenu est varié. Il est à noter que TRECVID en 2008 a

fourni une base de vidéos de vidéosurveillance afin d'évaluer des modules d'analyse vidéo. Une liste de concepts associés aux vidéos est prédéterminée. Cette liste évolue au fur et à mesure dans le temps pour :

- augmenter la spécificité ;
- augmenter la complexité ;
- avoir plus de concepts rares avec peu d'exemples ;
- avoir plus d'objets et d'événements.

TRECVID contient quatre types de concepts : objets, scènes, événements et personnes particulières. Les approches sont évaluées par leur capacités de détection des concepts. La détection des concepts d'un plan de vidéo consiste à classer ce plan dans une des classes prédéfinies pour les concepts en employant les descripteurs extraits sur ce plan. Les approches se distinguent par les descripteurs, les classifieurs et la présence ou non de fusion. Parce que de nombreux descripteurs et classifieurs sont disponibles, il est nécessaire de fusionner soit les descripteurs (connu sous le nom de fusion précoce) [Ayache 2007] soit les classifieurs (connu sous le nom de fusion tardive) [Souvannavong 2005], [Ayache 2007]. Une vue d'ensemble des approches dans TRECVID a été présentée dans [Naphade 2004].

La recherche de journaux télévisés consiste à retrouver des plans qui correspondent à une requête. La requête est notamment formulée par un concept ou une combinaison de concepts. D'autres types de requêtes sont également possibles. Dans [Snoek 2007a], la requête peut être formulée par des descriptions textuelles et/ou des concepts.

Afin d'améliorer les résultats de recherche de plans vidéo, deux grandes directions ont été explorées. La première direction est d'enrichir la sémantique des concepts en établissant le lien entre les détecteurs de concepts et une ontologie générique [Snoek 2007a]. Concrètement, 363 détecteurs entraînés de MediaMill ¹ et LSCOM (Large Scale Ontology for Multimedia) ont été établis avec WordNet ². La deuxième direction est d'interagir avec l'utilisateur. Un système interactif a été proposé par Snoek [Snoek 2007b]. L'utilisateur formule une requête par les concepts, les exemples et les descriptions textuelles. En utilisant le retour de l'utilisateur, le système combine les résultats de différents types de requête pour améliorer les résultats retrouvés. Tandis que dans [Hauptmann 2008], les auteurs ont prédéterminé des descripteurs appropriés pour chaque classe de requête. Par exemple, pour déterminer le nom d'une personne, la présence de visages, la taille, la position et la reconnaissance de visages sont des descripteurs appropriés. Les descripteurs appropriés d'une classe vont avoir un rôle plus important que d'autres descripteurs quand on calcule la similarité entre la requête et les informations indexées. Une requête fournie par l'utilisateur va être classifiée dans une des classes prédéfinies. Cinq classes sont définies dont une classe des noms de personne (p. ex. retrouver les plans contenant Yasser Arafat), une classe des noms d'objet (p. ex. retrouver des plans contenant le logo de Mercedes), une

¹<http://www.science.uva.nl/research/mediamill/index.php>

²<http://wordnet.princeton.edu/>

classe des objets généraux (p. ex. retrouver des plans avec un ou plusieurs chats), une classe de scènes (p. ex. retrouver des plans de la plage) et une classe des sports (p. ex. retrouver des plans contenant un but de football).

En plus des approches précédentes, il existe également des approches de l'indexation et la recherche de journaux télévisés qui n'effectuent pas la détection des concepts. Au lieu de détecter des concepts dans la phase d'indexation et de les comparer dans la phase de recherche, elles gardent des descripteurs de bas niveaux dans la phase d'indexation et améliorent les résultats de recherche soit par l'interaction avec l'utilisateur [Zhai 2006], soit par l'application de bonnes mesures de similarité [Peng 2007], [Basharat 2007]. De plus, d'autres caractéristiques des journaux télévisés sont prises en compte telles que le style du script sur lequel l'édition de vidéos se base, la présence des costumes [Jaffré 2004], [Jaffré 2005].

Dans le travail de Zhai et al. [Zhai 2006], en se basant sur un module de reconnaissance automatique de la parole qui permet de transformer la parole dans une vidéo en textes, les requêtes sont initialisées par un ou plusieurs mots clés. Les auteurs ont proposé une méthode de retour de pertinence afin d'améliorer des résultats. Ensuite, les résultats sont raffinés par la similarité visuelle entre des plans retrouvés. La figure 2.8 montre l'architecture de cet approche.

Soit Q_i la requête lors de l'itération i , D^+ , D^- des plans positifs et négatifs. En déterminant des mots clés sur des plans positifs et négatifs, l'histogramme des mots positifs et celui des mots négatifs sont déterminés par :

$$WH_{D^+} = \{(a_1^+, W_1^+), (a_2^+, W_2^+), \dots, (a_m^+, W_m^+)\}$$

$$WH_{D^-} = \{(a_1^-, W_1^-), (a_2^-, W_2^-), \dots, (a_m^-, W_m^-)\}$$

où a_i est le nombre d'occurrences normalisés de mot clé W_i .

Étant donné un plan S de la base de vidéos, la similarité entre le plan S et la requête lors de l'itération $i+1$ est définie par :

$$R(S) = VP(WH_S, WH_{D^+}) - VP(WH_S, WH_{D^-}) \quad (2.30)$$

où $VP()$ est le produit scalaire entre deux vecteurs.

Afin de raffiner des résultats, la distance EMD (cf. équation 2.9) entre deux images clés des plans est utilisée.

Peng et al. [Peng 2007] ont calculé la similarité entre deux plans de vidéos par la couleur et le mouvement :

$$Sim(X, Y_k) = \omega_1 * Sim_{color}(X, Y_k) + \omega_2 * Sim_{motion}(X, Y_k) \quad (2.31)$$

où ω_1 et ω_2 sont des degrés d'importance de la couleur et du mouvement.

Soit $X = \{x_1, x_2, \dots, x_p\}$ l'ensemble de frames dans le plan X , $Y_k = \{y_1, y_2, \dots, y_q\}$ l'ensemble de frames dans le plan Y_k de la base. En se basant sur les histogrammes des composants de couleur HSV, la similarité entre le frame x_i et le frame y_j est déterminée par les équations 2.32 et 2.33.

$$w_{ij} = \frac{1}{A(x_i, y_j)} \sum_h \sum_s \sum_v \min(H_i(h, s, v), H_j(h, s, v)) \quad (2.32)$$

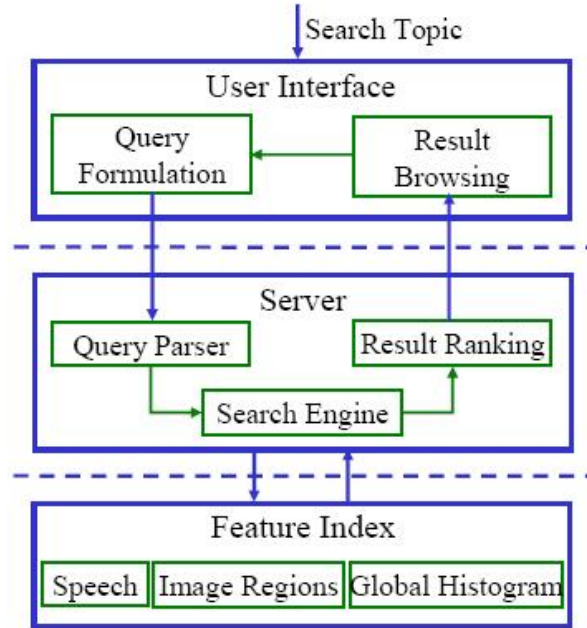


FIG. 2.8 – Indexation et recherche de journaux télévisés en se basant sur l'interaction avec l'utilisateur ([Zhai 2006]).

$$A(x_i, y_j) = \min\left\{\sum_h \sum_s \sum_v H_i(h, s, v), \sum_h \sum_s \sum_v H_j(h, s, v)\right\} \quad (2.33)$$

En appliquant la distance appelée OM (optimal matching) [Xiao 1993], la distance par la couleur entre deux plans est définie par :

$$Sim_{color}(X, Y_k) = \frac{\omega_{OM}(X, Y_k)}{\min(p, q)} \quad (2.34)$$

La distance OM cherche à avoir une mise en correspondance optimale. Cependant, elle est différente de la distance EMD car elle permet d'apparier un frame à un seul frame. La distance EMD permet d'apparier un frame à plusieurs frames.

Pour le mouvement, l'angle et l'intensité du mouvement qui sont définis dans MPEG-7 sont utilisés. L'angle est divisé en 8 niveaux, un histogramme d'intensité dont chaque élément est la somme des intensités correspondant au même niveau de l'angle est créé. La similarité entre deux plans par le mouvement est définie par les équations 2.35 et 2.36.

$$Sim_{motion}(X, Y_k) = \frac{1}{A(H_X, H_{Y_k})} \sum_{angle} \min\{H_X(angle), H_{Y_k}(angle)\} \quad (2.35)$$

$$A(H_X, H_{Y_k}) = \max\left\{\sum_{angle} H_X(angle), \sum_{angle} H_{Y_k}(angle)\right\} \quad (2.36)$$

Le travail de Jaffré et al. [Jaffré 2004], [Jaffré 2005] consiste à indexer des journaux télévisés en se basant sur le costume des présentateurs (voir figure 2.9). Des costumes sont détectés sur les frames par deux manières. Si la reconnaissance de visages est disponible, la localisation du costume se base sur la position du visage. Sinon, la détection du costume basée sur la technique appelée Mean shift [Comaniciu 2002] est effectuée.

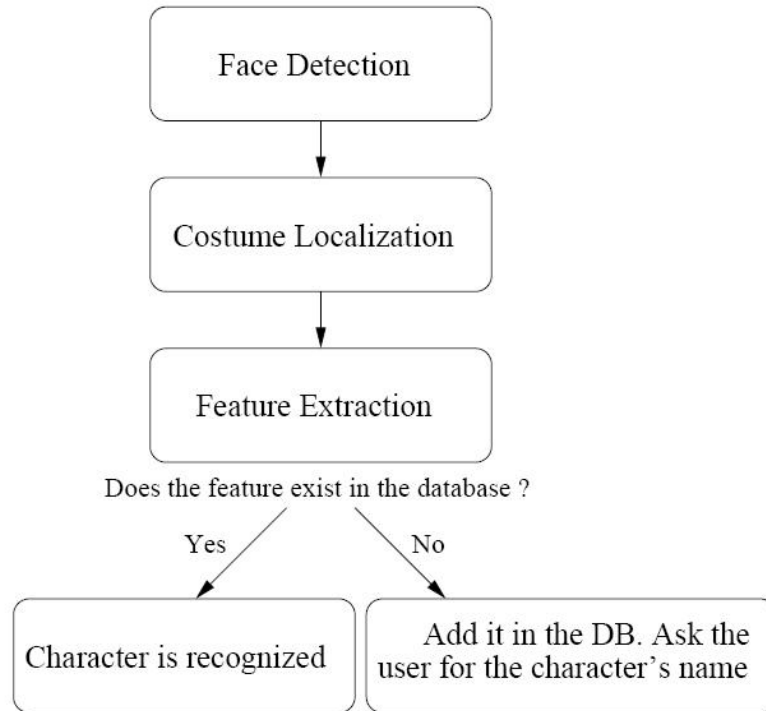


FIG. 2.9 – Reconnaissance de personnes dans les journaux télévisés basé sur leurs vêtements quand la reconnaissance de visages est disponible ([Jaffré 2004]).

Soit $\hat{q} = \{\hat{q}_u\}_{i=1,\dots,m}$ l'histogramme des couleurs du costume détecté dans le frame, $\hat{p} = \{\hat{p}_u\}_{i=1,\dots,m}$ l'histogramme des couleurs d'un costume de la base. La similarité entre deux costumes est déterminée par :

$$\rho(\hat{q}, \hat{p}) = \sum_{u=1}^m \sqrt{\hat{q}_u * \hat{p}_u} \quad (2.37)$$

Plus la valeur de ρ est grande, plus les deux costumes sont similaires.

Quant à l'indexation et la recherche de films, l'approche la plus connue a été introduite par Sivic et al. [Sivic 2006], [Sivic 2008]. Cette approche est également connue sous le nom de Video Google. Le but de cette approche est de retrouver les objets spécifiques dans des plans de films. Un ensemble d'images clés est détecté pour chacun des plans de films. Afin de retrouver les objets spécifiques qui peuvent

être filmés par les points de vue différents, les auteurs ont employé les régions covariantes affines (affine covariant regions). Un ensemble de régions covariantes affines sont extraites à partir des images clés des plans. Les régions covariantes affines détectées sont regroupées en classes pour avoir une représentation compacte des images clés. En représentant une classe par un terme, une image clé peut être considérée comme un document contenant certains termes. L'image d'exemple de l'utilisateur est également représentée par les termes. Le tf-idf qui est présenté en indexation et recherche d'images a été appliqué sur ces documents.

2.2.3 Indexation et recherche de vidéos non scénarisées

2.2.3.1 Terminologies associées à la représentation des vidéos non scénarisées

Play and Break : *un play est un ensemble de frames contenant des informations importantes à analyser et inversement pour un break.*

Dans les vidéos de vidéosurveillance, un play est la période où les objets sont apparus ou les événements d'intérêt auront lieu dans la scène.

Marqueur auditif : *est une séquence de frames consécutifs représentant une classe auditive qui correspond à un événement d'intérêt.*

Un bruit anormal détecté dans une vidéo de vidéosurveillance est un marqueur auditif. Il permet de reconnaître la chute de personne par exemple.

Marqueur visuel : *une séquence de frames consécutifs contenant visuellement un événement d'intérêt.*

Une séquence de frames contenant une personne qui est dans la cuisine est un marqueur visuel.

Highlight candidate : *une séquence de frames identifiés par les marqueurs auditifs et visuels.*

Highlight group : *un groupe de highlight candidates*

Les scénarios dans les vidéos de vidéosurveillance sont les *Highlight candidate* et *Highlight group*.

2.2.3.2 Indexation et recherche de vidéos non scénarisées

L'analyse d'une vidéo non scénarisée est effectuée d'une manière ascendante. Elle consiste à détecter les *plays and breaks*, identifier les marqueurs auditifs et visuels, déterminer les *highlight candidates* et les regrouper en *highlight groups*. La représentation et l'analyse d'une vidéo non scénarisée sont illustrées dans la figure 2.10.

L'indexation et la recherche de vidéos de sports, de réunion et de vidéosurveillance font partie de l'indexation et de la recherche de vidéos non scénarisées. Nous remarquons qu'il existe deux types de vidéos de sport : l'un contient des vidéos de sport prises par les caméras et l'autre sont des vidéos de sport télédiffusées. Les vidéos de sport télédiffusées sont créées par la tâche d'édition à partir des plans de vidéos acquis par les caméras. L'édition correspond au choix des plans et à leur

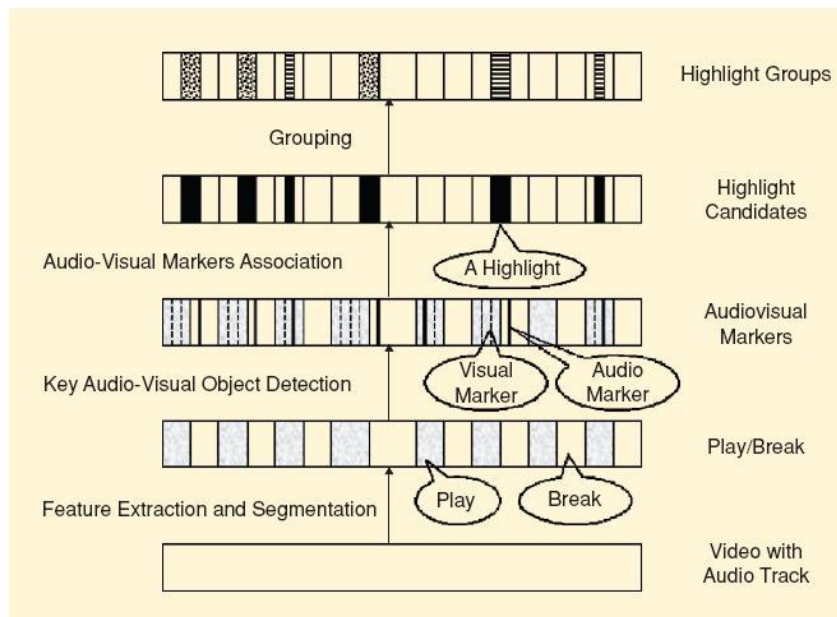


FIG. 2.10 – Représentation de vidéos non scénarisées à différents degrés de granularité (*break and play*, *marqueur visuels et auditifs*, *highlight candidate* et *highlight group*) et analyse des vidéos non scénarisées ([Xiong 2006]).

montage avant diffusion. Les vidéos du premier type sont des vidéos non scénarisées tandis que celles du deuxième type sont des vidéos scénarisées. Pour les vidéos de deuxième type, après une phase de prétraitement permettant d'identifier les plans, l'analyse de ces plans est celle de vidéos non scénarisées.

Notre thèse se focalise sur l'indexation et la recherche de vidéos pour la vidéosurveillance. Nous présentons brièvement dans cette section les approches proposées pour l'indexation et la recherche de vidéos de sport et d'enregistrements de réunion. Nous analysons en détail les approches dédiées à l'indexation et la recherche de vidéos pour la vidéosurveillance dans la section suivante.

2.2.3.3 L'indexation et la recherche de vidéos de sport

Avoir un résumé de vidéo de sports pour qu'il puisse être efficacement envoyé par Internet est une application intéressante qui a attiré beaucoup l'attention de chercheurs dans le domaine de la vision par ordinateur. Les vidéos télédiffusées ont deux types de connaissances a priori : la connaissance liée à la nature du sport considéré et celle liée à la production des vidéos télédiffusées. La connaissance a priori liée à la nature du sport sont les informations intrinsèques à la nature et aux règles du sport étudié, telles que :

- la surface de jeu ;
- le nombre de joueurs ;

- le déroulement du jeu.

La connaissance liée à la production des vidéos télédiffusées correspond aux règles de montage. Les règles de montage déterminent les plans, leurs ordres de diffusion et les rediffusions ajoutées. Les rediffusions proposent des ralentis de la scène précédente, elles interviennent immédiatement après qu'un événement a été jugé suffisamment intéressant. Il existe quatre classes de plans [Kijak 2003] :

- plan d'ensemble/général ou vue du terrain : il s'agit de plans larges fournissant une vue globale de l'aire de jeu ;
- plan moyen : il désigne un zoom de la caméra sur un champ restreint de l'aire de jeu ou un cadrage en pied du sujet ;
- plan rapproché ou gros plan : il désigne un cadrage à hauteur de poitrine au visage ou du visage sur le joueur, le public et l'arbitre ;
- plan de vue du public.

Les vidéos de sport peuvent être résumées par l'ensemble de *plays and breaks*, d'événements d'intérêt. Nous utilisons un seul terme événement pour les marqueurs visuels, les marqueurs auditifs et les *highlights*.

Au niveau du *play and break*, l'objectif est d'identifier des *plays and breaks* dans une vidéo. Ekin et al. [Ekin 2003b] propose une approche générique de détection des *breaks* et *plays* en se basant sur le type et la durée des plans vidéo. En se basant sur les analyses suivantes, l'approche de Ekin et al. est montrée dans la figure 2.11 :

- Les plans généraux habituellement correspondent aux *plays* tandis que les plans rapprochés correspondent aux *breaks* ;
- Un plan général peut être inséré pendant le *break*. La durée de ce plan est inférieure à celle des plans généraux correspondant aux *plays*.
- Un plan rapproché peut être inséré entre deux plans généraux pour souligner certains joueurs. La durée de ce plan est inférieure à celle des plans rapprochés correspondant aux *break*.

Au niveau des événements, l'objectif est d'identifier des événements prédéterminés dans une vidéo. Dans le cadre des événements sportifs, les événements sont déterminés par la nature du sport traité.

- pour le football : détection de buts ;
- pour le basketball : détection des paniers ;
- pour le baseball : les frappes réussies ;
- pour le tennis : les coups joués (revers, coup droit, smash) et les différents points marqués (ace, montées au filet).

Pour les vidéos de baseball, Lien et al. [Lien 2007] ont détecté quatre événements prédéterminés. Après avoir segmenté une vidéo en plans, l'approche extrait ensuite une image clé pour chacun des plans. En utilisant l'estimation du mouvement global, la taille, la position des objets calculées sur les images clés, l'approche classifie un

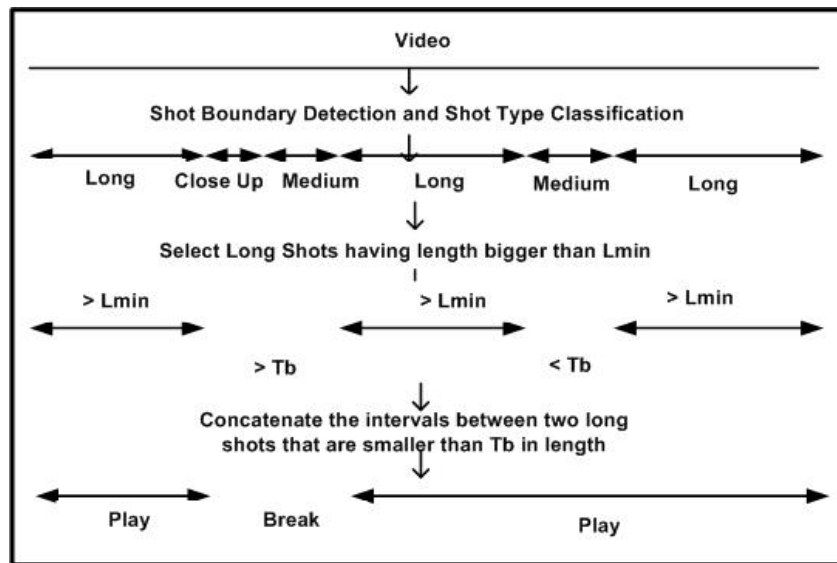


FIG. 2.11 – Détection de *break* et *play* dans une vidéo ([Ekin 2003b]).

plan en une des 8 classes. Un modèle de Markov caché de 4 états est créé pour reconnaître un des quatre événements prédéterminés.

Pour les vidéos de football, les auteurs de [Ekin 2003a] ont combiné les types de plans et les descripteurs visuels pour pouvoir détecter (1) les buts, (2) la présence d'arbitre et (3) la surface de réparation. Les buts sont reconnus en se basant sur les hypothèses :

- La durée du *break* après le but est entre 30 et 120 s ;
- Il existe un plan rapproché ou vue du public après le but ;
- Il existe au moins un plan ralenti ;
- Le plan ralenti suit le plan rapproché.

La présence d'arbitre est reconnue en utilisant la couleur dans les plans moyens et rapprochés. Ceci car l'arbitre s'habille avec des vêtements différents de ceux des joueurs et apparaît habituellement dans les plans moyens et rapprochés. Les auteurs ne cherchent pas à retrouver l'arbitre dans les plans généraux. Pour cela, les auteurs ont calculé les projections horizontale et verticale des couleurs dominantes des pixels. Les valeurs maximales sont détectées sur ces projections. La région de l'arbitre contient des pixels entourant le pixel ayant la valeur maximale. La figure 2.12 montre un exemple de détection de la présence d'arbitre dans un frame. La présence de l'arbitre est décidée par les critères sur le rapport entre la taille de la région et la taille du frame, le rapport entre le hauteur et la longueur de la région.

La surface de réparation est reconnue par la détection des trois lignes parallèles en appliquant le laplacien et la transformée de Hough. Un exemple de détection de la surface de réparation est montrée dans la figure 2.13.

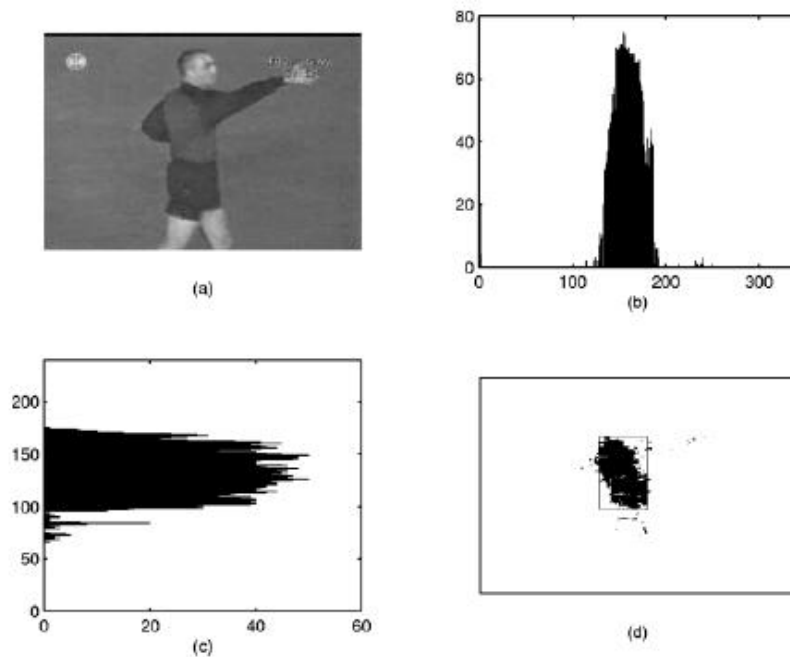


FIG. 2.12 – Détection de la présence d'un arbitre : (a) l'arbitre dans un frame ; (b) la projection horizontale des couleurs dominantes des pixels ; (c) la projection verticale des couleurs dominantes des pixels ; (d) la région déterminée pour l'arbitre ([Ekin 2003a]).

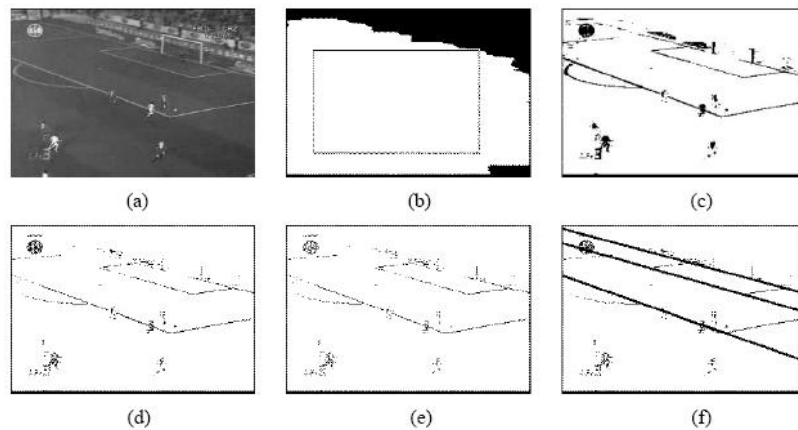


FIG. 2.13 – Détection de la surface de réparation : (a) le frame étudié ; (b) le masque du champ ; (c) les champs avec herbes ou sans herbe ; (d) le frame après avoir appliqué le Laplacien ; (e) le frame après le traitement ; (f) les trois lignes parallèles détectées ([Ekin 2003a]).

Assfalg et al. [Assfalg 2003] ont présenté un algorithme d'annotation automatique des vidéos de football en identifiant des *highlight*. Chacun des *highlight* est représenté par une machine à états finis en utilisant la connaissance du domaine. La figure 2.14 montre la machine à états finis correspondant à un but gagné.

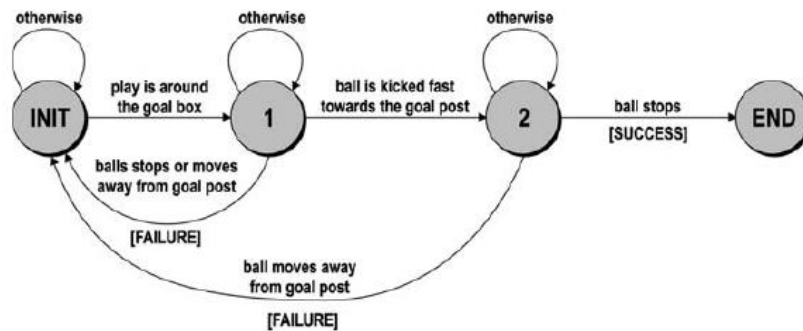


FIG. 2.14 – Machine à états finis correspondant à un but gagné est construite en se basant sur la connaissance du domaine ([Assfalg 2003]).

En extrayant les descripteurs visuels concernant le mouvement du ballon (p. ex. direction, vitesse, localisation), zone de jeu, position de joueurs, une vidéo est annotée par une liste des *highlight*.

2.2.3.4 L'indexation et la recherche d'enregistrements de réunion

Pour les enregistrements de réunion, l'objectif d'analyse de ces enregistrements est d'avoir une représentation compacte des enregistrements de réunion pour qu'une personne qui n'a pas participé à la réunion puisse savoir rapidement ce qui s'est passé dans la réunion et regarder dans les enregistrements les parties auxquelles elle s'intéresse. La représentation peut être faite par l'utilisation d'informations visuelle et auditive. Une approche est proposée par Cutler et al. dans [Cutler 2002]. Selon les analyses des auteurs, la personne qui n'a pas participé à la réunion s'intéresse à ce dont une personne spécifique a parlé (p. ex. le chef de l'équipe) ou au moment où il y a des choses importantes écrites sur le tableau. À partir de ces analyses, l'approche proposée consiste à décomposer les enregistrements en segments correspondant aux locuteurs et à détecter l'utilisation du tableau dans la salle de réunion. Pour cela, l'approche détecte les marqueurs auditifs (l'identification des locuteurs) et visuels (la détection de l'utilisation du tableau).

Les enregistrements de la réunion sont représentés par un ensemble des *highlight candidate* qui sont les segments correspondants aux locuteurs et les événements d'utilisation du tableau. L'utilisateur peut naviguer parmi les enregistrements en identifiant le locuteur ou l'événement d'intérêt.

2.2.4 Discussions

En indexation et recherche de vidéos scénarisées, les approches se focalisent sur la segmentation de vidéos, sur la détection des images clés et sur la détection de concepts. D'autres aspects tels que l'analyse du style de scénario des journaux télévisés [Haidar 2005] ou la reconnaissance de visages [Le 2006] sont également étudiés. Nous analysons à la fin de ce chapitre la relation entre l'indexation et la recherche de vidéos scénarisées et notre travail ce qui est dédié à l'indexation et la recherche de vidéos pour la vidéosurveillance.

Pour l'indexation et la recherche de vidéos non scénarisées, nous présentons brièvement les travaux dédiés à l'indexation et à la recherche de vidéos de sport et celles de réunion. La plupart de ces travaux consistent à détecter des *plays* et *breaks* et à reconnaître des événements prédéterminés. La recherche de plans de vidéo est simple. Il s'agit de retrouver des plans correspondant à un événement recherché. La mise en correspondance et le retour de pertinence ne sont pas considérés.

Dans notre travail de thèse, nous nous intéressons à analyser et à retrouver les *plays*, *Marqueurs visuels*, *Highlight candidates* et *Highlight groups* dans des vidéos de vidéosurveillance. La détection d'objets et la reconnaissance d'événements nous permettent d'enlever les *breaks*. Les *Marqueurs visuels*, *Highlight candidates* et *Highlight groups* sont représentés par deux concepts abstraits, les objets et les événements, qui seront présentés dans le chapitre 4.

2.3 Indexation et recherche de vidéos pour la vidéosurveillance

2.3.1 Introduction

Nous étudions dans cette section les approches dédiées à l'indexation et à la recherche de vidéos pour la vidéosurveillance. Toutes les approches se basent plus ou moins sur un module d'analyse. Ce module est cependant introduit comme s'il était un composant interne des approches. Afin de bien les analyser, nous séparons le module d'analyse vidéo de l'approche d'indexation et de recherche (voir figure 1.1 du chapitre 1) La figure 2.15 se focalise sur le module d'analyse vidéo. Ce module est constitué de trois grandes tâches : la détection d'objets, le suivi d'objets, et la reconnaissance des événements. Le but de la détection d'objets est d'identifier la présence de l'objet dans le frame actuel (on l'appelle une instance). Cette tâche classe également les objets détectés en classes prédéterminées. Le but de la tâche de suivi d'objets est de lier temporellement les instances détectées dans la tâche de détection d'objets. Le but de la tâche de reconnaissance des événements est d'identifier des événements prédéterminés. Les approches d'analyse vidéo pour la vidéosurveillance se différencient par les algorithmes proposés pour chacune des tâches.

Nous classons les approches en deux catégories selon la présence du retour de pertinence qui sont présentées dans les sections 2.3.2 et 2.3.3.

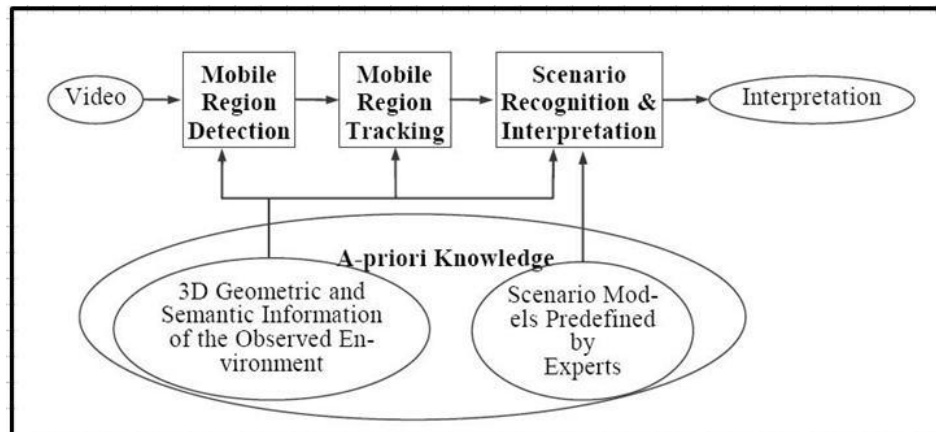


FIG. 2.15 – Architecture générale du module d’analyse vidéo pour la vidéosurveillance proposée par le projet PULSAR. Ce module est constitué de 4 tâches : la détection d’objets, la classification d’objets, le suivi d’objets et la reconnaissance d’événements. Les connaissances a priori comprenant l’information de contexte et la connaissances du domaine peuvent être utilisées ([Avanzi 2005]).

2.3.2 Approches sans retour de pertinence

Les résultats des modules d’analyse peuvent être les objets et les événements. Les approches d’indexation et de recherche de vidéos pour la vidéosurveillance basées sur la reconnaissance peuvent être effectuées (1) au niveau objets, (2) au niveau événements, (3) et au niveau composé.

2.3.2.1 Approches au niveau objets

Dans le cadre de l’indexation et de la recherche de vidéos pour la vidéosurveillance au niveau des objets, l’objectif est de retrouver les objets observés par une ou un ensemble de caméras qui sont semblables à une image ou un objet recherché. Nous divisons les approches proposées en trois catégories : celles basée sur la fusion précoce, celles sur la fusion tardive et les approches hybrides. Nous présentons le cas général où une scène est observée par plusieurs caméras. Dans le cas où il y a une seule caméra, la recherche des objets est celle de la première catégorie.

Les approches de la première catégorie fusionnent des vidéos provenant des caméras dans la tâche de détection et de suivi des objets. La figure 2.16 montre le processus commun des approches dans cette catégorie.

Le travail de Conaire et al. [Conaire 2006] fait partie de cette catégorie. Les auteurs ont combiné l’information d’une caméra infrarouge et d’une caméra CCTV (Closed Circuit Television). Ils ont rapporté que la combinaison de deux caméras permet non seulement d’observer des objets dans des conditions différentes d’illumination mais aussi d’obtenir de bons résultats de détection et de suivi d’objets. Afin de mettre en correspondance entre les objets, les descripteurs tels que la taille de

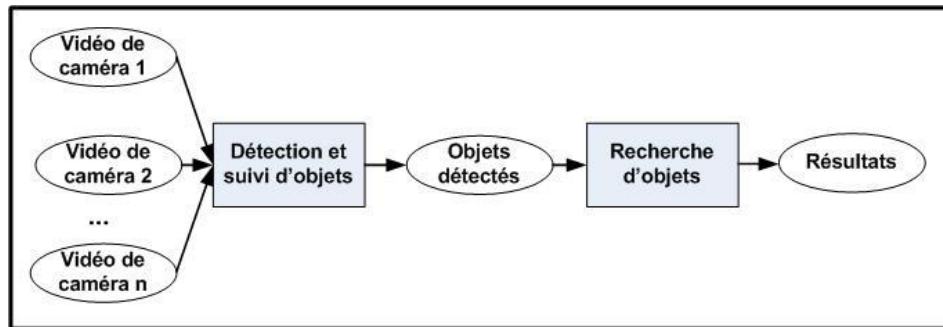


FIG. 2.16 – Indexation et la recherche de vidéos pour la vidéosurveillance au niveau objets avec la fusion précoce. Les données provenant de différentes caméras sont fusionnées dans la détection et le suivi d'objets. L'indexation et la recherche s'effectuent sur les données fusionnées.

l'objet sont calculés.

Dans [Yuk 2007], les auteurs ont proposé une approche d'indexation et de recherche des objets basée sur les descripteurs de MPEG-7. Après avoir détecté et suivi les objets en appliquant la segmentation du mouvement et le filtre de Kalman, l'approche calcule les couleurs dominantes et les histogrammes des contours des objets sur les frames où les objets sont détectés. Les couleurs dominantes moyennes et les histogrammes moyens des contours sont calculés à partir des couleurs dominantes et des histogrammes des contours extraits. Dans la phase de recherche, une requête est soit une image d'exemple soit une esquisse dessinée par l'utilisateur. La distance entre la requête et les objets indexés est définie par l'équation 2.39.

$$D = \alpha_{dc}D_{dc} + \alpha_{ed}D_{ed} \quad (2.38)$$

où α_{dc} , α_{ed} sont des seuils prédéfinis, D_{dc} et D_{ed} sont respectivement les distances basées sur les couleurs dominantes et sur les histogrammes de contours. La similarité entre la requête et les objets indexés est calculée par la formule suivante :

$$R = \begin{cases} (1 - \frac{D}{D_{max}}) \times 100\% & \text{si } D \leq D_{max} \\ 0\% & \text{sinon} \end{cases} \quad (2.39)$$

où D_{max} est la valeur maximale possible de distance.

Quelques exemples de résultats de recherche obtenus sur 19 vidéos contenant 1421 objets détectés sont présentés. Néanmoins, aucune évaluation quantitative est montrée. Les deux approches utilisent les valeurs moyennes des descripteurs extraits. Cela n'est pas toujours approprié parce qu'il accumule les erreurs produites par la détection et le suivi d'objets à chaque frame.

L'approche de Meessen et al. [Meessen 2006] se base sur une hypothèse : le module d'analyse ne comprend que la détection d'objets. L'approche de Meessen et al. consiste d'une part à détecter des keyframes et d'autre part à retrouver des

keyframes similaires à un frame recherché. Un frame F_i est défini par :

$$F_i = [G_i, M_i] \quad (2.40)$$

où M_i est l'ensemble des ni objets mobiles dans le frame F_i , G_i sont des descripteurs globaux du frame (p. ex. le rapport de nombre de pixels mobiles sur le nombre de pixels d'un frame). Un objet mobile O_i est représenté par des descripteurs au niveau numérique (p. ex. les histogrammes des couleurs) et des descripteurs au niveau sémantique (p. ex. les mots clés).

$$O_i = [L_i, H_i] \quad (2.41)$$

Étant donné un frame recherché Q , Q est également représenté par : $Q = [G_q, M_q]$. La distance entre deux frames est constituée de deux parties : l'une est la distance entre les descripteurs globaux et l'autre est la distance entre les deux ensembles d'objets détectés dans ces deux frames. Elle est définie par :

$$D(F_i, Q) = W_g * D_g(G_i, G_q) + W_m * D_m(M_i, M_q) \quad (2.42)$$

où W_g et W_m sont des participations de distance $D_g(G_i, G_q)$ et $D_m(M_i, M_q)$. La distance $D_g(G_i, G_q)$ est déterminée par :

$$D_g(G_i, G_q) = \sqrt{\sum_{k=1}^{nk} \frac{w_{gk}}{\sigma_{gk}^2} (G_{i,k} - G_{q,k})^2} \quad (2.43)$$

Afin de déterminer la distance $D_m(M_i, M_q)$, les auteurs ont défini la distance entre deux objets par leurs descripteurs :

$$D_o(O_i, O_q) = W_l * D_l(L_i, L_q) + W_h * D_h(H_i, H_q) \quad (2.44)$$

La distance $D_m(M_i, M_q)$ entre deux ensemble d'objets est calculée comme suit :

- l'étape 1 : choisir une paire d'objets dont la distance D_o est la plus faible (un objet de l'ensemble M_i et l'autre de l'ensemble M_q) ;
- l'étape 2 : enlever les objets déterminés dans l'étape 1 dans les deux ensembles M_i et M_q ;
- l'étape 3 : continuer les étapes 1 et 2 jusqu'à un de deux ensembles n'a plus d'objets.

La distance $D_m(M_i, M_q)$ est la somme des distances des paires d'objets déterminées à l'étape 1. Cette distance n'est pas pertinente si la détection d'objets n'est pas bonne car dans cette distance un objet est apparié à un seul objet.

Pour la détection des keyframes, un frame est considéré comme un keyframe si la distance ($D_m(M_i, M_q)$) entre ce frame et les frames voisins est supérieure à un seuil. Une vidéo est en effet représentée par un ensemble de keyframes détectés.

Cette approche est appropriée dans le cas où l'utilisateur s'intéresse à retrouver des frames dans une vidéo similaires à un frame recherché. L'analyse vidéo est très réduite dans ce travail. Elle ne comprend que la détection d'objets. Cependant, l'approche a trois inconvénients :

- Un objet mobile est représenté par un seul blob. La trajectoire et les relations temporelles des objets ne sont pas considérées ;
- L'approche ne convient pas pour les applications dans lesquelles l'interaction entre des objets est informative ;
- L'approche perd l'information de l'évolution des objets dans le temps qui est important pour définir les événements.

Contrairement aux approches de fusion précoce, celles de fusion tardive font la détection et le suivi d'objets séparément pour chacune des caméras. L'appariement des objets détectés et de la requête se fait tout d'abord indépendamment pour chacune de caméras. Une fusion est alors effectuée sur les résultats obtenus pour avoir les résultats finaux. La figure 2.17 illustre le processus commun des approches dans cette catégorie.

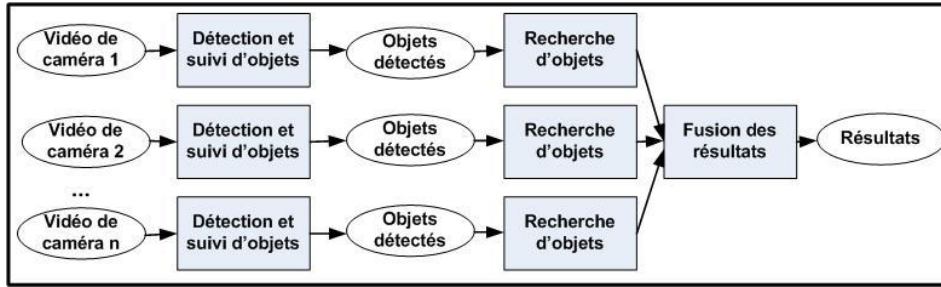


FIG. 2.17 – Indexation et la recherche de vidéos pour la vidéosurveillance au niveau objets avec la fusion tardive. L'indexation et la recherche s'effectuent sur la donnée de chaque caméra. Les résultats de recherche sont fusionnés.

Le travail de Ma et al. [Ma 2007], [Cohen 2008] appartient à cette catégorie. Dans ce travail, la détection des régions mobiles est effectuée sur les vidéos provenant de plusieurs caméras. Une région détectée est représentée par un modèle d'apparence. Soit f un vecteur de descripteurs :

$$f(x, y) = [x, y, R(x, y), G(x, y), B(x, y), \nabla R^T(x, y), \nabla G^T(x, y), \nabla B^T(x, y)] \quad (2.45)$$

où R , G , et B est les valeurs de couleurs du pixel (x, y) , $\nabla R^T(x, y)$, $\nabla G^T(x, y)$ et $\nabla B^T(x, y)$ sont les valeurs du gradient pour chaque composante de couleur. Le modèle d'apparence C_k de la région détectée B_k est défini par l'équation 2.46.

$$C_k = \sum_{x,y} (f - \bar{f})(f - \bar{f})^T \quad (2.46)$$

La distance entre deux modèles d'apparence est définie par :

$$d(C_i, C_j) = \sqrt{\sum_{k=1}^d \ln^2 \alpha_k(C_i, C_j)} \quad (2.47)$$

où $\alpha_k(C_i, C_j)$ sont les valeurs propres généralisées de C_i et C_j qui sont déterminées par :

$$\alpha_k C_i x_k - C_j x_k = 0 \quad k = 1, \dots, d \quad (2.48)$$

La requête contenant un seul objet d'intérêt est également représentée par un modèle d'apparence. Le processus de recherche consiste en :

- déterminer la caméra contenant un modèle qui est le plus semblable à celui de la requête en se basant sur la distance dans l'équation 2.47 ;
- appairer la requête avec tous les modèles de cette caméra par les K plus proches voisins ;
- les objets retrouvés dans l'étape précédente sont utilisés pour faire l'appariement de K plus proches voisins avec tous les modèles des caméras restantes, les résultats sont ensuite fusionnés.

De plus, les auteurs ont proposé une méthode de représentation d'un ensemble de régions détectées correspondantes à un objet. Soit $S^{(k)}$ l'ensemble de régions détectées de l'objet k dans n frames : $S^{(k)} = \{B_i, i = 1, \dots, n\}$. L'ensemble $S^{(k)}$ peut être représenté par : $S^{(k)} = \{C_i^{(k)}, i = 1, \dots, n\}$. La méthode de représentation permet de transformer $S^{(k)} = \{C_i^{(k)}, i = 1, \dots, n\}$ à $S^{r(k)} = \{C_j^{r(k)}, j = 1, \dots, m\}$ où $m \ll n$. Cette méthode peut être résumée par :

- appliquer le regroupement agglomératif (agglomerative clustering) aux n modèles d'apparence $C_i^{(k)}, i = 1, \dots, n$;
- enlever les groupes ayant peu d'éléments ;
- déterminer le modèle d'apparence représentatif $C_j^{r(k)}, j = 1, \dots, m$ pour chacun des groupes.

Soit $S^{(p)}$ et $S^{(q)}$ deux ensembles de modèles d'apparence de deux objets. La distance de Hausdorff de $S^{(p)}$ et $S^{(q)}$ est définie par l'équation 2.49.

$$d(S^{(p)}, S^{(q)}) = \max_{C_i^{r(q)} \in S^{r(q)}} \min_{C_j^{r(p)} \in S^{r(p)}} (d(C_i^{r(q)}, C_j^{r(p)})) \quad (2.49)$$

Cette approche permet de trouver les observations prises par les caméras qui se chevauchent ou pas. Cependant, aucune évaluation n'a été donnée. L'algorithme de détection des blobs représentatifs proposé par Ma et al. arrive à corriger les erreurs produites par la détection et le suivi d'objets si les erreurs se présentent dans un petit nombre de blobs par rapport au nombre total des blobs d'un objet car il enlève les groupes qui ont peu d'éléments. De plus, la mise en correspondance des objets basée sur la distance de Hausdorff (voir équation 2.49) n'est pas appropriée. Car la distance de Hausdorff n'est pas robuste au bruit. Si l'on a deux ensembles de points A et B qui sont parfaitement appariés sauf un seul point de A qui est semblable à aucun point de B, la distance de Hausdorff sera déterminée par ce point. Ce problème est fréquemment rencontré dans l'indexation et la recherche de vidéos pour la vidéosurveillance dû aux résultats imparfaits de la détection et du suivi d'objets.

Dans les approches hybride, la recherche des objets se fait sur les régions détectées avant la fusion et aussi après la fusion. L'approche proposée dans [Calderara 2006] est une approche hybrides. Une région appelée PA (person's appearance) est détectée pour un objet dans chacun des frames. Ensuite un ensemble de régions détectées appelé SCAT (single camera appearance trace) d'un objet dans un intervalle de temps est créé. Puis un MCAT (multicamera appearance trace) est construit pour chacun des objets en appliquant l'algorithme qui permet de relier les SCATs du même objet observé par différentes caméras. Cet algorithme est appelé *consistent labeling* (cf. figure 2.18). Le processus de l'appariement des objets est constitué de deux étapes (cf. figure 2.19) :

- Étape 1 appelée *Best PA selection* : un PA d'un objet est choisi comme une requête, dans MCAT de cet objet, un SCAT dont la taille d'objet est la plus grande est identifié. Le PA dans le SCAT identifié dont la variation de couleur est la plus grande est déterminé comme une requête intermédiaire ;
- Étape 2 appelée *Similarity-based retrieval* : la requête intermédiaire est comparée avec les MCATs indexés en se basant sur la densité de probabilité de couleur.

Nous détaillons la deuxième étape. Pour un MCAT dans la base d'indexation :

- pour le premier PA de MCAT, un histogramme de couleur R, G, B est calculé, dix pixels X_t dont leurs couleurs correspondent à dix valeurs maximales de l'histogramme sont choisis. Dix gaussiennes sont initialisées par les dix pixels, leurs poids sont également initiés par la même valeur.
- pour un PA suivant de MCAT, dix pixels dont leurs couleurs correspondent à dix valeurs maximales de l'histogramme sont déterminés. Pour chacun des pixels X_t , l'approche vérifie :
 - si toutes les dix gaussiennes ayant la différence de leurs moyennes et les pixels X_t est supérieur à $2.5 * \sigma$, une nouvelle gaussienne va être créée. L'approche remplace la gaussienne dont le poids est le plus petit par la nouvelle gaussienne ;
 - sinon l'approche met à jour les moyennes et les écart types des gaussiennes selon l'équation 2.50 et leurs poids.

$$\begin{aligned}\mu_t &= (1 - \alpha) * \mu_{t-1} + \alpha * X_t \\ \sigma_t^2 &= (1 - \alpha) * \sigma_{t-1}^2 + \alpha * (X - \mu_t)^T * (X - \mu_t)\end{aligned}\quad (2.50)$$

Un MCAT est en effet représenté par dix gaussiennes pondérées. Afin de mettre en correspondance la requête et un MCAT dans la base d'indexation, les n gaussiennes parmi dix gaussiennes du MCAT sont déterminées par l'équation 2.51.

$$\sum_{i=1}^n w_i \leq T \leq \sum_{i=1}^{n+1} w_i \quad (2.51)$$

où T est un seuil prédéterminé. Une fois une requête donnée, dix pixels X_l dont leurs valeurs correspondent à dix valeurs maximales de l'histogramme sont déterminés.

La similarité entre la requête et le MCAT dans la base d'indexation est :

$$S = \sum_{l=1}^{10} \sum_{i=1}^n b_{il} w_i \quad (2.52)$$

où b_{il} est déterminé par l'équation 2.53.

$$b_{il} = \begin{cases} 1 & \text{si } |\mu_i - X_l| \leq 2.5 * \sigma_i \\ 0 & \text{sinon} \end{cases} \quad (2.53)$$

Nous remarquons quatre inconvénients de cette approche :

- les caméras doivent se chevaucher ;
- l'approche demande de relier les SCATs du même objet pour créer son MCAT ;
- l'utilisateur ne choisit pas toujours un PA du MCAT pour formuler la requête. La façon de formuler une requête doit être plus flexible. L'utilisateur peut identifier une région contenant l'objet d'intérêt à partir d'un frame ;
- l'approche est efficace si la détection et le suivi d'objets ont les résultats fiables dans la plupart des temps. Sans quoi les anciennes gaussiennes sont toujours remplacées par les nouvelles gaussiennes.

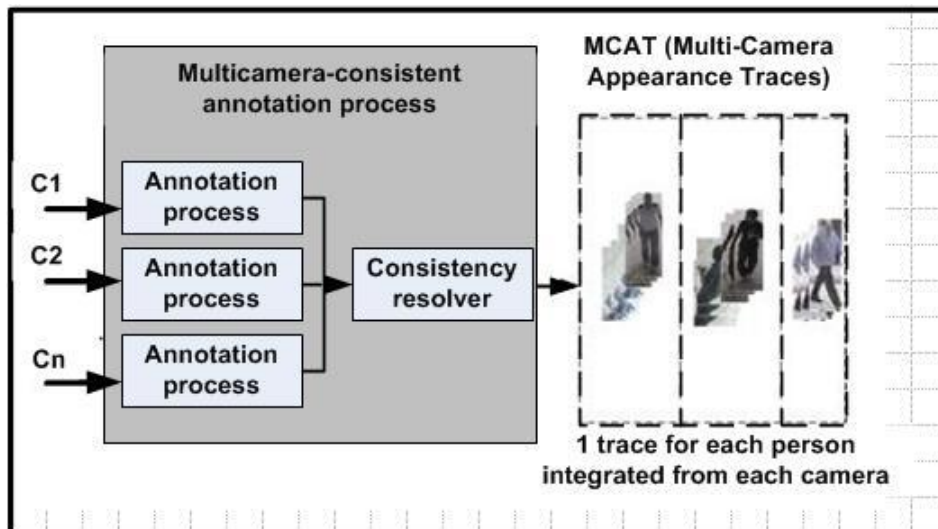


FIG. 2.18 – Processus de génération PA, SCAT et MCAT pour chacun des objets observés par un ensemble de caméras ([Calderara 2006]).

2.3.2.2 Approches au niveau événements

Dans le cadre de l'indexation et de la recherche de vidéos pour la vidéosurveillance au niveau événements. Au lieu de reconnaître tous les événements d'intérêt de l'utilisateur ce qui est dans la plupart des cas impossible, l'approche de Ghannem

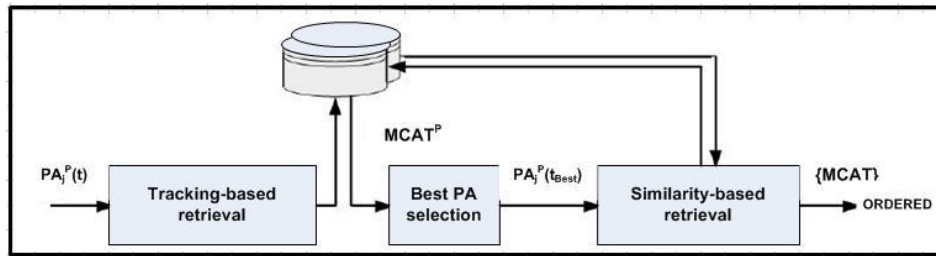


FIG. 2.19 – Appariement des objets est constitué de deux étapes : la première étape (Best PA selection) permettant de déterminer le PA d'un objet à travers de PAs créés à partir des vidéos provenant de toutes les caméras, la deuxième étape consistant de comparer le PA choisi avec les MCATs de tous les objets détectés ([Calderara 2006]).

et al. [Ghanem 2004] permet aux utilisateurs de définir eux-mêmes des événements d'intérêt à partir d'événements primitifs reconnus à l'aide des relations logiques et temporelles. Cette approche est illustrée dans la figure 2.20. Un module de vision consiste à suivre les objets et à détecter les événements primitifs. Les réseaux de Petri ont été choisis dans ce travail. Les auteurs ont proposé trois extensions des réseaux de Petri :

- des transitions conditionnelles qui sont associées à une condition ;
- des transitions hiérarchiques qui visent à représenter un réseau de Petri complexe par des réseaux de Petri simples ;
- des tokens en couleurs qui permet de contenir plusieurs objets impliqués dans un événement.

Les événements primitifs reconnus par les modules d'analyse vidéo sont représentés par des réseaux de Petri. Une requête qui définit un événement complexe à partir des événements primitifs, est également représentée par un réseau de Petri. Ce travail fournit une méthode de représentation des requêtes basée sur les réseaux de Petri. Quelques exemples de requêtes formulées par réseau de Petri sont également montrés. Cependant, aucun résultat quantitatif ou qualificatif n'est donné. Les auteurs n'ont pas expliqué comment faire la mise en correspondance entre la requête et les informations indexées. De plus, ce travail ne permet pas d'associer aux requêtes des images d'exemple ce qui est très souhaitable dans les systèmes d'indexation et de recherche. La recherche des objets par leurs apparences et leurs trajectoires n'est pas abordée dans ce travail.

2.3.2.3 Approches au niveau composé

Les approches combinant les objets et les événements font l'objet d'un travail considérable. Une des approches présentée par Hu et al. [Hu 2007] est dédiée aux enregistrements de routes. Les auteurs ont défini une activité et un modèle d'activité. Un modèle d'activité représente un ensemble d'activités similaires. Les descripteurs de l'activité et ceux du modèle d'activité sont présentés dans les tableaux 2.2 et 2.3.

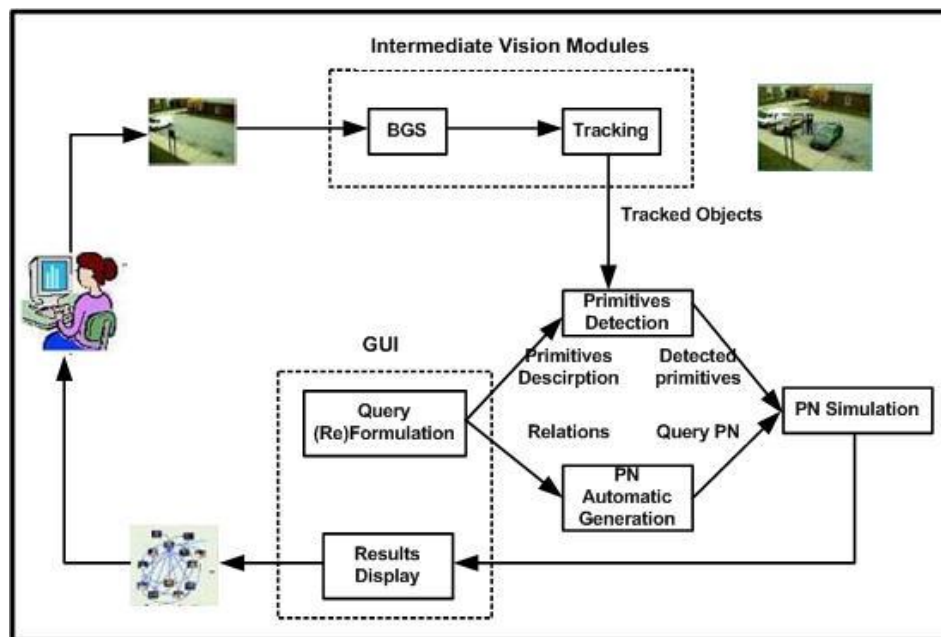


FIG. 2.20 – Objets sont détectés et suivis par un module vision (*Intermediate Vision Modules*). Les événements primitifs reconnus par *Primitives Detection* sont représentés par des réseaux de Petri. Les requêtes qui définissent un événement complexe à partir des événements primitifs, sont également représentées par un réseau de Petri ([Ghanem 2004]).

TAB. 2.2 – 7 descripteurs d'une activité [Hu 2007].

Composant	Valeur
ACT_ID	l'étiquette de l'activité
VIDEO_ID	l'étiquette de la vidéo
Birth_Time	le frame de début
Death_Time	le frame du fin
Trajectory	$T_{ST} = (f_1, f_2, \dots, f_n), f_i = (x_i, v_{x_i}, y_i, v_{y_i})$
Obj_Color	les 3 composants de couleur R, G, B
Obj_Size	la hauteur et la longueur

TAB. 2.3 – 4 descripteurs d'un modèle d'activité [Hu 2007].

Composant	Valeur
AM_ID	l'étiquette du modèle d'activité
ACT_List	la liste d'activités
Trajectory	$T_{ST} = (f_1, f_2, \dots, f_n), f_i = (x_i, v_{x_i}, y_i, v_{y_i})$
Conceptual_Descriptions	la liste de mots clés {turn left, low speed, ...}

Afin de déterminer un modèle d'activités, les trajectoires des activités sont regroupées. Pour chaque groupe, un modèle d'activité est créé. Le descripteur *Trajectory* du modèle est la trajectoire moyenne des activités appartenant à ce modèle. Les auteurs ont annoté les modèles d'activités par des descriptions textuelles tels que tourner à gauche (*turn left*). Ces mots clés sont stockés dans le descripteur *Conceptual_Descriptions*. Dans le travail de Hu et al. [Hu 2007], les auteurs ont appliqué la classification hiérarchique composée d'une classification spatiale sur la position et d'une classification temporelle sur la vitesse des objets. Il est à noter que le travail de Xie et al. [Xie 2004a] est similaire à celui de Hu et al. Cependant, au lieu d'utiliser une classification hiérarchique, Xie et al. ont employé un HSOM (Hierarchical Self-Organizing Map).

La requête est exprimée soit par les mots clés, soit par une esquisse dessinée par l'utilisateur. Si la requête est sous la forme des mots clés, les modèles d'activités contenant ces mots clés sont identifiés. Les activités sont retrouvées. Si la requête est une esquisse représentant la trajectoire recherchée. Tout d'abord, cette trajectoire est comparée avec la trajectoire du modèle d'activité. Ensuite, les trajectoires des activités appartenant aux modèles retrouvés sont comparées avec la trajectoire recherchée afin de raffiner des résultats.

Nous notons deux avantages de ce travail :

- L'annotation se fait sur les modèles d'activités, elle est donc moins coûteuse que celle sur les activités ;

- Les modèles d'activités sont construits d'une manière flexible. Lorsqu'une nouvelle vidéo avec les activités déterminées arrivent, l'approche vérifie s'il est nécessaire de créer de nouveaux modèles pour pouvoir la représenter.

Le travail n'utilise néanmoins que la trajectoire des objets.

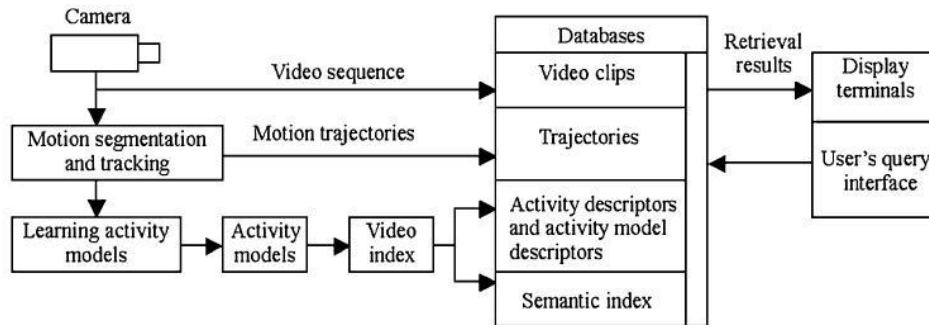


FIG. 2.21 – Approche proposée par Hu et al. : après avoir suivi d'objets, les trajectoires des objets et les modèles d'activités appris sont stockés dans la base de données ([Hu 2007]).

L'objectif de travail de Stringa et al. [Stringa 1998], [Stringa 2000] est de détecter l'événement d'abandon d'un objet et de retrouver les frames concernant cet événement. L'architecture de cette approche est présentée dans la figure 2.22. Une alarme est déclenchée dès qu'un événement est détecté. Un plan de vidéo doit être construit et affiché au personnel de sécurité afin de lui expliquer l'événement correspondant à cette alarme. Les auteurs ont présenté une méthode afin de déterminer ce plan. Le plan est constitué de 24 frames du $X-8$ à $X+16$ où X est le moment auquel l'alarme est déclenchée (voir figure 2.23). Les descripteurs extraits sur les objets abandonnés peuvent être utilisés pour retrouver des plans de vidéo contenant l'abandon d'un objet particulier. L'approche proposée se limite à retrouver l'événement d'abandon d'un objet.

Le travail dans [Foresti 2002] cherche à améliorer l'approche de Stringa et al. [Stringa 2000]. Les auteurs ont fait d'une manière parallèle la détection d'objets abandonnés, la détection et le suivi d'objets mobiles et la classification d'activités.

La détection et le suivi d'objets mobiles permettent de construire un graphe temporel. Dans ce graphe, un blob détecté dans le frame en cours est représenté par un noeud. Ce noeud sera lié aux noeuds du frame précédent et à ceux du frame suivant. Afin de représenter l'interaction entre des objets dans la scène, quelques noeuds peuvent être fusionnés à un seul noeud ou un noeud peut être divisé en quelques noeuds.

L'approche proposée construit des plans de vidéos concernant l'événement d'abandon d'un objet d'une manière plus flexible que celle de [Stringa 2000]. Il est à noter que dans le travail de Stringa et al. [Stringa 1998], [Stringa 2000] la taille et les frames de début et de fin du plan sont fixés. Cependant, ils sont variables dans le

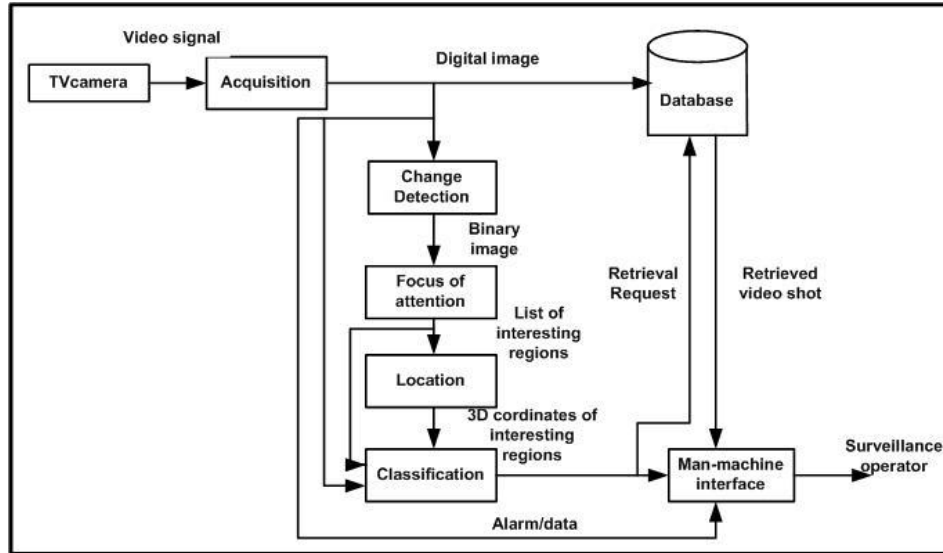


FIG. 2.22 – Approche proposée par Stringa et al. Une fois l'abandon d'un objet reconnu (une alarme est déclenchée), un plan de vidéo constitué de 24 frames est créé. L'approche se limite à retrouver des plans vidéo concernant l'événement d'abandon d'un objet ([Stringa 2000]).

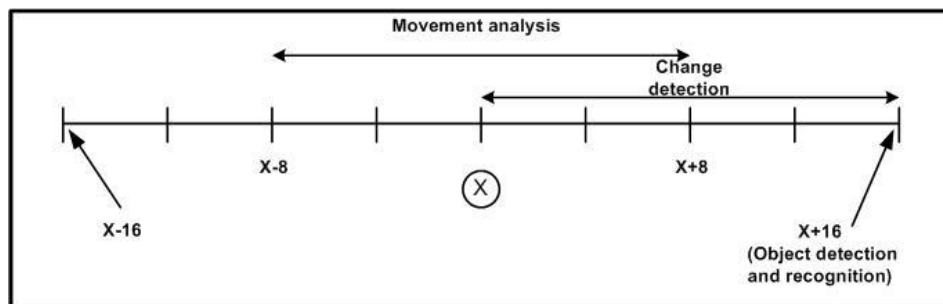


FIG. 2.23 – Construction d'un plan vidéo concernant l'événement d'abandon d'un objet. Le plan est constitué de 24 frames du $X-8$ à $X+16$ où X est le moment auquel l'alarme est déclenchée ([Stringa 1998]).

travail de Foresti et al. [Foresti 2002]. Les auteurs ont proposé une méthode permettant de déterminer le moment de début (t_{in}) et celui de fin (t_{fin}) du plan.

La détection de moment de fin est constituée de deux étapes. La première étape consiste à estimer le moment de fin. Soit $E_k^{AO}(t_i)$ la détection d'un objet abandonné au moment t_i , le moment de fin t_{fin} peut être estimé par $f_{fin} = t_i + K$. Cela correspond aux $n_t = K * f_t$ frames où f_t mesure le nombre de frames par seconde. Le frame de fin correspond au moment où un noeud est divisé en deux dans le graphe temporel. La deuxième étape vise à raffiner le moment de fin. Des blobs détectés dans la zone centrée par la position de l'objet abandonné (x_{AO}, y_{AO}) du frame n_t au frame correspondant au moment t_i sont vérifiés. Soit $Hi_{E_j^{OT}}(t_j)$ l'histogramme de couleurs du blob au moment t_j , $Hi_{E_i^{AO}}(t_i)$ l'histogramme de couleurs de l'objet abandonné. Pour chaque composant de couleur, 32 niveaux sont choisis. La distance $d(Hi_{E_k^{AO}}(t_i), Hi_{E_j^{OT}}(t_j))$ est définie par :

$$\frac{1}{32^3} \sum_{(I^R, I^G, I^B)} (Hi_{E_k^{AO}}(t_i)(I^R, I^G, I^B), Hi_{E_j^{OT}}(t_j)(I^R, I^G, I^B)) \quad (2.54)$$

t_{fin} est déterminé par t_j où la distance (cf. équation 2.54) est la plus petite.

Le moment de début t_{in} est déterminé en cherchant dans le graphe, le chemin simple connecté au noeud correspondant à l'objet abandonné. Les figures 2.24 et 2.25 montrent la détection des moment de fin et celui de début. Pour d'autres types d'événements, la détection des moments de fin et de début est similaire à celle de l'événement d'abandon d'un objet.

L'objectif du travail de Foresti et al. [Foresti 2002] est de préparer des plans correspondant aux événements d'intérêt (pour la phase d'indexation). Lors qu'une alarme est déclenchée, le plan correspondant est construit et affiché au personnel de sécurité afin de lui expliquer l'événement concernant cet alarme. Ce plan peut être stocké. Cependant, la phase de recherche n'est pas considérée.

Le travail présenté dans [Greenhill 2002] (cf. figure 2.26) consiste à (1) détecter et suivre des objets, (2) extraire des descripteurs, faire des annotations (3) et répondre aux requêtes des utilisateurs. L'approche classe les objets en 3 classes (personne, véhicule, autre). Une interface a été implémentée pour que l'utilisateur puisse poser des questions sur les caractéristiques des objets telles que la couleur, la vitesse. Les requêtes sous la forme SQL (Structured Query Language) sont également autorisées. Nous remarquons deux limitations de cette approche :

- La mise en correspondance entre des objets se base sur les méta-données ;
- Il est impossible d'associer aux requêtes exprimées par le langage de requête les images d'exemple.

Hampapur et al. [Hampapur 2007], [Tian 2008] ont présenté un système pour la vidéosurveillance. Le module d'analyse de ce système vise à détecter les objets, les suivre, les classifier et à reconnaître les événements prédéfinis. Une fois l'analyse de vidéos faite, les résultats sont stockés comme les méta-données. Neuf types de requête sont possibles dans ce système :

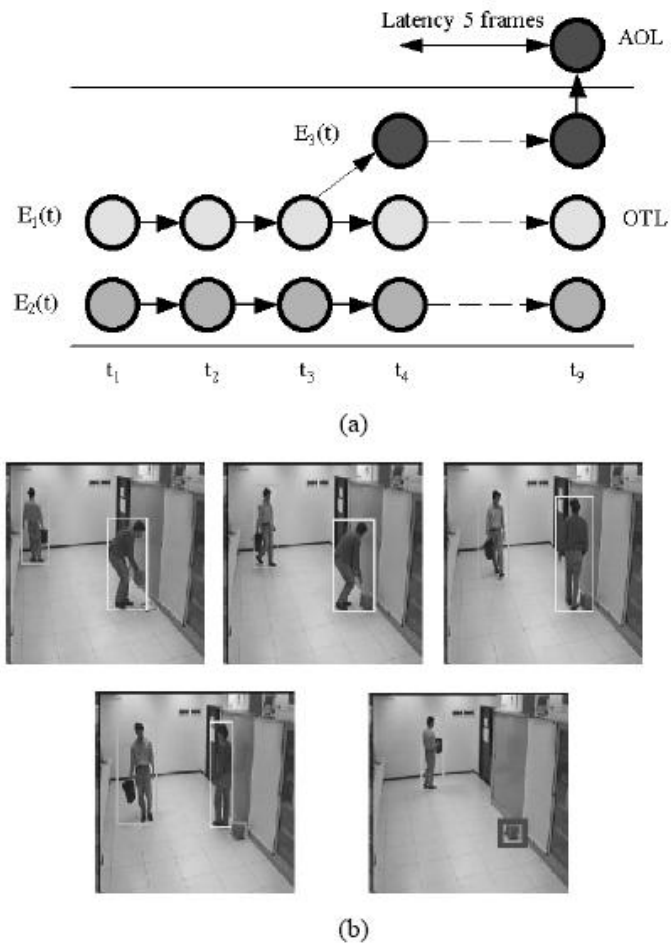


FIG. 2.24 – Exemple de la détection du moment de fin t_{fin} de l'événement d'abandon d'un objet : (a) le graphe temporel correspondant aux objets détectés et suivis (Object Tracking Layer - OTL) et à l'objet abandonnée (Abandoned Object Layer-AOL). Le frame de fin correspond au moment où un noeud est divisé en deux dans le graphe temporel ; (b) les frames correspondant à ce graphe temporel ([Foresti 2002]).

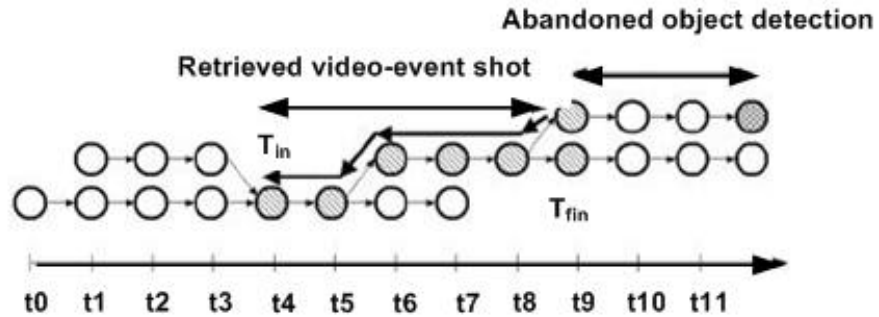


FIG. 2.25 – Exemple de la détection du moment de début t_{in} de l'événement d'abandon d'un objet. Le moment de début t_{in} est déterminé par le chemin simple connecté au noeud correspondant à l'objet abandonné dans le graphe ([Foresti 2002]).

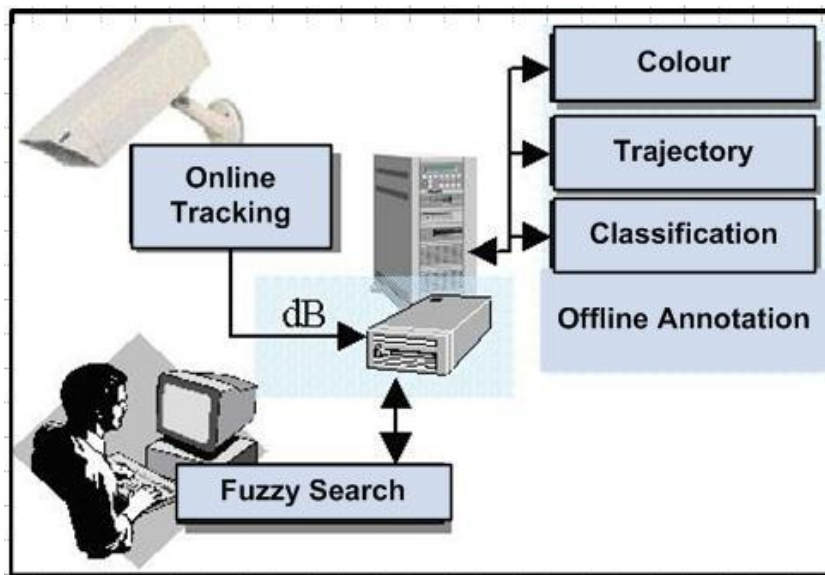


FIG. 2.26 – Approche de Greenhill et al. consiste à (1) détecter et suivre les objets, (2) extraire les descripteurs et faire les annotations (3) et répondre aux requêtes des utilisateurs ([Greenhill 2002]).

- par le temps (p. ex. trouver les événements qui ont eu lieu de 3h à 4h le 10 avril 2007) ;
- par la présence de l'objet (p. ex. trouver les cent derniers événements dans un système *live*) ;
- par la taille de l'objet (p. ex. trouver les événements produits par les objets dont les tailles maximales sont [10, 100] pixels) ;
- par le type de l'objet (p. ex. trouver les objets appartenant à la classe 'Personne') ;
- par la vitesse de l'objet (p. ex. trouver les objets dont les vitesses sont [10, 100]) ;
- par la couleur de l'objet (p. ex. trouver les objets dont la couleur est rouge) ;
- par la localisation de l'objet (p. ex. trouver les objets appartenant à la boîte déterminée par deux points [10, 100] et [50, 200]) ;
- par la durée de l'activité (p. ex. trouver les événements dont les durées sont [10, 100] frames) ;
- par la combinaison de ces types.

Bien que ce travail soit assez complet, il prend un seul blob pour chacun des objets afin de les indexer. Un seul descripteur d'apparence (les couleurs dominantes) est utilisé. Un objet est représenté par une parmi huit couleurs. Les indexations sous la forme de méta-données ne permettent pas de corriger l'imperfection des modules d'analyse vidéo.

2.3.3 Approches basées sur le retour de pertinence

Le retour de pertinence est une technique qui permet le système d'apprendre les retours des utilisateurs et de s'adapter pour mieux répondre aux requêtes des utilisateurs [Rui 2001]. Jusqu'à présent, il existe deux approches permettant de faire le retour de pertinence pour l'indexation et la recherche de vidéos pour la vidéosurveillance : l'une de Meessen et al. [Meessen 2007] et l'autre de Chen et al. [Chen 2006], [Chen 2007].

En se basant sur l'hypothèse : le module d'analyse ne contient que la détection d'objets, Meessen et al. [Meessen 2007] ont proposé une méthode de recherche interactive des frames dans les vidéos de vidéosurveillance. Cette méthode combine l'apprentissage de plusieurs instances (multiple instance learning - MIL), les machines à vecteurs de support (SVM) et l'apprentissage actif (active learning).

Le problème de l'apprentissage de plusieurs instances consiste à classifier un sac (bag) ou un élément (instance) en se basant sur les sacs annotés dans la base d'apprentissage. Dans le cas binaire, un sac est positif s'il y a au moins un élément positif tandis qu'un sac est négatif si tous de ses éléments sont négatifs. Dans l'apprentissage de plusieurs instances, pour les sacs dans la base d'apprentissage, l'information sur la classe du sac est déterminée. Cependant, l'information sur la classe de chaque élément n'est pas disponible.

En appliquant l'apprentissage de plusieurs instances à l'indexation et à la recherche des frames dans les vidéos de vidéosurveillance, un frame correspond à un sac tandis qu'un blob correspond à un élément.

Etant donné un sac B_i , et un élément x^k , la similarité $s(B_i, x^k)$ entre le sac B_i et l'élément x^k est déterminée par :

$$s(B_i, x_k) = \max_j \exp\left(-\frac{d(x^k, x_{ij})}{\sigma^2}\right) \quad (2.55)$$

La distance $d(x^k, x_{ij})$ est définie par :

$$d(x^k, x_{ij}) = \sqrt{\sum_{v=1}^V u_v(x_{ij,v} - x_v^k)^2} \quad (2.56)$$

où V est le nombre de descripteurs utilisés. Les auteurs ont utilisé la position de l'objet, sa hauteur, sa longueur et les histogrammes des couleurs RGB.

Soit l^+ l'ensemble de sacs positifs et l^- l'ensemble de sacs négatifs fournis par l'utilisateur, le nombre total d'éléments x des deux ensembles est n . Chaque sac B_i dans l'ensemble d'apprentissage est représenté par un vecteur comme suit :

$$m(B_i) = [s(B_i, x^1), s(B_i, x^2), \dots, s(B_i, x^n)] \quad (2.57)$$

Les SVM sont entraînés sur l'espace de ces vecteurs.

Pour un nouveau sac B_j avec le vecteur $m(B_j)$ défini par l'équation 2.57, la classe du sac est déterminée par :

$$y = \text{sign}(w^T m(B_j) + b) \quad (2.58)$$

où w^T et b sont déterminés dans les SVM entraînés.

Les auteurs ont montré que l'équation 2.58 devient :

$$y = \text{sign}\left(\sum_k w_k^* s(B_j, x^k) + b\right) \quad (2.59)$$

Afin de choisir les nouveaux résultats, en se basant sur l'apprentissage actif, les auteurs ont choisi les sacs dont la distance avec l'hyperplan des SVM est la plus faible (le cas le plus ambigu) :

$$B^M = \text{argmin}_i |w^{*T} m_i + b| \quad (2.60)$$

Cette méthode ne demande pas beaucoup d'effort des modules d'analyse vidéo. Les modules d'analyse vidéo doivent comporter une seule tâche : la détection d'objets. Cependant, avec ce travail, l'indexation et la recherche de vidéos pour la vidéosurveillance sont réduites à celles d'images au niveau régions. La seule différence est que la segmentation d'images divise une image en régions qui peuvent être les objets ou le fond alors que la détection d'objets ne détermine que les régions des objets. Ce travail possède deux inconvénients. Premièrement, l'aspect temporel qui fait

une grande différence entre l'image et la vidéo n'est pas considéré. Deuxièmement, ce travail ne profite pas les résultats obtenus en suivi d'objets et reconnaissance d'événements pour la vidéosurveillance.

Le travail de Chen et al. dans [Chen 2006], [Chen 2007] permet de reconnaître les événements en se basant sur l'interaction de l'utilisateur. La trajectoire de l'objet est utilisée dans ce travail. La figure 2.27 montre les modules effectués dans ce travail.

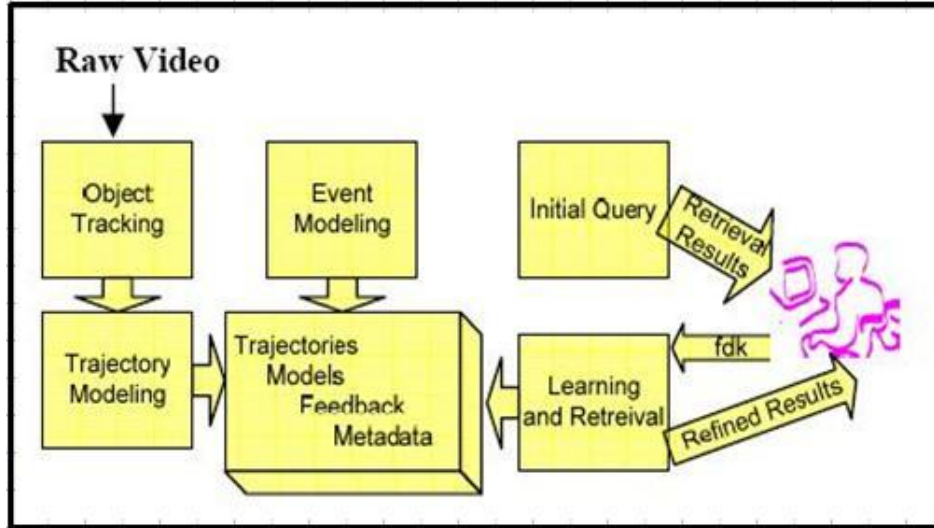


FIG. 2.27 – Architecture de l'approche interactive pour la détection des accidents dans les enregistrements de routes. Les véhicules sont détectés et suivis. Les événements d'intérêt sont modélisés. Un réseau de neurones est créé et entraîné en utilisant les retours des utilisateurs ([Chen 2006]).

Une fois la détection et le suivi d'objets faits, les trajectoires des objets sont représentées. Le modèle de trajectoire pour un seul véhicule :

$$\alpha = [\alpha_1, \dots, \alpha_n] \quad (2.61)$$

α_i est défini par :

$$\alpha_i = [v_i, vdiff_i, \theta_i] \quad (2.62)$$

où v_i est la vitesse au point i , $vdiff_i$ et θ_i sont la différence de vitesse et celle d'angle entre deux points consécutifs. Le modèle de trajectoire pour deux véhicules qui sont présents dans les mêmes frames est également défini par 2.61. Cependant α_i est :

$$\alpha_i = [v1diff_i, v2diff_i, cat_i] \quad (2.63)$$

où $v1diff_i$ et $v2diff_i$ sont des différences de vitesse entre deux points consécutifs de deux véhicules, cat_i est la distance entre deux véhicules. Plus la valeur de cat_i est petite, plus la probabilité d'accident est élevée.

Les auteurs ont étendu le réseau de neurones pour qu'il puisse travailler avec la trajectoire. Pour cela, une fenêtre ayant la taille m est utilisée. L'objectif est de déterminer l'événement correspondant au point k si l'on connaît les m points précédents. Le problème peut être modélisé par :

$$f_c : (x_{k-m}, \dots, x_{k-1}) \rightarrow c_i \in C \quad (2.64)$$

où C est l'ensemble d'événements prédéfinis. Afin de représenter le retour de l'utilisateur, en plus des m noeuds de la couche d'entrée du réseau, un noeud prenant en compte le retour de l'utilisateur nommé fdk est rajouté.

Le processus de recherche peut être résumé comme suit : l'utilisateur indique son événement d'intérêt. Si l'événement concerne un véhicule, le modèle de trajectoire pour un véhicule (cf. équations 2.61 et 2.62) est utilisé. Les véhicules dont la valeur $vdiff_i * \theta$ est la plus élevée sont retrouvés. Si l'événement concerne deux véhicules, le modèle de trajectoire pour deux véhicules (cf. équations 2.61 et 2.63) est employé. Pour cela, les véhicules dont le modèle ayant la valeur de cat_i petite et la valeur de $v1diff_i * v2diff_2$ élevée sont retrouvés.

L'utilisateur peut indiquer des résultats pertinents et non pertinents. Dans le cas où l'événement concerne un seul véhicule, le réseau est entraîné par le vecteur $[\alpha_{t-2}, \alpha_{t-1}, \alpha_t, fdk, opt]$ où la taille de la fenêtre est 3 (qui est représenté par les 3 vecteurs $\alpha_{t-2}, \alpha_{t-1}$ et α_t), fdk est 0 si le résultat est non pertinent, fdk est augmenté une valeur ε si le résultat est pertinent, opt est la valeur souhaitable pour la sortie du réseau, opt est 1 (respectivement 0) si le résultat est pertinent (respectivement non pertinent).

Ce travail est spécifique, il est dédié à la recherche d'accidents dans les vidéos. Le réseau de neurones est étendu afin de travailler avec les informations temporelles. Cependant, le choix de taille de la fenêtre est difficile.

2.3.4 Discussions

Il est difficile de comparer les approches proposées, car elles sont basées sur des modules d'analyse vidéo différents et elles ont été évaluées sur des bases de vidéos différentes. Dans cette section, nous les résumons dans le tableau 2.4 selon les niveaux auxquels l'indexation et la recherche effectuent : objets (Objets) ou événements (Événements), la présence du langage de requête (Requête) et la présence du retour de pertinence (Retour de pertinence). Dans la section suivante, nous présentons les problèmes ouverts dans ce domaine et expliquons la différence entre notre travail de thèse et les travaux dans l'état de l'art.

2.4 Conclusion et problèmes ouverts

Dans les sections précédentes, nous avons analysé séparément des techniques dans l'indexation et la recherche d'images, de vidéos scénarisées et de vidéos non scénarisées (surtout pour la vidéosurveillance). Nous présentons des résultats obtenus avec des techniques présentées en indexation et recherche de vidéos pour la

TAB. 2.4 – Résumé des approches pour l'indexation et la recherche de vidéos pour la vidéosurveillance selon les niveaux auxquels l'indexation et la recherche effectuent : niveau objets (Objets) ou/et niveau événements (Événements), la présence du langage de requête (Langage) et la présence du retour de pertinence (RP).

Référence	Objets	Événements	Langage	RP
[Conaire 2006]	✓	-	-	-
[Yuk 2007]	✓	-	-	-
[Ma 2007] [Cohen 2008]	✓	-	-	-
[Calderara 2006]	✓	-	-	-
[Meessen 2006]	✓	-	-	-
[Stringa 1998] [Stringa 2000]	✓	✓	-	-
[Foresti 2002]	✓	✓	-	-
[Greenhill 2002]	✓	✓	✓	-
[Hampapur 2007] [Tian 2008]	✓	✓	-	-
[Hu 2007]	✓	✓	-	-
[Xie 2004b]	✓	✓	-	-
[Ghanem 2004]		✓	✓	-
[Meessen 2007]	✓	-	-	✓
[Chen 2006] [Chen 2007]	✓	✓	-	✓

vidéosurveillance et aussi les problèmes ouverts à aborder. Nous rappelons brièvement nos contributions de notre travail et les différences entre notre travail avec les techniques dans l'état de l'art. Nous analysons également les relations entre l'indexation et la recherche d'images et celles de vidéos scénarisées avec notre travail de thèse.

2.4.1 Problèmes ouverts

Nous rappelons les 6 questions soulevées en indexation et recherche de vidéos pour la vidéosurveillance que nous avons présentées dans le chapitre 1.

- Question 1 : Quelles sont les informations noyées dans une vidéo qu'il faut calculer et stocker pendant la phase d'indexation, pour que la phase de recherche soit capable de répondre à toutes les requêtes de l'utilisateur ?
- Question 2 : Quels descripteurs peut-on extraire ?
- Question 3 : Que cherche l'utilisateur dans un enregistrement de vidéosurveillance ?

- Question 4 : Comment l'utilisateur formule-t-il ses propres requêtes et que peut fournir l'utilisateur ?
- Question 5 : Comment peut-on mettre en correspondance entre les informations indexées et la requête en se basant sur les descripteurs ?
- Question 6 : Comment et à quel niveau l'utilisateur peut-il interagir avec le système ?

L'analyse d'approches dans l'état de l'art montrent qu'elles arrivent à répondre partiellement ces questions.

Cependant, les quatre problèmes suivants sont encore ouverts dans ce domaine. Le premier problème est que les approches d'indexation et de recherche proposées sont spécifiques pour un module d'analyse vidéo. Le deuxième problème concerne l'expression des requêtes. La plupart des approches supposent d'avoir un ensemble de requêtes prédéfinies. Le troisième problème concerne la mise en correspondance entre les informations indexées et la requête en se basant sur les descripteurs. Très peu d'approches étudient les problèmes des modules d'analyse vidéo à savoir : (1) les objets ne sont pas toujours bien détectés, (2) les événements ne sont pas tous reconnus. Le quatrième problème est que l'interaction avec l'utilisateur est rarement considérée ([Meessen 2007], [Chen 2006] et [Chen 2007]).

L'objectif de notre thèse est de résoudre les quatre problèmes. Nous proposons un modèle de données qui nous permette de travailler avec les résultats provenant de modules d'analyse vidéo différents. Un langage de requête est également proposé. Afin de représenter des objets, nous présentons une méthode de détection des blobs représentatifs. Pour la mise en correspondance entre objets, nous proposons une nouvelle mise en correspondance. Concernant le retour de pertinence, deux méthodes de retour de pertinence sont présentées.

Nous expliquons les différences de notre travail dans cette thèse et les travaux de l'état de l'art.

Le modèle de données proposé comprend deux concepts : objets et événements. La notion SCAT de Calderara et al. [Calderara 2006] peut être considérée un cas particulier de concept objets. Cependant, le concept objets est très riche. En plus de l'apparence visuelle, d'autres informations de l'objet telles que l'intervalle du temps sont caractérisées.

Pour les descripteurs, nous extrayons aussi les couleurs dominantes et les histogrammes de contours. Cependant, à la différence des deux techniques de Yuk et al. [Yuk 2007] et Connaire et al. [Connaire 2006] qui calculent les valeurs moyennes de couleurs dominantes et d'histogrammes de contours sur tous les blobs des objets, nous ne les extrayons que sur les blobs représentatifs qui sont déterminés pour chacun des objets. Cela peut éviter d'accumuler des erreurs de la détection et du suivi d'objets.

La méthode de détection des blobs représentatifs présentée dans cette thèse incluant une phase de classification qui permet d'enlever des blobs non pertinents peut dépasser la limitation de celle de Ma et al. [Ma 2007] : corriger des erreurs qui sont présentes dans un grand nombre de blobs.

Pour la mise en correspondance entre objets, les approches de Ma et al. [Ma 2007], [Cohen 2008] et de Calderara et al. [Calderara 2006] permettent d'apparier les objets ayant un nombre de blobs différent. Cependant, elles ne sont pas appropriées pour des vidéos bruitées. Nous montrons dans les chapitres 5 et 6 que la nouvelle mise en correspondance arrive à travailler avec des vidéos bruitées.

Nous proposons également un langage de requête sous la forme SQL qui nous permet de dépasser les limitations de Greenhill et al. [Greenhill 2002] et de Ghanem et al. [Ghanem 2004] qui travaillent sur les méta-données.

À la différence des approches de Meessen et al. [Meessen 2007] et Chen et al. [Chen 2006], [Chen 2007], nous nous intéressons au retour de pertinence pour la recherche de vidéos au niveau objets pour deux raisons. Premièrement, la recherche des objets ne donne pas toujours les résultats pertinents à la première fois de recherche en présence de l'imperfection de la détection et du suivi d'objets. Elle a besoin d'échanges avec l'utilisateur. Deuxièmement, comme nous expliquons dans la chapitre 1, dans un système de vidéosurveillance, une alarme est déclenchée si le système détecte un événement intéressant. Habituellement, le personnel de sécurité veut trouver des informations antérieures concernant le ou les objets impliqués dans cet événement. Si la recherche d'objets donne des résultats appropriés, le personnel de sécurité peut prendre connaissance des événements associés aux objets trouvés en regardant la partie de la vidéo.

2.4.2 Relation entre l'indexation et la recherche d'images et notre travail

Il faut noter un grand point commun entre l'indexation et la recherche d'images et celles de vidéos pour la vidéosurveillance : la recherche de vidéos pour la vidéosurveillance au niveau objets basée sur les descripteurs d'apparence est équivalente à celle d'images au niveau régions. En représentant un objet physique par un ensemble de ses blobs, un objet physique correspond à une image alors qu'un blob correspond à une région. Avec cela, des descripteurs, des méthodes de mise en correspondance et de retour de pertinence en indexation et recherche d'images peuvent être appliqués à celle de vidéos pour la vidéosurveillance. Cependant, cela n'est pas simple et direct pour 4 raisons.

La première raison est qu'une image correspond à un petit nombre de régions. La détection de régions représentatives n'est pas nécessaire. Il est peut-être nécessaire d'enlever des régions dont la taille est mineure. Les régions sont disjointes. Des méthodes de segmentation différentes permettent de diviser une image de manières différentes. Cependant, aucune nouvelle région n'est rajoutée et aucune région n'est disparue. La qualité de la segmentation peut être compensée par la mise en correspondance appropriée. Dans la vidéosurveillance, selon le temps pendant lequel l'objet est présent dans la scène, un objet peut avoir un très grand nombre de blobs. En plus, certains blobs sont visuellement similaires. Comme les objets physiques interagissent, des méthodes de détection et de suivi d'objets différentes peuvent rajouter des blobs d'autres objets physiques ou perdre des blobs de l'objet

suivi. Cela nous demande fortement de détecter des blobs représentatifs pour chaque objet physique.

La deuxième raison est que l'apparence de l'objet physique dans les vidéos peut être difficile à caractériser (le mouvement de l'objet est complexe, l'interaction avec d'autres objets est souvent présente, l'objet est observé sous des conditions différentes). De plus, la définition "visuellement similaire" est plus stricte que celle dans la recherche d'images. Par exemple, deux personnes indexées sont similaires si elles représentent de la même personne réelle à différents moments ou observée par différentes caméras. Dans la recherche d'images, deux images de la mer prises à différent l'endroit sont similaires car ces images ont la même couleur (bleu). Le choix de descripteurs utilisés dans la recherche de vidéos pour la vidéosurveillance est donc rigoureux.

La troisième raison concerne le retour de pertinence. Pour le retour de pertinence au niveau régions en indexation et recherche d'images, si on ne demande pas à l'utilisateur de juger des régions positives/négatives, une image jugée positive peut avoir des régions positives et aussi négatives. Il est nécessaire d'appliquer les techniques afin de déterminer des régions positives. Ce problème est réglé en recherche de vidéos pour la vidéosurveillance au niveau objets. Comme les blobs représentatifs d'un objet sont des blobs pertinents qui représentent des aspects d'apparence différents de l'objet, si un objet est jugé positif, tous ses blobs sont positifs.

À partir de cette analyse, nous pouvons profiter certains résultats en indexation et recherche d'images pour notre travail. Premièrement, après avoir détecté des blobs représentatifs, méthode de mise en correspondance se base sur la distance EMD qui a été utilisée en indexation et recherche d'images. Deuxièmement, des descripteurs d'apparence proposés en indexation et recherche d'images sont également utilisés afin de comparer des objets. Finalement, pour le retour de pertinence, nous proposons deux techniques en inspirant à partir de celles en indexation et recherche d'images dont l'une appartenant à la famille des techniques basées sur la modification de requête et l'autre appartenant à la famille des techniques basées sur la classification.

2.4.3 Relation entre l'indexation et la recherche de vidéos scénarisées et notre travail

L'indexation et la recherche de vidéos pour la vidéosurveillance font partie de celles de vidéos non scénarisées. Cependant, la détection des blobs représentatifs pour chaque objet est similaire à la détection des images clés. L'objectif de ces détections est de choisir parmi un grand ensemble de blobs (images), des blobs (images) représentatifs selon certains critères. De nombreuses techniques dédiées à la détection des images clés en indexation et recherche de vidéos scénarisées peuvent être appliquées à la détection des blobs représentatifs. Il est à noter deux caractéristiques. Premièrement, la détection des blobs représentatifs travaille sur un niveau plus fin que celle des images clés. Les descripteurs utilisés doivent tenir compte de cette caractéristique. Deuxièmement, puisque la détection et le suivi d'objets ne sont pas parfaits, la détection des blobs représentatifs doit pouvoir enlever des blobs

non pertinents. La détection des images clés en indexation et recherche de vidéos scénarisées ne possède pas cette caractéristique.

Il faut remarquer que de nombreux travaux intéressants sur la fusion de plusieurs modalités sont présentés en indexation et recherche des journaux télévisés. Comme notre travail dans cette thèse ne se focalise pas sur la multimodalité, nous n'analysons pas ce point. Cependant, une de nos perspectives visant à étendre notre approche en combinant de plusieurs modalités est très proche des travaux en fusion de plusieurs modalités pour les vidéos scénarisées. Nous présentons cette perspective dans le chapitre 7.

Approche proposée

Nous présentons brièvement dans ce chapitre l'approche que nous proposons pour l'indexation et la recherche de vidéos pour la vidéosurveillance. L'approche proposée est composée de deux phases : la phase d'indexation et celle de recherche qui seront ensuite détaillées dans les chapitres 4 et 5 respectivement.

3.1 Description générale de l'approche proposée

La figure 3.1 illustre l'architecture de l'approche proposée qui se base sur un module externe et est composée de deux phases : l'indexation et la recherche. Les boîtes ovales correspondent aux données tandis que les rectangles illustrent les tâches. Nous précisons en bleu les parties correspondant à nos contributions dans cette thèse.

Comme nous l'indiquons dans le chapitre 1, notre approche se base sur les trois hypothèses suivantes :

- Hypothèse 1 : les vidéos doivent être prétraitées par un module d'analyse de vidéos ;
- Hypothèse 2 : nous ne travaillons que sur des vidéos non éditées ;
- Hypothèse 3 : notre travail n'utilise que des données visuelles.

Ces hypothèses sont prises en compte dans l'approche proposée. Les vidéos sont tout d'abord prétraitées par un module externe (hypothèse 1). Ce module d'analyse de vidéos doit effectuer obligatoirement la détection d'objets et le suivi d'objets. La classification d'objets et la reconnaissance d'événements sont facultatives. La compatibilité entre les sorties des modules d'analyse vidéo et l'entrée de la phase d'indexation est assurée par une interface analyse de vidéos/indexation. Puisque les vidéos étudiées sont des vidéos non éditées (hypothèse 2) et que nous n'exploitons que des données visuelles (hypothèse 3), la détection de transitions, la décomposition d'une vidéo en plans, le regroupement des plans en scènes et en groupes, l'analyse auditive ne sont donc pas analysés ni dans les modules d'analyse, ni dans la phase d'indexation.

Pour la phase d'indexation, afin d'utiliser les sorties des modules d'analyse de vidéos, un modèle de données est proposé. La phase d'indexation vise à sélectionner à partir des sorties des modules d'analyse de vidéos des éléments définis dans le modèle de données (la tâche **représentation d'objets**), à extraire des descripteurs (la tâche **extraction de descripteurs**) et à les stocker dans la base de données (la tâche **indexation**).

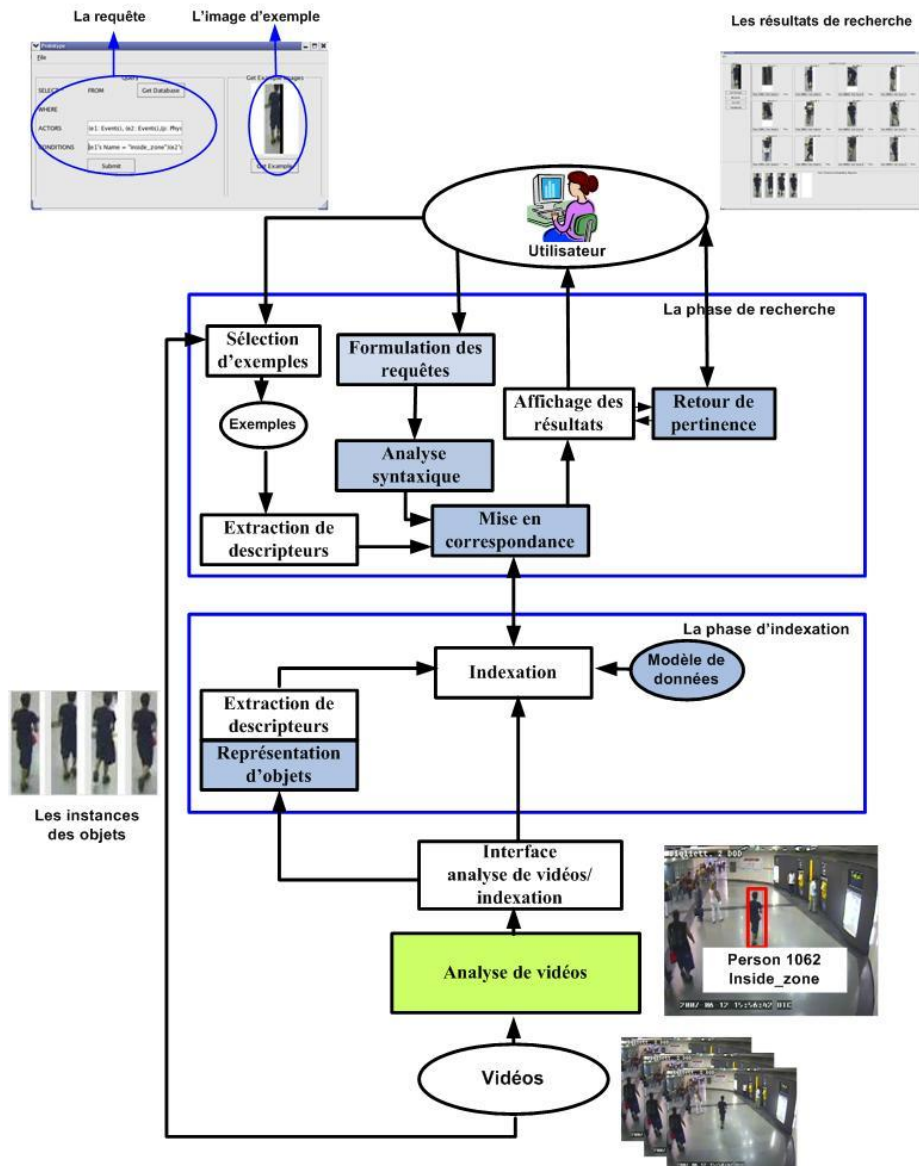


FIG. 3.1 – Tâches principales de la phase d'indexation et de celle de recherche de l'approche proposée. Les parties en bleu sont nos contributions dans cette thèse.

La phase de recherche communique avec les utilisateurs d'une part pour recevoir leurs requêtes, d'autre part pour leur rendre les résultats de la recherche. Les requêtes sont formulées en utilisant le langage de requêtes proposé (la tâche **formulation des requêtes**). Elles peuvent être associées à des images d'exemple (la tâche **sélection d'exemples**). Afin de pouvoir mettre en correspondance les images d'exemple et les objets indexés, des descripteurs sont également extraits dans les images d'exemple (la tâche **extraction de descripteurs**). Les requêtes sont analysées par la tâche **analyse syntaxique**. La tâche **mise en correspondance** consiste à appairer la requête avec les informations indexées. Ensuite, la tâche **affichage des résultats** ordonne les résultats selon leurs distances avec la requête et les rend à l'utilisateur. La tâche **retour de pertinence** permet d'interagir avec l'utilisateur afin de mieux répondre à ses requêtes.

3.2 Interface analyse vidéo/indexation

Puisque des modules d'analyse vidéos différents peuvent avoir des syntaxes différents de leurs sorties, cette interface assure la compatibilité entre les sorties des modules d'analyse vidéo et l'entrée de la phase d'indexation. Nous distinguons deux cas : les sorties sont des événements ou des objets mobiles. Si les sorties sont des événements, cette interface convertit la syntaxe des modules d'analyse vers notre syntaxe. Si les sorties sont des objets, l'interface prépare d'une part l'entrée de la tâche de représentation d'objets (l'ensemble de blobs) et d'autre part présente des information d'objets selon notre syntaxe.

3.3 La phase d'indexation

La phase d'indexation se base sur un modèle de données et effectue trois tâches principales : la représentation d'objets, l'extraction de descripteurs et l'indexation.

3.3.1 Modèle de données

Le modèle de données a un double objectif. D'une part, il permet de déterminer les informations qui seront indexées. D'autre part, il fournit aux utilisateurs des éléments pour exprimer leurs requêtes. Nous définissons dans le modèle de données deux concepts principaux : les objets et les événements. Ces concepts seront décrits dans le chapitre 4. Le modèle de données proposé est général.

Premièrement, il est indépendant des algorithmes de vision de l'analyse vidéo. Quels que soient les algorithmes de vision utilisés, si les sorties sont des objets détectés (et des événements reconnus), le modèle de données peut être appliqué.

Deuxièmement, le modèle de données peut travailler avec les modules d'analyse vidéo à différents niveaux d'analyse : objets et événements. La détection et le suivi d'objets sont obligatoires alors que la classification d'objets et la reconnaissance d'événements sont facultatives.

3.3.2 Représentation d'objets

À la différence des objets dans les images fixes, les objets dans les vidéos sont des objets mobiles. Ils sont habituellement détectés et suivis dans plusieurs frames. En effet, un objet possède un ensemble de blobs. Cela nous permet de travailler avec des informations riches. Cependant, l'utilisation de tous les blobs a deux inconvénients : la redondance et l'inefficacité. Le nombre de blobs d'un objet est important et il existe des blobs dont les différences visuelles sont mineures. La détection d'objets peut être imparfaite et créer des blobs de mauvaise qualité. La mise en correspondance utilisant ces blobs n'est pas efficace. Malheureusement, la plupart des approches dans l'état de l'art (sauf le travail de Ma et al. [Ma 2007]) utilisent tous les blobs des objets. L'analyse précédente montre qu'il est nécessaire de choisir les blobs pertinents (qui sont également appelés blobs représentatifs) parmi les blobs d'un objet qui seront utilisés pour la mise en correspondance entre des objets.

La tâche de représentation d'objets vise à déterminer des blobs représentatifs. Nous proposons dans chapitre 4 deux méthodes de détection des blobs représentatifs : la méthode basée sur le changement d'apparence et celle basée sur le regroupement des blobs.

La méthode basée sur le changement d'apparence balaie tous les blobs d'un objet et choisit des blobs qui sont largement différents des blobs antérieurs comme blobs représentatifs. Cette méthode permet de prendre des blobs correspondant aux changements d'apparence de l'objet. Cette méthode suppose que des modules d'analyse vidéo ont une bonne détection et un bon suivi d'objets.

La méthode basée sur le regroupement des blobs permet de choisir efficacement des blobs représentatifs d'un objet quand la détection et le suivi de cet objet ne sont pas parfaits. Les blobs sont tout d'abord classés en deux classes : blobs avec objets ou sans objet. Ensuite, des blobs sans objet sont enlevés lorsque des blobs avec objets sont regroupés. Pour chacun des groupes, un blob représentatif sera déterminé.

3.3.3 Extraction de descripteurs

L'extraction de descripteurs vise à choisir et à extraire des descripteurs qui peuvent caractériser les objets. Bien que l'extraction de descripteurs soit faite dans les modules d'analyse de vidéos, il y a deux caractéristiques à prendre en compte. Premièrement, les modules d'analyse de vidéos ont en général une contrainte : ils doivent travailler en temps réel. Ils n'extraient pas beaucoup de descripteurs. Deuxièmement, le but de l'extraction de descripteurs pour l'indexation et celui pour l'analyse de vidéos sont différents. L'extraction de descripteurs des modules d'analyse vidéo a par exemple pour but de permettre de lier les occurrences des objets d'une frame à l'autre frame tandis que celle de l'indexation a pour but de pouvoir retrouver parmi plusieurs objets indexés les objets semblables aux requêtes.

Certaines approches dans l'état de l'art emploient le même type de descripteurs pour la détection, le suivi et l'indexation d'objets. Cependant, il est à noter que chacun des descripteurs possède des points forts et aussi des points faibles. L'utili-

sation du même type de descripteurs pour les deux tâches (le suivi et l'indexation d'objets) ne donc permet pas de surmonter ses points faibles. Dans notre travail de thèse, nous ne fixons pas le descripteur utilisé dans des module d'analyse vidéo. Dans la phase d'indexation, nous extrayons des descripteurs qui sont habituellement employés en indexation et recherche d'images et de vidéos. Si l'un des descripteurs est déjà extrait dans des modules d'analyse vidéo, il n'est donc pas calculé dans la phase d'indexation.

Nous employons quatre types de descripteurs : les couleurs dominantes, les matrices de covariance, les histogrammes de contours et les points d'intérêt. Ces descripteurs sont extraits sur les blobs représentatifs des objets. Par défaut, tous les quatre types de descripteurs sont calculés.

3.3.4 Indexation

L'indexation vise à précalculer sur la base de vidéos les informations qui permettront de retrouver efficacement les données recherchées. Les informations viennent d'une part du module d'analyse de vidéos et d'autre part de la tâche de représentation des objets et d'extraction des descripteurs. Dans le chapitre 4, nous décrivons les informations provenant du module d'analyse de vidéos et celles provenant de la tâche de représentation des objets et d'extraction des descripteurs. Il est à noter que nous nous intéressons à la qualité de recherche et non pas la vitesse de recherche. Les techniques permettant d'accélérer la vitesse de recherche telles que *Kd-tree* ne sont pas considérées dans cette thèse.

3.4 La phase de recherche

La phase de recherche comporte sept tâches : la formulation des requêtes, la sélection d'exemples, l'analyse syntaxique, l'extraction des descripteurs, la mise en correspondance, l'affichage des résultats et le retour de pertinence.

3.4.1 Formulation des requêtes

La recherche commence par une requête de l'utilisateur. Afin de permettre à l'utilisateur d'exprimer ses requêtes, nous proposons un langage de requêtes basé sur le modèle de données proposé. Les requêtes exprimées dans ce langage sont à deux niveaux : les objets et les événements. Le langage de requêtes est comparable à celui Ghanem et al. [Ghanem 2004]. Cependant, grâce à la représentation des objets et à la mise en correspondance des objets, le langage de requête permet de retrouver non seulement les événements (comme le travail de Ghanem et al. [Ghanem 2004]) mais aussi les objets ayant des caractéristiques souhaitées. Nous fournissons dans le langage de requêtes plusieurs opérateurs permettant de comparer les objets par des critères différents.

3.4.2 Sélection d'exemples

Cette tâche permet à l'utilisateur de rajouter également des images d'exemple aux requêtes. Grâce à cette tâche, les requêtes peuvent être au niveau image. Les images d'exemple sont soit un blob représentant l'objet soit une région dans une image déterminée par l'utilisateur. Les images d'exemple peuvent être dans la même vidéo sur laquelle l'approche effectue la recherche ou d'autres vidéos.

3.4.3 Analyse syntaxique

Elle consiste à décomposer une requête en composantes. L'analyse syntaxique permet à l'approche (1) de comprendre ce que l'utilisateur veut chercher, (2) de savoir où elle va effectuer la recherche et (3) de savoir comment elle met en correspondance la requête et les informations indexées. Grâce à ces analyses, la tâche de mise en correspondance peut effectuer la recherche.

3.4.4 Extraction de descripteurs

Afin de comparer les images d'exemple et les objets indexés, l'extraction de descripteurs dans la phase de recherche calcule les mêmes types de descripteurs que celle dans la phase d'indexation. Cependant, elle est en ligne et ne s'effectue que sur les images d'exemple.

3.4.5 Mise en correspondance

La mise en correspondance consiste à apparier les composantes de la requête et les informations indexées. Nous distinguons les mises en correspondance par les concepts qui y participent (les images d'exemple, les objets ou les événements) et par le type de descripteurs (temporels ou visuels). Nous détaillons ces mises en correspondance dans le chapitre 5. Parmi ces mises en correspondance, nos contributions se focalisent sur la mise en correspondance entre des objets basée sur leurs blobs.

La mise en correspondance entre des objets en indexation et recherche de vidéos doit avoir trois propriétés : un objet possède un ensemble des blobs sur plusieurs frames, chacun des blobs représente un aspect visuel de l'objet ; le poids associé à un blob montre son degré d'importance ; le nombre de blobs des objets est varié. Nous présentons dans cette thèse une nouvelle mise en correspondance entre des objets basée sur la distance EMD. Cette distance a été utilisée en indexation et recherche d'images et de vidéos télévisées [Rubner 1998]. Dans cette thèse, elle est appliquée pour la première fois en indexation et recherche de vidéos pour la vidéosurveillance au niveau objets. La mise en correspondance proposée permet de comparer des objets ayant un nombre différent de blobs. Elle tient compte des poids des blobs et de la similarité visuelle de chaque paire de blobs. Nous décrivons cette méthode de mise en correspondance dans le chapitre 5. Une comparaison de cette méthode avec deux méthodes de mise en correspondance dans l'état de l'art est montrée dans le chapitre 6.

3.4.6 Affichage des résultats

Les résultats de la mise en correspondance de la requête avec les informations indexées sont ordonnés et transmis à l'utilisateur. La tâche d'affichage des résultats présente simplement une liste des résultats à l'utilisateur selon leurs distances avec la requête. Les techniques d'affichage des images telles que l'affichage d'images en utilisant des SOM (Self-Organizing Map) de Laaksonen et al. [Laaksonen 1999] ne sont pas considérées.

3.4.7 Retour de pertinence

Le retour de pertinence permet d'interagir avec l'utilisateur. Cette tâche utilise les retours de l'utilisateur pour améliorer les résultats pour cet utilisateur. Le retour de pertinence a été étudié en profondeur pour la recherche d'images [Rui 1998], [Rui 2001] tandis qu'il est récent pour les vidéos. Le retour de pertinence pour la recherche de vidéos est intéressant mais difficile à cause de l'aspect dynamique. Les objets sont en général caractérisés par plusieurs blobs et les événements peuvent être exprimés par un ensemble de contraintes spatiales et temporelles. En effet, les résultats de recherche peuvent être un ensemble de blobs et/ou un ensemble d'occurrences qui vérifient les contraintes prédéfinies. La technique de retour de pertinence prenant en compte cet aspect doit soit demander à l'utilisateur d'indiquer explicitement les parties pertinentes des résultats soit arriver à déduire à partir des retours globaux de l'utilisateur.

Nous nous intéressons au retour de pertinence pour la recherche de vidéos au niveau objets et pas au niveau événements pour deux raisons. Premièrement, la recherche d'objets ne donne pas toujours de résultats pertinents la première fois en raison de l'imperfection de la détection et du suivi des objets. Elle a besoin d'échanges avec l'utilisateur. Deuxièmement, comme nous l'expliquons dans le chapitre 1, dans un système de vidéosurveillance, une alarme est déclenchée si le système détecte un événement intéressant. Habituellement, les personnels de sécurité veulent trouver des informations antérieures concernant le ou les objets impliqués dans cet événement. Si la recherche d'objets donne les résultats appropriés, ils peuvent retrouver les événements des objets trouvés en voyant la partie de la vidéo concernant ces objets.

Nous proposons deux méthodes de retour de pertinence au niveau objets : le retour de pertinence basé sur plusieurs images d'exemple et le retour de pertinence basé sur les SVM à une classe. Nous remarquons que le retour de pertinence contient lui-même la mise en correspondance. Le retour de pertinence basé sur plusieurs images d'exemple met à jour la requête qui est initialisée par une image d'exemple, en utilisant des exemples positifs. Cette requête est comparée avec des objets indexés de la base de données grâce à la mise en correspondance proposée. Le retour de pertinence basée sur les SVM à une classe entraîne les SVM par des exemples positifs.

3.5 Conclusion

Nos contributions se focalisent donc sur :

- un **modèle de données** pour la phase d'indexation ;
- deux méthodes de **détection des blobs représentatifs** de l'objet pour la tâche de représentation d'objets dans la phase d'indexation ;
- un **langage de requêtes** avec un analyseur syntaxique pour les tâches de formulation des requêtes et d'analyse syntaxique dans la phase de recherche ;
- une nouvelle **mise en correspondance des objets** en se basant sur leurs blobs pour la tâche mise en correspondance dans la phase de recherche ;
- deux méthodes de **retour de pertinence pour la recherche au niveau des objets** pour la tâche de retour de pertinence dans la phase de recherche.

Nous pouvons résumer les caractéristiques de l'approche proposée :

- L'approche proposée travaille sur les sorties d'analyse vidéo. Les résultats obtenus par cette approche dépendent donc de la qualité des modules d'analyse vidéo. L'objectif est quelle que soit la qualité des modules d'analyse de vidéos de la compenser par les phases d'indexation et de recherche. Les méthodes présentées dans notre approche sont prévues pour travailler avec des modules d'analyse vidéo de qualités différentes ;
- L'approche proposée est générale. Cette caractéristique permet d'appliquer notre approche à des applications différentes.

Nous remarquons également les limitations de notre approche :

- Les tâches de sélection d'exemples et d'affichage des résultats sont simples ;
- Les techniques d'indexation permettant d'accélérer la vitesse de recherche ne sont pas encore considérées.

Indexation de vidéos de vidéosurveillance

4.1 Introduction

Ce chapitre est dédié à décrire la phase d'indexation. La phase d'indexation consiste à préparer les données pour que la phase de recherche puisse les employer pour répondre aux requêtes des utilisateurs. La préparation des données doit identifier d'une part quels types de données seront stockés dans la base de données et d'autre part comment ces données seront caractérisées. Concernant le type de données, nous définissons un modèle de données contenant deux principaux concepts abstraits : les objets et les événements. Les objets comprennent habituellement un grand nombre de blobs. Nous proposons deux méthodes qui permettent de choisir des blobs pertinents pour un objet. Concernant le descripteur, nous employons deux types de descripteurs : des descripteurs d'apparence et des descripteurs temporels. Les descripteurs d'apparence proposés dans l'état de l'art sont extraits sur les blobs représentatifs déterminés. Pour les descripteurs temporels, nous présentons deux méthodes de représentation des trajectoires. Le chapitre est composé de quatre grandes sections : le modèle de données, l'extraction des descripteurs d'apparence, l'analyse des descripteurs temporels et la représentation d'objets.

4.2 Modèle de données

Dans cette section, nous donnons tout d'abord la définition du modèle de données. Puis, nous détaillons les concepts abstraits de notre modèle de données.

Le modèle de données est un modèle qui décrit de façon abstraite comment sont représentées les données extraites à partir des vidéos.

Le modèle de données doit répondre à un double objectif : d'une part, il sert à déterminer les informations qui sont calculées dans la phase d'indexation et d'autre part, il est utilisé pour formuler les requêtes.

Notre modèle de données contient deux concepts abstraits : les objets, les événements.

4.2.1 Objets

Vu [Vu 2004] a défini l'objet physique dans les vidéos de vidéosurveillance. Selon l'auteur :

"Les objets physiques sont les objets du monde réel qui apparaissent dans les scènes observées par les caméras."

Les objets physiques sont divisés en deux types : les objets de contexte et les objets mobiles.

Les objets de contexte sont des objets physiques qui sont habituellement statiques (p. ex. les murs). Dans le cas où ils ne sont pas statiques, leurs mouvements peuvent être prédits par les informations contextuelles p. ex. les chaises, les portes sont des objets de contexte.

Les objets mobiles sont des objets physiques qui peuvent être perçus dans les scènes par leurs mouvements. Il est cependant difficile de prédire leurs mouvements p. ex. les personnes, les véhicules.

TAB. 4.1 – 9 attributs des objets mobiles.

Nom	Description
ID	l'étiquette de l'objet
Class	la classe à laquelle l'objet appartient
[2D_positions]	les positions en 2D (x, y) en plan image
[3D_positions]	les positions en 3D (X, Y, Z)
Blobs	les blobs représentatifs
Weights	les poids des blobs représentatifs
Time_interval	l'intervalle de temps
Descripteurs	
R_{ap}	la représentation de l'objet par les descripteurs d'apparence
$[R_t]$	la représentation de l'objet par les descripteurs temporels

Le tableau 4.1 liste les 9 attributs des objets mobiles. Parmi ces attributs *ID*, *Class*, *2D_positions*, *3D_positions*, *Time_interval* sont des attributs provenant directement de la sortie du module d'analyse vidéo.

Un objet détecté par les modules d'analyse de vidéos a son étiquette (*ID*). L'étiquette de l'objet est maintenue pendant le suivi d'objet. L'étiquette est un numéro généré par la détection et le suivi d'objets. Elle est un attribut distinct.

La classe de l'objet (*Class*) est une chaîne de caractères qui indique le type de l'objet. Elle est *Physical_objects* par défaut. Si la classification d'objets est disponible dans les modules d'analyse vidéo, cet attribut sera déterminé par la classification.

Les positions en 2D de l'objet sont habituellement des positions du centre de son blob dans le plan image. Elles sont déterminées sur les frames où l'objet est détecté et suivi. Les positions en 3D de l'objet sont des positions dans le monde réel. Les positions en 2D et 3D des objets sont facultatifs.

Les attributs *Blobs* et *Weights* comportent l'ensemble des blobs et leurs poids de l'objet. Si l'objet est détecté dans un frame, une boîte englobante minimale

qui l'entoure est créée par les modules d'analyse vidéo. Le blob est une partie du frame déterminé par la boîte englobante minimale. Comme un objet est détecté et suivi pendant certains frames, un ensemble de blobs sont donc identifiés. Grâce aux méthodes de détection des blobs représentatifs qui seront présentées dans les sections suivantes, chacun des objets possède un ensemble de blobs représentatifs associés à leurs poids.

L'attribut *Time_interval* indique le temps pendant lequel l'objet est présent dans la scène. Cet attribut est déterminé par deux points limites : $[I^l, I^h]$ où I^l et I^h indiquent les frames où l'objet apparaît et disparaît dans la scène.

Les attributs R_{ap} et R_t sont calculés par la tâche *Extraction de descripteurs* de notre approche. Ils comprennent des descripteurs visuels extraits sur les blobs ou des descripteurs de trajectoires de l'objet.

Les attributs *ID*, *Class*, *2D_positions*, *3D_positions*, *Time_interval* sont des méta-données. Une fois ces attributs calculés par les modules d'analyse vidéo, ils sont stockés et utilisés pour la mise en correspondance entre des objets dans la phase de recherche. Notons que la mise en correspondance entre des objets en se basant sur ces attributs est exacte comme dans les bases de données traditionnelles.

Les objets de contexte sont considérés comme un cas particulier des objets mobiles : ils ont trois attributs obligatoires : *ID*, *Class*, *3D_positions*.

La figure 4.1 montre les attributs d'un objet mobile dont $ID = 57$. Cet objet appartient à la classe *Person*. Il est détecté pendant 143 frames (du frame #2017 au frame #2160). Son attribut *Time_interval* est $[2017, 2160]$. Cinq blobs représentatifs associés à leurs poids sont déterminés pour cet objet. Pour chacun des blobs, un vecteur de 5 éléments de l'histogramme de contours est calculé pour l'attribut R_{ap} .

La figure 4.2 montre les attributs d'un objet de contexte dont $ID = 1$. Cet objet appartient à la classe *Gates*.

En indexation et recherche d'images et de vidéos, l'utilisateur cherche habituellement les objets dans la base d'images et de vidéos qui sont semblables à une image d'exemple. Nous proposons un concept image d'exemple. Ce concept abstrait n'est utilisé que dans la phase de recherche. Le tableau 4.2 montre les 2 attributs des images d'exemple.

TAB. 4.2 – 2 attributs des images d'exemple.

Nom	Description
Image	
Descripteurs	
R_{ap}	la représentation de l'image d'exemple par les descripteurs d'apparence

L'attribut *Image* est une imagerie contenant un objet recherché. La représentation de l'image d'exemple par les descripteurs d'apparence R_{ap} est un cas particulier de celle des objets où l'objet a un seul blob. La figure 4.3 illustre une image d'exemple et sa représentation. Le descripteur d'apparence choisi est l'histogramme


ID	57
Class	Person
2D_positions	(91, 46), (82,51), ...
3D_positions	
Blobs	
Weights	0.1 0.79 0.03 0.03 0.02
Time_interval	(2017, 2160)
Rap	0.298900 0.146351 0.170871 0.159658 0.088181 0.270665 0.140725 0.158958 0.144340 0.128179 0.354022 0.131959 0.136261 0.171444 0.044730 0.326137 0.160978 0.141311 0.179290 0.064959 0.248816 0.151436 0.132339 0.167929 0.109414
Rt	

FIG. 4.1 – Attributs d'un objet dont ID = 57. Cet objet appartient à la classe Person. Il est détecté et suivi pendant 143 frames (du frame #2017 au frame #2160). Cinq blobs représentatifs associés au poids sont déterminés pour cet objet. Pour chacun des blobs, un vecteur de 5 éléments de l'histogramme de contours est calculé pour l'attribut R_{ap} .


ID	1
Class	Gates
3D_positions	(-170, 300, 0), (-170, 500, 0), (200, 500, 0), (200, 320, 0)
Blobs	
Weights	1.0

FIG. 4.2 – Attributs d'un objet de contexte (Gates) dont ID = 1.

des contours.


Image	
Rap	0.364357 0.169483 0.148970 0.115303 0.027044

FIG. 4.3 – Une image d'exemple, l'attribut R_{ap} est un vecteur de 5 éléments de l'histogramme de contours.

4.2.2 Événements

Un événement en vidéosurveillance est tout ce qui concerne l'évolution des objets et l'interaction des objets dans la scène. Plusieurs termes ont été proposés pour caractériser les événements à différents degrés de granularité : les états comprenant les états primitifs et les états composés, les événements comprenant les événements primitifs et les événements composés, et les activités. Dans notre modèle de données, nous les représentons dans un seul concept : événements. Le tableau 4.3 liste les 6 attributs de ce concept.

Chaque événement est identifié par son étiquette (ID). L'attribut ID est un attribut distinct.

L'attribut $Name$ est une chaîne de caractères indiquant le nom de l'événement.

L'attribut $Confidence_value$ montre la valeur de confiance de la reconnaissance de cet événement. Cet attribut est 1 par défaut.

L'attribut $Involved_Physical_objects$ contient un ensemble d'étiquettes des objets impliqués dans l'événement. Les objets impliqués dans l'événement sont caractérisés par le concept d'objets.

L'attribut Sub_events est facultatif. Si un événement comprend des sous-événements, l'attribut Sub_events est un ensemble d'étiquettes de ses sous-événements. Les sous-événements sont également des événements.

L'attribut $Time_interval$ est l'intervalle de temps pendant lequel l'événement est reconnu. Il est représenté de même manière que celui de l'objet.

Les attributs des événements sont déterminés par la reconnaissance d'événements dans les modules d'analyse vidéo.

Un exemple d'un événement est montré dans la figure 4.4. L'objet impliqué dans cet événement est montré dans la figure 4.1.

TAB. 4.3 – 6 attributs des événements.

Nom	Description
ID	l'étiquette de l'événement
Name	le nom de l'événement (p. ex. Close_to)
Confidence_value	le degré de confiance
Involved_Physical_objects	les objets impliqués dans l'événement
[Sub_events]	les sous-événements de l'événement
Time_interval	l'intervalle de temps pendant lequel l'événement est reconnu

ID	50
Name	« Inside_zone_Patform »
Confidence_value	1.0
Involved_Physical_objects	57
Sub_events	
Time_interval	(2017,2017)



FIG. 4.4 – Un événement “inside_zone_Platforme” dont ID=50 est représenté dans le modèle de données. L’objet impliqué dans cette événement est montré dans la figure 4.1.

4.2.3 Discussions

Un modèle comprenant deux concepts principaux, objets et événements, est présenté. Les deux concepts vont être remplis par les informations provenant des modules d’analyse vidéo et des descripteurs analysés par les tâches dans la phase d’indexation. Le concept d’image exemple est déterminé par l’utilisateur dans les sessions de recherche. Le langage de requête présenté dans le chapitre 5 fournit un moyen à l’utilisateur pour accéder aux attributs des concepts dans ce modèle.

4.3 Extraction de descripteurs d’apparence

De nombreux descripteurs d’apparence des objets avec autant de mesures de similarités pouvant permettre de mettre en correspondance des objets ont été proposés. Nous employons trois types de descripteurs : la couleur (section 4.3.1), le contour (section 4.3.2) et les points d’intérêts (section 4.3.3). Puisque notre approche peut utiliser des sorties de plusieurs modules d’analyse vidéos, dans le cas où un de ces descripteurs est déjà calculé dans ces modules, ce descripteur n’est pas extrait dans la phase d’indexation. Par exemple, Trichet et al. [Trichet 2008] ont employé les

points d'intérêt pour la détection et le suivi d'objets. Ces descripteurs ne sont donc pas calculés dans la phase d'indexation. Il est à noter que notre contribution n'est pas de proposer un nouveau descripteur. Nous employons des descripteurs d'apparence habituellement utilisés dans l'état de l'art. Dans les sections suivantes, nous détaillons l'extraction de ces descripteurs. Par défaut, tous les descripteurs choisis sont extraits pour des objets de la base de données dans la phase d'indexation. Le descripteur utilisé dans la phase de recherche est décidé par l'utilisateur. Dans le chapitre 6, nous analysons la performance de ces descripteurs en indexation et recherche de vidéos pour la vidéosurveillance. En effet, le choix de descripteur peut être fait a priori.

4.3.1 Analyse de la couleur

4.3.1.1 Couleurs dominantes

Pour la couleur, nous employons les couleurs dominantes comme les descripteurs de MPEG-7 [Deng 1999], [Deng 2001], [Manjunath 2001]. Les évaluations effectuées par Annesley et al. [Annesley 2005] ont montré la performance des couleurs dominantes en indexation et recherche de vidéos de vidéosurveillance. Les couleurs dominantes donnent une représentation compacte des couleurs dans une image. L'algorithme d'extraction des couleurs dominantes se base sur l'algorithme généralisé de Lloyd [Linde 1980]. Avant de présenter l'algorithme d'extraction des couleurs dominantes, nous donnons quelques notions :

Soit :

- $CB = \{C_i | i = 1, \dots, N\}$: l'ensemble des groupes C_i avec les centres de gravité $c_{CB, i}$;
- $CB^0 = \{C_i^0 | i = 1, \dots, N_0\}$: l'ensemble des groupes initiaux ;
- $CB^t = \{C_i^t, i = 1, \dots, N_t\}$: l'ensemble des groupes à l'itération t ;
- $DC(I) = \{(c_i, p_i) | i = 1, \dots, N\}$: les couleurs dominantes identifiées pour l'image I ;
- N : le nombre de couleurs dominantes ;
- c_i : le vecteur de 3 composants de couleur ;
- p_i : le poids associé à la couleur c_i .

L'extraction des couleurs dominantes pour une image est montré par l'algorithme 1. Le centre de gravité du groupe C_i^t est $c_{CB, i}^t$ qui est défini par :

$$c_{CB, i}^t = \frac{\sum v(n)x(n)}{\sum v(n)}, \quad x(n) \in C_i^t \quad (4.1)$$

où $x(n)$ est un vecteur de trois composants de couleur et $v(n)$ est le poids du pixel n . La distorsion du groupe C_i^t est :

$$D_i^t = \sum_n v(n) \|x(n) - c_{CB, i}^t\|^2, \quad x(n) \in C_i^t \quad (4.2)$$

Algorithme 1 : Extraction des couleurs dominantes de l'image I .**Input :** I : l'image N_{max} : le nombre maximum autorisé de couleurs dominantes θ : le seuil pour la distorsion, $0 \leq \theta \leq 1$ T_d : le seuil pour fusionner deux groupes**Output :** $DC(I) = \{(c_i, p_i)\}$: les couleurs dominantes de I **begin**1 convertir l'image I de l'espace de couleurs RVB à l'espace de couleurs CIE LUV2 initialiser le nombre de groupes $n=1$ 3 diviser I en 1 groupe C_1^0 4 calculer le centre $c_{CB,1}^0$ et la distorsion D_1^0 par les équations 4.1 et 4.2**do**

5 diviser le groupe dont la distorsion est plus élevé en deux groupes

6 incrémenter n de 1

7 calculer le centre de gravité et la distorsion de nouveaux groupes par l'équation 4.2

until la condition 4.3 est vérifiée8 **while** deux groupes proches sont détectés(la distance de couleur de leurs centres est inférieure à T_d)

9 fusionner les deux groupes détectés

10 **end**11 déterminer $DC(I) = \{(c_i, p_i)\} | i = 1, \dots, N$ **end**

Si le changement de distorsion de deux itérations consécutives est inférieur à un seuil θ ou le nombre de couleurs dominantes est supérieur au nombre maximum autorisé (cf. équation 4.3), la boucle de l'algorithme se termine.

$$\frac{(D^{t-1} - D^t)}{D^{t-1}} < \theta \text{ ou } n \geq N_{max} \quad (4.3)$$

où D^t, D^{t-1} sont les moyennes des distorsions D_i^t, D_i^{t-1} .

4.3.1.2 Matrices de covariance

La matrice de covariance a été utilisée en analyse de vidéos car elle arrive à fusionner plusieurs descripteurs tels que la couleur, la texture. De plus, elle permet de localiser les propriétés de la couleur et de la texture. Cette propriété est appropriée pour la vidéosurveillance car l'échelle et la couleur d'un objet peuvent varier considérablement entre des caméras différentes.

De nombreux descripteurs peuvent être mis dans une matrice de covariance. Afin de comparer notre approche avec celle de Ma et al. [Ma 2007], nous utilisons



FIG. 4.5 – (a) blob d'une personne détectée ; (b) et projection de 3 couleurs dominantes sur le blob.

les mêmes types de descripteurs que Ma et al. Ces descripteurs sont la position, la couleur et le gradient des composants de couleur. C'est pourquoi nous mettons l'analyse de matrice de covariance dans la section d'analyse de la couleur.

Soit f le vecteur de descripteurs du pixel (x, y) de l'image I , $CM(I)$ la matrice de covariance de l'image I . Le vecteur f est défini par :

$$f(x, y) = [x, y, R(x, y), G(x, y), B(x, y), \nabla R^T(x, y), \nabla G^T(x, y), \nabla B^T(x, y)] \quad (4.4)$$

où R , G , et B sont les valeurs de couleurs du pixel (x, y) . $\nabla R^T(x, y)$, $\nabla G^T(x, y)$ et $\nabla B^T(x, y)$ sont les valeurs du gradient spatial pour chaque composant de couleur R , G , B .

La matrice de covariance $CM(I)$ est déterminée pour tous les pixels de l'image I par :

$$CM(I) = \frac{1}{n-1} \sum_{x,y} (f - \bar{f})(f - \bar{f})^T \quad (4.5)$$

où n est le nombre de pixels de l'image I .

4.3.2 Analyse du contour

Nous choisissons les histogrammes des contours définis dans MPEG-7 [Park 2000], [Won 2002] afin de caractériser les contours. Les contours sont divisés en cinq types : vertical, horizontal, 45 degré, 135 degré et non directionnel (voir figure 4.6).

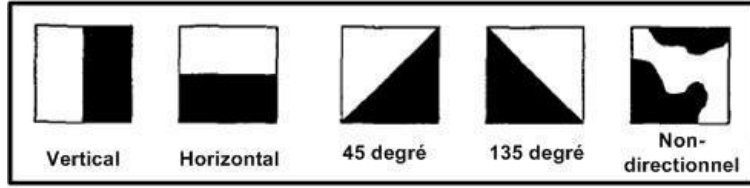


FIG. 4.6 – Cinq types de contours : vertical, horizontal, 45 degré, 135 degré et non directionnel [Park 2000].

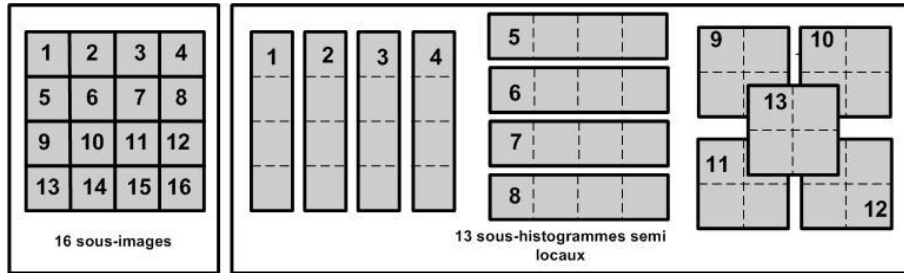


FIG. 4.7 – Décomposition d'une image en 16 sous-images pour calculer l'histogramme local et l'identification de 13 sous-histogrammes semi-locaux [Won 2002].

Trois histogrammes des contours (local, semi-local, global) peuvent être extraits pour une image. Nous présentons l'extraction de ces histogrammes des contours d'une image par l'algorithme 2. L'image est tout d'abord divisée en 16 sous-images (voir figure 4.7). Un histogramme de 5 éléments (un élément pour chacun des types de contours) est calculé pour chacune des sous-images. Un élément pour un type de contours est le nombre de fois que ce type de contours est présent dans la sous-image. Afin de calculer l'occurrence d'un type de contours pour une sous-image, cette sous-image est divisée en N blocs. La valeur de N est habituellement fixée pour que la taille du bloc est proportionnel à celle de l'image. Dans cette thèse, la valeur de N est 1110. Chaque bloc est convolué avec 5 filtres définis par :

$$\begin{aligned}
 filtre^{ver} &= \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix} & filtre^{hor} &= \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \\
 filtre^{45} &= \begin{bmatrix} \sqrt{2} & 0 \\ 0 & -\sqrt{2} \end{bmatrix} & filtre^{135} &= \begin{bmatrix} 0 & \sqrt{2} \\ -\sqrt{2} & 0 \end{bmatrix} \\
 filtre^{non} &= \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}
 \end{aligned} \tag{4.6}$$

Soient A_n^{ver} , A_n^{hor} , A_n^{45} , A_n^{135} , A_n^{non} les valeurs de convolution entre bloc A_n et les filtres, la présence d'un type de contours dans un bloc est déterminée par :

$$\max\{|A_n^{ver}|, |A_n^{hor}|, |A_n^{45}|, |A_n^{135}|, |A_n^{non}|\} \geq Th_{con} \tag{4.7}$$

Le nombre d'occurrences d'un type de contours dans une sous-image est accumulé par son occurrence dans tous les N blocs. L'histogramme local des contours est un histogramme de 80 (5×16) éléments qui est formulé à partir de 16 histogrammes des sous-images. L'histogramme global contient 5 éléments correspondant à 5 types de contours. La valeur de chaque élément est la somme des éléments de même type de contours de 16 histogrammes des sous-images. Afin de calculer l'histogramme semi-local, 13 sous-histogrammes semi-locaux des 5 éléments sont déterminés comme dans la figure 4.7. L'histogramme semi-local est un histogramme de 65 éléments (13×5) qui est créé à partir des 13 sous-histogrammes semi-locaux. Nous extrayons tous les trois types de histogrammes de contours (histogramme local, histogramme semi-local et histogramme global). Dans le chapitre 6, nous évaluons la performance de ces histogrammes de contours en indexation et recherche de vidéos pour la vidéo-surveillance. Ce choix d'histogramme utilisé peut être décidé en se basant sur cette évaluation.

Algorithme 2 : Extraction des histogrammes des contours de l'image I .

Input :

I : image en niveau de gris,
 N : le nombre de blocs pour une sous-image,
 Th_{con} : un seuil pour décider de la présence de contour.

Output :

$EH^{local}(I)$: l'histogramme local des contours
 $EH^{semi-local}(I)$: l'histogramme semi-local des contours
 $EH^{global}(I)$: l'histogramme global des contours

begin

```

1   diviser l'image  $I$  en 16 sous-images
   for chacune des sous-images do
2       créer un histogramme de 5 éléments
3       diviser la sous-image en  $N$  blocs
       for chacun des blocs  $A_n$  do
4           calculer la valeur moyenne de luminance de  $A_n$ 
5           convoluer le bloc  $A_n$  avec 5 filtres définis dans l'équation 4.6,
           les réponses obtenues  $A_n^{ver}$ ,  $A_n^{hor}$ ,  $A_n^{45}$ ,  $A_n^{135}$ ,  $A_n^{non}$ 
6           identifier le type de contour (cf. équation 4.7)
7           mettre à jour l'histogramme
       end
8       normaliser l'histogramme par le nombre de blocs
   end
9   créer l'histogramme local à partir de 16 histogrammes des sous-images
10  calculer l'histogramme semi-local et global
end

```

4.3.3 Analyse des points d'intérêt

La représentation des images par des descripteurs locaux s'est imposée dans nombre d'applications telles que la mise en correspondance entre des images, la classification, ou encore la détection d'objets. Cette représentation est en effet plus robuste à certaines transformations et altérations de l'image que les approches globales. De nombreux travaux ont prouvé que l'utilisation de points d'intérêts associés aux descripteurs est capable de pallier deux difficultés rencontrées dans l'appariement des images :

- la visibilité partielle (p. ex. l'occlusion des objets, la présence d'une partie de l'objet dans l'image) ;
- le changement de point de vue comprenant le changement d'échelle et le changement de l'angle de vue.

L'utilisation des points d'intérêt en indexation et recherche d'images et de vidéos comporte deux étapes : la détection des points d'intérêt et l'extraction des descripteurs pour les points détectés.

4.3.3.1 Détection des points d'intérêt

Un point d'intérêt (PI) est défini comme étant un point dans l'image où des changements significatifs se produisent. Des exemples de PI sont les coins, les jonctions, les points noirs sur fond blanc ou tout autre point marqué par un changement important de la texture.

Les points d'intérêt sont divisés en deux grandes catégories : l'une correspond aux coins et l'autre correspond aux régions. Les points d'intérêts concernant des coins peuvent être détectés par les méthodes de Harris-Laplacien, Harris Affine, tandis que ceux concernant les régions peuvent être détectés par les méthodes de DoG (Difference of Gaussians), Hessien- Laplacien, Hessien Affine, MSER (Maximally Stable Extremal Regions), etc. Une évaluation des points d'intérêts a été donnée dans [Mikolajczyk 2004].

Nous utilisons trois types de points d'intérêt : Harris Affine (cf. algorithme 3), MSER (cf. algorithme 4), DoG (cf. algorithme 5).

Nous remarquons que la sortie de l'algorithme 3 et 4 sont les régions affines centrées par les points d'intérêt tandis que celle de l'algorithme 5 sont les points d'intérêt. Nous utilisons le même terme points d'intérêt pour tous ces algorithmes. Soit p_i un point d'intérêt, p_i est un vecteur de 5 éléments si il est le point d'intérêt de Harris Affine ou MSER : (u, v, a, b, c) où u, v sont les positions du point d'intérêt, a, b, c déterminent l'ellipse $a(x-u)^2 + b(x-u)(y-v) + c(y-v)^2$. Si le point est celui de DoG, p_i est un vecteur de 4 éléments : (u, v, s, o) où u, v sont les positions du point, s est l'échelle sur laquelle le point est détecté et o est l'orientation dominante du point.

À l'étape (1) de l'algorithme 3, la détection de points d'intérêt de Harris Affine se base sur le détecteur de point d'intérêt de Harris.

Algorithme 3 : Détection des points d'intérêt de Harris Affine

Input : I : image**Output :** $IP(I)^{HarAff} = \{p_i\}$: les régions affines centrées par les points d'intérêt**begin**

- 1 détecter des points Harris multi-échelles
en appliquant les équations 4.8 et 4.9
- 2 sélectionner itérativement l'échelle et la localisation des points d'intérêt
- 3 déterminer la région affine centrée sur le point d'intérêt

end

Le détecteur de Harris vise à identifier des coins dans une image. Les points détectés par ce détecteur sont invariants à une rotation dans le plan image et au changement affine de luminosité. Ils ne sont cependant pas invariants au changement d'échelle et d'angle de vue. Une version adaptée aux changements d'échelles du détecteur de points d'intérêt de Harris est appliquée dans l'étape (1) :

$$\mu(x, \sigma_l, \sigma_D) = \begin{bmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{bmatrix} = \sigma_D^2 g(\sigma_l) * \begin{bmatrix} L_x^2(x, \sigma_D) & L_x L_y(x, \sigma_D) \\ L_x L_y(x, \sigma_D) & L_y^2(x, \sigma_D) \end{bmatrix} \quad (4.8)$$

où $g(\sigma_l)$ est une fenêtre gaussienne d'intégration, σ_D est l'échelle de détection, L_x , L_y sont les dérivées premières.

$$cornerness = \det(\mu(x, \sigma_l, \sigma_D)) - \text{atrace}^2(\mu(x, \sigma_l, \sigma_D)) \quad (4.9)$$

Le maximum local de *cornerness* détermine la localisation des points d'intérêts.

À l'étape (2) de l'algorithme 3, la sélection itérative d'échelle et localisation s'appuie sur une fonction F . Un point détecté à l'échelle s_n est un point d'intérêt si la réponse de la fonction F sur ce point vérifie :

$$\begin{aligned} F(x, s_n) &> F(x_w, s_n) \quad \forall x_w \in W \\ F(x, s_n) &> t_h \end{aligned} \quad (4.10)$$

où W identifie les 8 voisins de x à l'échelle s_n

$$\begin{aligned} F(x, s_n) &> F(x, s_{n-1}) \wedge F(x, s_n) > F(x, s_{n+1}) \\ F(x, s_n) &> t_l \end{aligned} \quad (4.11)$$

où s_n , s_{n-1} et s_{n+1} sont des échelles.

La fonction F est fixée par le laplacien [Mikolajczyk 2001] :

$$F = |s^2(L_{xx}(x, s) + L_{yy}(x, s))| \quad (4.12)$$

À l'étape (3) de l'algorithme 3, en utilisant la matrice des moments du second ordre, une estimation de forme des régions est obtenue.

Avant de décrire le détecteur de MSER (Maximally Stable Extremal Regions), nous donnons quelques définitions utilisées dans [Matas 2002].

Soit une image $I : D \subset Z^2 \rightarrow S$, les régions extrémales sont bien définies sur l'image I si :

- S est totalement ordonné c'est-à-dire qu'il existe une relation binaire \leq qui est réflexive, transitive et antisymétrique ;
- Une relation de voisinage $A \subset D \times D$ est définie : $p, q \in D$, pAq c'est-à-dire p est le voisin de q par la relation de voisinage A . La relation de voisinage peut être une relation de 4, 6 ou 8 voisinage.

Une région Q est un sous-ensemble connexe de D , c'est-à-dire pour chaque $p, q \in Q$, il existe les séquences $p, a_1, a_2, \dots, a_n, q$ et $pAa_1, a_1Aa_2, \dots, a_nAq$.

Une frontière $\varphi Q = \{q \in D \setminus Q : \exists p \in Q : qAp\}$, c'est-à-dire la frontière φQ de Q est l'ensemble des pixels qui sont voisins d'au moins un pixel de Q et qui n'appartiennent pas à Q .

Une région extrémale $Q \subset D$ est une région où $\forall p \in Q, q \in \varphi Q : I(p) > I(q)$ (correspondant aux régions d'intensité maximale), $I(p) < I(q)$ (correspondant aux régions d'intensité minimale).

Soit $Q_1, \dots, Q_{i-1}, Q_i, \dots$ une séquence des régions extrémales, $Q_i \subset Q_{i+1}$. La région extrémale Q_{i^*} est la région extrémale la plus stable si $q(i)$ définie par l'équation 4.13 obtient le minimum local à i^* .

$$q(i) = |Q_{i+\Delta} \setminus Q_{i-\Delta}| / |Q_i| \quad (4.13)$$

où $|\cdot|$ signifie la cardinalité, $\Delta \in S$ est un paramètre.

Il existe deux types de points d'intérêt de MSER. On l'appelle *MSER+* les points d'intérêt de MSER détectés sur l'image I et *MSER-* ceux détectés dans l'image en couleur négative de I . Les points d'intérêt de MSER sont détectés sur les images en niveaux de gris. La détection des points d'intérêts de MSER est montrée dans l'algorithme 4. Les points d'intérêt de MSER sont invariants une transformation affine de l'intensité des images ; Ils sont covariants aux transformations qui préservent la relation de voisinage.

La figure 4.8 présente les points d'intérêts de Harris Affine (b) et de MSER (c) détectés sur l'image de la personne (a).

Le détecteur DoG tout d'abord lisse une image par une gaussienne :

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (4.14)$$

où $*$ est opérateur de convolution, $G(x, y, \sigma)$ est une gaussienne à l'échelle σ et $I(x, y)$ est l'image d'entrée. La différence de deux images à l'échelle $k\sigma$ et l'échelle k est calculée par :

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (4.15)$$

Afin d'identifier des points d'intérêts, chacun des points à l'échelle i est comparé avec ses 8 voisins de l'échelle i et ses 9 voisins de l'échelle supérieure ($i + 1$) et

Algorithme 4 : Détection des points d'intérêt MSER.

Input :

I : image

Output :

$IP^{MSER}(I) = \{p_i\}$: les régions affines détectées de I

begin

- 1 ordonner les pixels de l'image I par leurs intensités
- 2 détecter les composantes connexes pour avoir les régions maximales
- 3 déterminer les régions extrémales les plus stables (cf. équation 4.13)
- 4 approximer les régions extrémales par les régions affines

end



FIG. 4.8 – (a) blob d'une personne détectée ; (b) points d'intérêt de Harris Affine ; (c) points d'intérêt de MSER.

inférieure ($i - 1$). Les points obtenant le maximum ou le minimum sont choisis comme des points d'intérêts.

Pour déterminer les orientations dominantes du point d'intérêt, la norme m (cf. équation 4.16) et l'orientation θ (cf. équation 4.17) du gradient sont calculées sur les pixels de la région centrée par le point d'intérêt à l'échelle où ce point est détecté.

$$m = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (4.16)$$

$$\theta = \tan^{-1} \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \quad (4.17)$$

Algorithme 5 : Détection des points d'intérêt DoG

Input :

I : image

Output : $IP^{DoG}(I) = \{p_i\}$: les points d'intérêts de DoG de I

begin

- 1 lisser l'image I par les gaussiennes à multi-échelles (cf. équation 4.14)
- 2 calculer la différence de deux images consécutives (cf. équation 4.15)
- 3 déterminer la position des points d'intérêt
- 4 déterminer les orientations dominantes des points d'intérêt

end

Un histogramme d'orientation de 36 éléments (pour couvrir 360 degrés d'orientation) est construit. L'approche consiste à identifier le maximum de cet histogramme et quelques maxima locaux ayant au moins 80% de la valeur maximale. Correspondant à chaque maximum, une orientation dominante du point d'intérêt est déterminée. L'identification des orientations dominantes pour chacun des points d'intérêt se basant sur les maximums de l'histogramme de l'orientation est illustrée dans la figure 4.9. En conséquence, un point d'intérêt peut avoir plusieurs orientations dominantes. À chaque orientation dominante, nous calculons un vecteur de descripteurs. Un point d'intérêt en effet peut avoir plusieurs vecteurs de descripteurs.

4.3.3.2 Descripteurs

Afin de mettre en correspondance entre des points détectés, nous calculons les descripteurs pour ces points. Parmi plusieurs descripteurs, le descripteur SIFT [Lowe 2004] a prouvé son efficacité pour la mise en correspondance entre des images [Mikolajczyk 2005].

Pour calculer le vecteur de descripteurs, l'angle et le facteur d'échelle servent à délimiter une zone elliptique autour du point. La normalisation ramène alors cette zone dans une orientation standard et sous la forme d'un disque. Le descripteur est alors calculé dans le disque. Cette disque est divisée en 16×16 sous-régions. Cette région est divisée en 16×16 sous-régions. Pour chaque bloc de 4×4 sous-régions, un histogramme de 8 éléments, correspondant à 8 orientations, est créé. La valeur de chaque élément est la somme des magnitudes du gradient des pixels ayant l'orientation correspondante. Le vecteur de descripteurs en effet contient 128 éléments. Pour un point d'intérêt p_i , le descripteur SIFT est $d_{ij} | j = 1, \dots, 128$. L'extraction du descripteur SIFT est illustrée dans la figure 4.10.

4.3.4 Discussions

Pour un objet, les couleurs dominantes, les matrices de covariance, les histogrammes des contours, les points d'intérêts sont extraits sur ses blobs représentatifs.

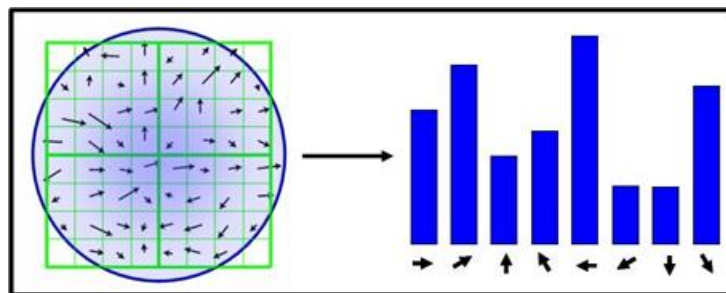


FIG. 4.9 – Identification des orientations pour chacun des points d'intérêts se base sur les maximums de l'histogramme de l'orientation ([Lowe 2004]).

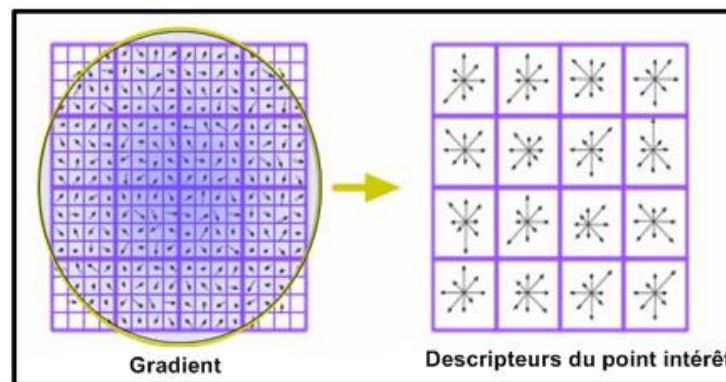


FIG. 4.10 – Région divisée en 16×16 sous-régions. Pour chaque bloc de 4×4 sous-régions, un histogramme de 8 éléments, correspondant à 8 orientations est créé ([Lowe 2004]).

Les couleurs dominantes et les matrices de covariance tiennent compte d'informations de la couleur alors que les histogrammes des contours et les points d'intérêts travaillent sur des images en niveau de gris. La différence entre les matrices de covariance et les couleurs dominantes est que les matrices de covariance considèrent la disposition des couleurs. Les couleurs dominantes et l'histogramme global des contours sont des descripteurs globaux. Les matrices de covariance, les points d'intérêt et d'autres histogrammes des contours (local, semi-local) sont des descripteurs locaux. Selon le détecteur de points d'intérêt choisi, les points d'intérêt peuvent correspondre aux coins ou aux régions.

Concernant la taille, les matrices de covariance et les histogrammes des contours ont une taille fixe. Le nombre de couleurs dominantes et celui de points d'intérêt varient en fonction du contenu de l'image.

La performance des descripteurs dépend de la caractéristique de l'image de requête et celle de la base de données. Dans le chapitre 6, une comparaison quantitative de ces descripteurs en indexation et recherche de vidéos de vidéosurveillance

est fournie. Cette comparaison nous montre des pistes afin de choisir a posteriori le descripteur approprié.

4.4 Extraction de descripteurs temporels

La trajectoire est un descripteur informatif des objets. Afin de permettre de mettre en correspondance les objets en se basant sur leurs trajectoires, il faut avoir (1) une représentation des trajectoires (2) une mesure de similarité. Une trajectoire peut être représentée soit par ses points de début et de fin soit par sa forme. La représentation d'une trajectoire par ses points de début et de fin permet de déterminer des changements de zones des objets. Elle ne décrit cependant pas la façon de se déplacer des objets. La représentation d'une trajectoire par sa forme consiste à déterminer la forme de la trajectoire considérée comme une série temporelle. Naftel et al. [Naftel 2006] ont approché les trajectoires en appliquant les polynômes de Tchebychev. Les trajectoires sont donc représentées par des coefficients obtenus à partir de cette approximation. Les trajectoires peuvent être également représentées par leurs coefficients obtenus par l'analyse en composantes principales (ACP) [Bashir 2007]. Chen et al. dans [Chen 2004], [Chen 2005] ont proposé deux façons de représenter une trajectoire associée à deux mesures de similarité. L'une est au niveau numérique et l'autre est au niveau symbolique. Au niveau numérique, la représentation s'appuie sur la direction et l'incrément relatif de distance parcourue entre deux positions consécutives tandis qu'au niveau symbolique, un espace identifié par la direction et l'incrément est divisé en sous-régions qui sont ensuite assignées par les symboles. Une trajectoire est donc transformée en une séquence de symboles. Certaines approches cherchent à identifier des activités en se basant sur les trajectoires des objets. De telles approches [Naftel 2006], [Foresti 2002] regroupent tout d'abord les trajectoires en classes, ensuite lient chacune des classes à une activité.

Dans notre travail de thèse, nous proposons une méthode de représentation des trajectoires en se basant sur le travail de Chen et al. [Chen 2004], [Chen 2005]. Au lieu d'utiliser tous les points des trajectoires, nous n'employons que les points de contrôle détectés de la trajectoire. La représentation de Chen et al. s'applique à ces points.

4.4.1 Représentation des trajectoires

L'algorithme de représentation des trajectoires comporte trois étapes : (1) la détection des points de contrôle, (2) la représentation des trajectoires au niveau numérique et (3) la représentation des trajectoires au niveau symbolique.

4.4.1.1 Détection des points de contrôle

Les points de contrôle d'une trajectoire sont les points de la trajectoire qui permettent d'approcher la forme de la trajectoire à partir de ces points. La détection des points de contrôle a été proposée par Chetverikov [Chetverikov 2003], [Hsieh 2006].

Algorithme 6 : La détection les points de contrôle de la trajectoire T **Input :**

$T = [(x_1, y_1), \dots, (x_n, y_n)]$: la trajectoire
 $d_{min}, d_{max}, T_\alpha$: les seuils

Output :

$T^p = [(x_1^p, y_1^p), \dots, (x_m^p, y_m^p)]$ où $m \leq n$

begin

```

1   for chacun des points  $p$  do
2       while trouver deux points  $p^+$  et  $p^-$  qui vérifient le critère (équation 4.18)
3       calculer l'angle  $\alpha(p)$  de  $p$  par l'équation 4.19
4       identifier l'angle étant le plus petit  $\alpha'(p)$ 
5       if  $\alpha'(p)$  est supérieur à  $T_\alpha$  then
6           rajouter  $p$  dans  $T^p$ 
7       end
8   for chacune des paires de deux points de  $T^p$  do
9       if il existe deux points qui sont proches (cf. équation 4.20)
10          enlever le point ayant l'angle  $\alpha$  le plus grand de  $T^p$ 
11       end
12  end
end

```

Pour chacun des points p , si l'on trouve deux points p^+ et p^- qui se situent de deux côtés de p et vérifient le critère défini par l'équation 4.18, l'angle au point p est défini par l'équation 4.19.

$$d_{min} \leq \|p - p^+\| \leq d_{max} \text{ et } d_{min} \leq \|p - p^-\| \leq d_{max} \quad (4.18)$$

où d_{min} et d_{max} sont deux seuils, $\|p - p^+\|$ et $\|p - p^-\|$ sont des distances entre p et p^+ et entre p et p^- .

$$\alpha(p) = \cos^{-1} \frac{\|p - p^+\|^2 + \|p - p^-\|^2 - \|p^+ - p^-\|^2}{2\|p - p^+\|\|p - p^-\|} \quad (4.19)$$

Deux points de contrôle sont proches :

$$\text{proche}(p_1, p_2) = \begin{cases} \text{vrai} & \text{si } \|p_1 - p_2\| \leq d_{min} \\ \text{faux} & \text{sinon} \end{cases} \quad (4.20)$$

La figure 4.11 illustre un exemple de détection des points de contrôle dans une trajectoire.

Dès lors, nous n'utilisons par défaut que les points de contrôle détectés pour une trajectoire. Dans le cas où tous les points de trajectoires sont utilisés, nous l'indiquons explicitement. Nous appelons T^{num} , T^{sym} les représentations de la trajectoire T aux niveaux numérique et sémantique.

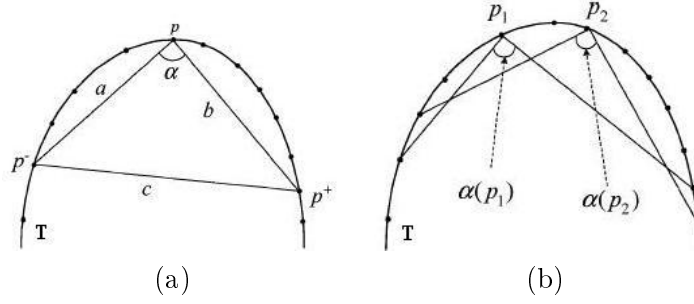


FIG. 4.11 – Détection des points de contrôle dans une trajectoire T : (a) un point de contrôle (p), deux points p^- et p^+ reliant à p sont satisfaits l'équation 4.18 ; (b) p_1 et p_2 are deux points détectés qui sont proches l'un de l'autre. Le point ayant l'angle le plus petit est gardé comme le point de contrôle ([Hsieh 2006]).

4.4.1.2 Représentation des trajectoires au niveau numérique

La représentation des trajectoires au niveau numérique consiste à transformer les trajectoires identifiées dans l'espace des positions absolues en celles identifiées dans l'espace des positions relatives [Chen 2004]. L'espace des positions relatives se détermine par la direction de mouvement et l'incrément relatif de distance parcourue entre deux positions consécutives.

Algorithme 7 : La représentation des trajectoires au niveau numérique

Input :

$$T = [(x_1, y_1), \dots, (x_m, y_m)]$$

Output :

$$T^{num} = [(\theta_1, \delta_1), \dots, (\theta_{m-1}, \delta_{m-1})]$$

begin

- 1 **for** chacune des paires de deux points consécutifs de T **do**
- 2 calculer la direction de mouvement (l'équation 4.21)
- 3 calculer l'incrément relatif (l'équation 4.22 et l'équation 4.23)
- 4 **end**

end

La direction de mouvement θ_i est définie par :

$$\theta_i = \begin{cases} \arctan \frac{y_{(i+1)} - y_{(i)}}{x_{(i+1)} - x_{(i)}} - \pi & \text{si } x_{(i+1)} - x_{(i)} < 0 \text{ et } y_{(i+1)} - y_{(i)} \leq 0 \\ \arctan \frac{y_{(i+1)} - y_{(i)}}{x_{(i+1)} - x_{(i)}} & \text{si } x_{(i+1)} - x_{(i)} \geq 0 \\ \arctan \frac{y_{(i+1)} - y_{(i)}}{x_{(i+1)} - x_{(i)}} + \pi & \text{si } x_{(i+1)} - x_{(i)} < 0 \text{ et } y_{(i+1)} - y_{(i)} > 0 \end{cases} \quad (4.21)$$

et le l'incrément δ_i relatif de distance parcourue entre position i et position $i + 1$:

$$\delta_i = \begin{cases} \frac{\sqrt{(y_{(i+1)} - y_{(i)})^2 + (x_{(i+1)} - x_{(i)})^2}}{TD(T)} & \text{si } TD(T) \neq 0 \\ 0 & \text{si } TD(T) = 0 \end{cases} \quad (4.22)$$

où $TD(T)$ est calculé par :

$$TD(T) = \sum_{1 \leq j \leq m-1} \sqrt{(y_{(j+1)} - y_j)^2 + (x_{(j+1)} - x_j)^2} \quad (4.23)$$

4.4.1.3 Représentation des trajectoires au niveau symbolique

La représentation des trajectoires au niveau symbolique cherche à transformer la trajectoire en séquences de symboles. L'objectif de cette représentation est d'appliquer les techniques de mise en correspondance de séquences de symboles qui ont été proposées dans le domaine de recherche de textes à la mise en correspondance des trajectoires.

La valeur de la direction de mouvement (θ_i) varie de $-\pi$ à π tandis que l'incrément relatif (δ_i) peut prendre valeurs de 0 à 1. Un espace à deux dimensions (la direction de mouvement et l'incrément relatif) est formulé. Cet espace est divisé en $\frac{2\pi}{\varepsilon_{dir}} * \frac{1}{\varepsilon_{dis}}$ sous-régions SB_i dont la taille est $\varepsilon_{dir} \times \varepsilon_{dis}$. La sous-région SB_i est représentée dans l'espace (θ, δ) par le point en bas à gauche $(\theta_{bl,i}, \delta_{bl,i})$ et le point en haut à droite $(\theta_{ur,i}, \delta_{ur,i})$. Chacune des sous-régions est ensuite assignée à un symbole distinctif. La figure 4.12 illustre un exemple de cet espace. La représentation des trajectoires au niveau symbolique consiste à identifier le symbole propre à chacun des points de la trajectoire en se basant sur sa position dans l'espace. Elle est montrée par l'algorithme 8.

Algorithme 8 : La représentation des trajectoires au niveau symbolique

Input :

$$T^{num} = [(\theta_1, \delta_1), \dots, (\theta_{m-1}, \delta_{m-1})]$$

$\varepsilon_{dir}, \varepsilon_{dis}$: détermine la taille de la sous-région

Output :

$$T^{sym} = [A_1, \dots, A_{m-1}]$$

begin

- 1 diviser l'espace de 2 dimensions (θ, δ) en $\frac{2\pi}{\varepsilon_{dir}} * \frac{1}{\varepsilon_{dis}}$ sous-régions SB_i .
- 2 assigner un symbole distinctif A_i pour chacune de sous-régions SB_i
- 3 **for** chacun des points (θ_j, δ_j) de T^{num} **do**
- 4 identifier son propre symbole par l'équation 4.24
- 5 **end**

end

Un point (θ_j, δ_j) prend le symbole de la sous-région SB_i si :

$$\theta_{bl,i} \leq \theta_j < \theta_{ur,i} \text{ et } \delta_{bl,i} \leq \delta_j < \delta_{ur,i} \quad (4.24)$$

4.4.2 Discussions

Pour la représentation temporelle d'un objet dans le modèle de données, deux méthodes de représentation de trajectoire, l'une au niveau numérique et l'autre au

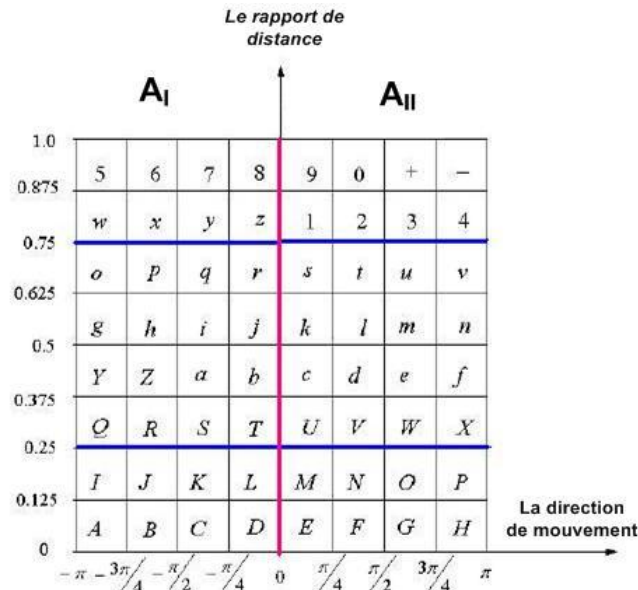


FIG. 4.12 – Exemple de l'espace défini par la direction de mouvement et l'incrément relatif de distance parcourue entre deux positions consécutives. Cet espace est divisé en 64 sous-régions, chacune des sous-régions est assignée à un symbole distinctif ([Chen 2004]).

niveau symbolique, sont présentées. Ces deux méthodes sont inspirées à partir du travail de Chen et al. [Chen 2004]. Au lieu d'employer tous les points de la trajectoire [Chen 2004], nous n'utilisons que ses points de contrôle détectés. Cela nous permet de réduire le temps de calcul et l'information à stocker. De plus, cette représentation est robuste à petites erreurs de détermination de trajectoire parce qu'elle utilise les points de contrôle permettant d'approcher la forme de la trajectoire. La représentation d'une trajectoire par une séquence de symboles possède un avantage : les techniques d'appariement de textes peuvent être appliquées.

4.5 Représentation d'objets mobiles

La représentation d'objets mobiles consiste à calculer les attributs *Blobs* et *Weights* pour les objets dans le modèle de données. Dans cette section, nous présentons la représentation des objets mobiles par leurs blobs représentatifs. Deux méthodes permettant de détecter des blobs représentatifs sont proposées.

4.5.1 Introduction

Les objets sont en général détectés et suivis pendant un intervalle de temps. À chacun des moments dans cet intervalle, un blob de l'objet est déterminé. Un objet possède, par conséquent, un ensemble de blobs. Si la détection et le suivi d'objets

sont fiables, la différence de contenu visuel des blobs consécutifs est habituellement petite car l'apparence de l'objet dans deux frames consécutifs ne change pas beaucoup. Le choix de blobs lorsque l'on fait la mise en correspondance entre des objets joue un rôle important. Nous donnons trois définitions que nous utilisons dans ce travail.

Un blob d'un objet est une région déterminée par la boîte englobante minimale dans le frame où l'objet est détecté

La boîte englobante minimale est déterminée par la détection d'objets dans les modules d'analyse vidéo.

Le blob pertinent pour un objet O appartenant à la classe C est le blob dans lequel une grande partie (le pourcentage de taille de la partie de l'objet présentée dans le blob et le taille de l'objet dans le frame où l'on détermine le blob est supérieur à 50) de l'objet O (dans le meilleur cas, c'est l'objet O entier) est présente.

Le blob non pertinent pour un objet O appartenant à la classe C est le blob dans lequel une petite partie (le pourcentage de taille de la partie de l'objet présentée dans le blob et le taille de l'objet dans le frame où l'on détermine le blob est inférieur à 50) de l'objet O est présente.

Il est à noter que si la classification d'objets est disponible dans les modules d'analyse vidéo, les classes sont des classes déterminées par la classification. Sinon, les objets appartiennent, par défaut, à la classe *Physical_objects*. Comme la détection des blobs représentatifs est automatique et qu'au moment où elle s'effectue, l'on ne sait pas quel objet est présent dans le blob. Cette tolérance nous permet de ne pas perdre d'informations. La présence des objets appartenant à la même classe que celui qu'on essaie de représenter est importante. Le blob pertinent pour un objet peut contenir les objets d'autres classes. L'utilisation des descripteurs locaux tels que les points d'intérêts nous permet de retrouver l'objet approprié dans un blob contenant plusieurs objets.

Soit $\{B_i\}$, $i \in (1, M)$ l'ensemble de blobs d'un objet déterminés par les modules d'analyse vidéo, les attributs *Blobs* et *Weights* de l'objet sont représentés par $\{(B_j^r, w_j^r)\}$, $j \in (1, K)$ où $K \ll M$, B_j^r est un blob représentatif et w_j^r est le degré d'importance du blob B_j^r . Les blobs représentatifs d'un objet doivent être pertinents et garder des aspects d'apparence de l'objet.

4.5.2 Détection des blobs représentatifs

La détection de blobs représentatifs d'un objet consiste à trouver l'ensemble minimal de blobs pertinents de l'objet pour que l'on puisse reproduire l'évolution de l'objet.

La détection de blobs représentatifs permet de :

- réduire l'information stockée : au lieu de stocker tous les blobs d'un objet, l'approche ne stocke que les blobs représentatifs ;
- réduire le temps de mise en correspondance entre des objets : au lieu de comparer tous les blobs des objets, la mise en correspondance entre des objets basée sur les blobs représentatifs permet de réduire le temps de calcul ;

- corriger dans certains cas l'erreur produite par les modules d'analyse car elle enlève les blobs non pertinents des objets avant de les mettre en correspondance.

La figure 4.13 illustre le cas où il est nécessaire de détecter les blobs représentatifs. Une personne est détectée et suivie pendant 24 frames. 24 blobs dont 11 blobs non pertinents sont déterminés. L'utilisation de blobs non pertinents peut conduire à une forte dégradation des résultats de la mise en correspondance entre des objets.

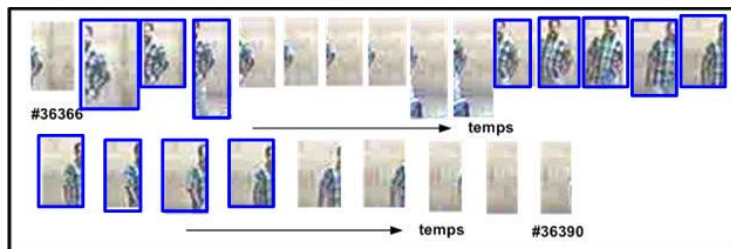


FIG. 4.13 – Exemple d'une personne détectée pendant 24 frames. 24 blobs dont 12 blobs pertinents (en bleu) et 12 blobs non pertinents sont déterminés.

Nous définissons deux mesures afin d'évaluer la détection de blobs représentatifs de l'objet O :

$$\begin{aligned} F(O) &= \frac{n \cdot 100\%}{N} \\ P(O) &= \frac{n_A \cdot 100\%}{n} \end{aligned} \quad (4.25)$$

où n est le nombre de blobs représentatifs déterminés pour l'objet O , N est le nombre de blobs de l'objet O , n_A est le nombre de blobs pertinents parmi n blobs. La mesure $F(O)$ exprime la capacité de réduire les informations à stocker et à calculer alors que la mesure $P(O)$ montre la capacité à corriger les erreurs produites par la détection et le suivi d'objets. Un algorithme de détection des blobs représentatifs est efficace s'il obtient une petite valeur de F et une grande valeur de P .

Nous proposons dans cette thèse deux méthodes de détection des blobs représentatifs, la détection des blobs représentatifs basée sur le changement d'apparence et celle basée sur le regroupement des blobs, qui seront détaillées dans les prochaines sections.

4.5.2.1 Détection des blobs représentatifs basée sur le changement d'apparence

La détection des blobs représentatifs par le changement d'apparence (cf. algorithme 9) se base sur l'observation suivante : le blob pertinent est le blob dont le contenu visuel est assez différent de ceux des blobs antérieurs. La différence des contenus des blobs se mesure par la distance des blobs en utilisant les descripteurs visuels. Les descripteurs visuels sont les descripteurs décrits dans la section précédente.

Soit $p(B_i, B_{i+1})$ la distance des descripteurs visuels de blob B_i et B_{i+1} , le changement de contenu visuel est défini :

$$\text{changement}(B_i, B_{i+1}) = \begin{cases} \text{oui} & \text{si} \begin{cases} p(B_{i-1}, B_i) \geq \theta \text{ et } p(B_i, B_{i+1}) < \theta \\ \text{ou} \\ p(B_{i-1}, B_i) < \theta \text{ et } p(B_i, B_{i+1}) \geq \theta \end{cases} \\ \text{non} & \text{sinon} \end{cases} \quad (4.26)$$

Le poids du blob représentatif :

$$w_j^r = \frac{M}{N} \quad (4.27)$$

où M est le nombre de blobs entre deux changements, N est le nombre total de blobs. La figure 4.14 montre le nombre des points d'intérêt de MSER appariés dans les blobs de la personne détectée. Les blobs représentatifs identifiés par cet algorithme sont montrés dans la figure 4.15. Cette méthode est simple. Elle est simplement une façon de résumer les blobs d'un objet. Elle permet de réduire l'information stockée et le temps de mise en correspondance entre des objets. L'algorithme 9 est assez efficace dans le cas où les résultats des modules d'analyse de vidéos sont fiables. Il a cependant deux limitations :

- les blobs représentatifs peuvent être les blobs pertinents ou non pertinents. L'approche ne permet pas d'enlever les blobs non pertinents ;
- l'approche est redondante s'il existe la reappartition de même apparence. Cette limitation augmente la mesure d'évaluation $F(O)$ (cf. équation 6.4) de l'algorithme.

4.5.2.2 Détection des blobs représentatifs basée sur le regroupement des blobs

Le travail de Ma et al. présenté dans le chapitre 2 [Ma 2007] est, à notre connaissance, le seul travail dédié à la détection des blobs représentatifs pour les objets mobiles dans les vidéos de vidéosurveillance. En résumé, cette méthode comprend trois étapes : (1) regrouper les blobs par le regroupement agglomératif par les matrices de covariance ; (2) enlever les groupes ayant peu d'éléments ; (3) déterminer un blob représentatif pour chacun des groupes (le blob est le plus similaire à tous les blobs du groupe). La méthode de Ma et al. assure d'avoir une petite valeur de $F(O)$ car elle prend un seul blob pour un groupe. La méthode de Ma et al. arrive à corriger les erreurs produites par la détection et le suivi d'objets si les erreurs sont présentes dans un petit nombre de blobs par rapport au nombre total des blobs d'un objet car la méthode de détection des blobs enlève des groupes ayant peu d'éléments. Il est pourtant impossible de corriger les erreurs qui sont présentes dans plusieurs blobs.

En se basant sur la méthode de Ma et al., nous proposons une méthode appelée la détection des blobs représentatifs basée sur le regroupement des blobs. La méthode proposée essaie de dépasser la limitation de la méthode de Ma et al. : corriger

Algorithme 9 : Détection des blocs représentatifs basée sur le changement d'apparence

Input :

$\{B_i\}, i \in 1, N$: l'ensemble de N blocs pour un objet,
 θ : un seuil déterminant le changement
 d : le descripteur choisi

Output :

$\{(B_j^r, w_j^r)\}, j \in 1, K, K \leq N$: l'ensemble de blocs représentatifs
 et leurs poids

begin
for chaque blob **do**

1 extraire le descripteur choisi

end
for deux blocs consécutifs B_i et B_{i+1} **do**

 2 calculer la distance des blocs par le descripteur choisi $p(B_i, B_{i+1})$

3 détecter le changement selon l'équation 4.26

4 déterminer le blob représentatif entre deux changements

5 calculer le poids du blob représentatif (cf. équation 4.27)

end
end

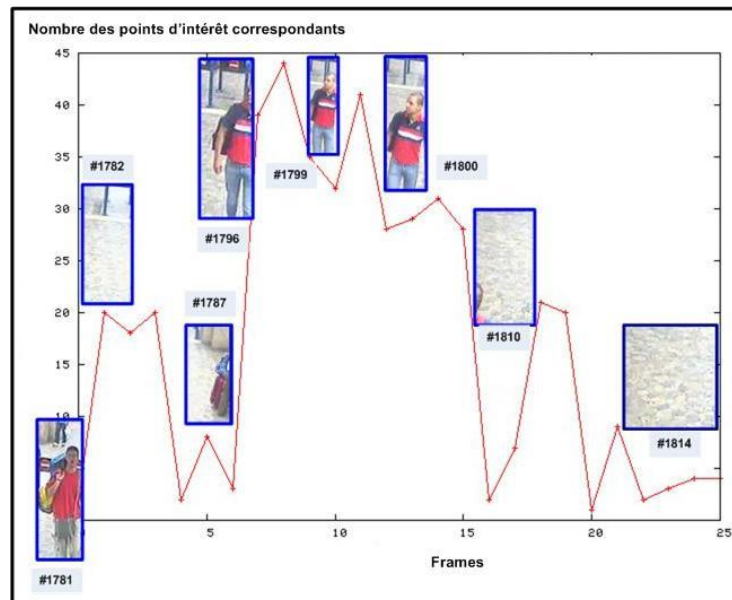


FIG. 4.14 – 26 blocs d'une personne détectée et les points d'intérêt de MSER appariés pour 25 frames consécutifs.



FIG. 4.15 – Blobs représentatifs associés à leurs poids détectés par la méthode basée sur le changement d'apparence.

les erreurs qui sont présentes dans un grand nombre de blobs. Pour cela, la méthode doit pouvoir d'enlever les blobs non pertinents. Cela peut être obtenu par une classification des blobs en blobs qui contiennent l'objet d'intérêt (nous l'appelons le blob avec objets) ou blobs qui ne contiennent pas l'objet d'intérêt (nous l'appelons le blob sans objet). L'approche proposée donc peut augmenter la valeur de mesure $P(O)$ (cf. équation 6.4).

La détection des blobs représentatifs basée sur le regroupement des blobs est montrée par l'algorithme 10. Elle comprend 5 étapes dont les 3 dernières sont reprises de la méthode de Ma et al. Nous choisissons les machines à vecteurs de support (Support Vector Machine - SVM) pour la classification des blobs avec objets et des blobs sans objet dans l'étape 1. Parmi les descripteurs présentés auparavant, nous employons les histogrammes des contours.

Les SVM permettent de trouver une surface qui sépare au mieux les classes de données en maximisant la marge entre ces classes. Il s'agit de minimiser le majorant de l'erreur réelle. Intuitivement ce classifieur est un hyperplan qui maximise la marge d'erreur, qui est la somme des distances entre l'hyperplan et les exemples positifs et négatifs les plus proches de cet hyperplan.

Si $\{x_1, \dots, x_n\}$ est l'ensemble des données et $y_i \in \{1, -1\}$ la classe de x_i , la frontière de décision devrait mener à classer correctement tous les points : $y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i$

Maximiser la marge revient donc à minimiser $\frac{1}{2}w^T w + C \sum_{i=1}^l \xi_i$. Il faut déterminer w et b et ξ qui minimisent :

$$\min_{w,b,\xi} \frac{1}{2}w^T w + C \sum_{i=1}^l \xi_i \quad (4.28)$$

sous les contraintes (hyperplan séparateur) :

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \quad (4.29)$$

où $\xi_i \geq 0$

Dans le cas où les données ne peuvent pas être séparées par une fonction linéaire, une non-linéarité peut être introduite grâce à l'utilisation d'une fonction symétrique positive appelée fonction de noyau K (cf. équation 4.30).

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \quad (4.30)$$

Le noyau choisi dans ce travail est le noyau à fonction à base radiale (radial basis function (RBF)) :

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \quad \gamma > 0 \quad (4.31)$$

Soit $B = \{B_i\}$, $i \in 1, N$ est l'ensemble de N blobs pour un objet. \mathcal{B} est l'ensemble des B de tous les objets appartenant à la même classe dans une vidéo. $B^r = \{B_j^r\}$, $j \in 1, K$ est l'ensemble de K blobs représentatifs détectés pour un objet. \mathcal{B}^r est l'ensemble des B^r de tous les objets appartenant à la même classe dans une vidéo.

Sachant que la première phase de l'algorithme 10 s'effectue pour tous les ensembles B des objets appartenant à la même classe dans une vidéo tandis que la deuxième phase travaille sur un ensemble B à la fois.

Algorithme 10 : Détection des blobs représentatifs par le regroupement des blobs

Input :

\mathcal{B} : l'ensemble des blobs B

Output :

\mathcal{B}^r : l'ensemble des blobs représentatifs B^r

begin

- 1 classifier les blobs en blobs avec objets et blobs sans objet de \mathcal{B} par les SVM et les histogramme des contours.
- 2 enlever les blobs sans objet de \mathcal{B}
- for** chacun B de \mathcal{B} **do**
- 3 regrouper les blobs avec objets (matrices de covariance + regroupement agglomératif)
- 4 effacer les groupes contenant peu d'éléments
- 5 déterminer le blob représentatif et son poids pour chaque groupe
- end**

end

4.5.3 Discussions

Nous proposons deux méthodes de détection des blobs représentatifs : l'une basée sur le changement d'apparence et l'autre basée sur le regroupement des blobs pour les objets dans les vidéos de vidéosurveillance. Les deux méthodes proposées sont

générales dans la mesure où elles peuvent travailler avec plusieurs descripteurs d'apparence. Dans le chapitre 6, nous présentons des résultats de méthode de détection des blobs représentatifs par le changement d'apparence avec des descripteurs d'apparence différents. Pour la détection des blobs représentatifs par le regroupement des blobs, afin de la comparer avec celle de Ma et al. [Ma 2007], nous utilisons aussi les matrices de covariance et le regroupement agglomératif. Cependant, d'autres descripteurs d'apparence et d'autres méthodes de regroupement peuvent être appliqués.

La méthode de détection des blobs représentatifs basée sur le changement d'apparence (cf. algorithme 9) est efficace si on a une bonne détection et un bon suivi d'objets dans des modules d'analyse vidéos. Elle permet de garder des aspects visuels différents de l'objet. De plus, elle effectue en deux modes : en ligne et hors ligne car il ne faut pas avoir tous les blobs pour commencer à effectuer la méthode. Cette méthode est rapide. Pour un objet comprenant N blobs, elle fait $N - 1$ comparaisons des blobs consécutifs.

La méthode de détection des blobs représentatifs basée sur le regroupement des blobs (cf. algorithme 10) est robuste à l'imperfection de la détection et du suivi d'objets. Elle est cependant hors ligne car elle demande de savoir tous les blobs d'un objet avant d'effectuer la détection des blobs. De plus, cette méthode a besoin d'exemples annotés pour entraîner les SVM. Elle prend du temps pour classifier des blobs en blobs avec objets et blobs sans objet et le regroupement des blobs. Pour un objet contenant N blobs, elle demande $\frac{N*(N-1)}{2}$ comparaisons des blobs pour le regroupement.

Nous employons toutes les deux méthodes de détection des blobs représentatifs proposées en indexation et recherche de vidéos pour la vidéosurveillance.

4.6 Conclusion

Dans ce chapitre, un modèle de données pour l'indexation et la recherche de vidéos de vidéosurveillance est présenté. Ce modèle de données comprend deux concepts abstraits principaux : objets et événements. Les concepts objets et événements nous permettent de caractériser les objets, leurs évolutions et leurs interactions dans les scènes observées par les caméras.

Tous les attributs de l'événement et certains attributs de l'objet sont directement déterminés par des modules d'analyse vidéo. Deux méthodes de détection des blobs représentatifs visent à déterminer les attributs *Blobs* et *Weights* pour un objet. Elles détectent, pour un objet, un ensemble des blobs représentatifs et leurs poids.

La représentation de l'objet par les descripteurs d'apparence R_{ap} est faite en extrayant des descripteurs d'apparence sur les blobs représentatifs. Les descripteurs d'apparence choisis dans cette thèse sont les couleurs dominantes, les matrices de covariance, les histogrammes de contours et les points d'intérêt.

Pour la représentation de l'objet par les descripteurs temporels R_t , nous présentons deux représentations de la trajectoire au niveau numérique et symbolique en utilisant ses points de contrôle détectés.

Recherche de vidéos de vidéosurveillance

5.1 Introduction

Ce chapitre décrit la phase de recherche de vidéos de notre approche. Elle comporte trois tâches principales : la formulation de requêtes, la mise en correspondance des éléments indexés et des requêtes et le retour de pertinence. La formulation de requêtes consiste à fournir à l'utilisateur un outil pour qu'il puisse exprimer ses propres requêtes. Contrairement aux approches mentionnées dans le chapitre 2 (sauf le travail de Ghanem et al [Ghanem 2004]) qui limitent les requêtes possibles pour les utilisateurs, nous montrons dans cette thèse que la formulation de requêtes permet à l'utilisateur de définir de nouveaux événements à partir des événements reconnus. La mise en correspondance consiste à mesurer la similarité entre des éléments indexés et la requête. Le retour de pertinence consiste d'une part à apprendre à partir des retours de l'utilisateur et d'autre part à lui rendre de nouveaux résultats de recherche. Nous analysons tout d'abord les scénarios de recherche de vidéos de vidéosurveillance (voir section 5.2). Nous présentons ensuite la mise en correspondance par des descripteurs d'apparence, des descripteurs temporels et des intervalles de temps (voir sections 5.4 et 5.5). Afin de fournir à l'utilisateur un outil qui lui permet d'exprimer ses propres requêtes, nous proposons un langage de requêtes (voir section 5.7). La dernière section présente le retour de pertinence (voir section 5.8)

5.2 Scénarios de recherche

L'indexation et la recherche de vidéos de vidéosurveillance ont deux types principaux d'utilisateur : les personnels de sécurité et des développeurs du système. Les requêtes de recherche de vidéos de vidéosurveillance peuvent être à différents niveaux sémantiques. Nous analysons les scénarios de recherche selon les niveaux sémantiques des requêtes :

- au niveau images : l'utilisateur a une imagerie, il a l'intention de savoir si des objets qui sont semblables à l'imagerie apparaissent dans une scène ;
- au niveau objets : l'utilisateur connaît les informations d'un objet dans une scène. Il veut trouver les informations de cet objet dans d'autres scènes ou des objets quelconques qui vérifient un critère de l'apparence ou du temps avec l'objet qu'il connaît ;

- au niveau événements : l'utilisateur veut savoir s'il existe un événement composé à partir des événements reconnus dans la scène. L'événement composé est une série des événements qui vérifient un ensemble de relations temporelles entre eux ;
- à multiple niveaux : la requête à ce niveau est une requête composée à partir des requêtes aux trois niveaux précédemment mentionnés.

Dans le chapitre 6, nous montrons l'expression de ces requêtes dans le langage proposé selon des niveaux différents de l'analyse vidéos.

5.3 Mise en correspondance des éléments indexés

En indexation et recherche d'information, une fois l'indexation faite, la performance des approches dépend fortement de la mise en correspondance entre des éléments en se basant sur les descripteurs extraits dans la phase d'indexation.

La mise en correspondance entre des éléments consiste à définir comment l'on compare ces éléments et comment ces éléments se ressemblent. Autrement dit, la mise en correspondance détermine quel type de descripteurs et quelle mesure de similarité seront utilisés et calcule la valeur de cette mesure sur les éléments en question.

Dans ce travail de thèse, nous distinguons trois mises en correspondance : celle entre des objets par les descripteurs d'apparence, celle par leurs trajectoires et celle entre des objets et des événements par leurs relations temporelles. Comme nous le présentons dans le chapitre 4, un objet peut posséder plusieurs blobs. Afin de déterminer la mise en correspondance entre des objets par leurs descripteurs d'apparence, nous présentons tout d'abord celle entre des blobs. La mise en correspondance entre des objets est définie à partir de celle entre leurs blobs.

Notons \mathcal{F}_B l'ensemble des mises en correspondance entre des blobs en s'appuyant sur les descripteurs d'apparence, \mathcal{F}_O l'ensemble des mises en correspondance entre des objets basées sur les descripteurs d'apparence, \mathcal{F}_T l'ensemble des mises en correspondance entre des objets en s'appuyant sur leurs trajectoires, et \mathcal{F}_{Temp} l'ensemble des relations temporelles des objets et des événements.

Les sections suivantes visent à déterminer les éléments des ensembles \mathcal{F}_B , \mathcal{F}_O (cf. section 5.4), \mathcal{F}_T (cf. section 5.5) et \mathcal{F}_{Temp} (cf. section 5.6). Le langage de requêtes est créé en se basant sur ces éléments.

5.4 Mise en correspondance entre des objets basée sur les descripteurs d'apparence

Cette section est dédiée à définir l'ensemble des mises en correspondance entre des blobs (\mathcal{F}_B) et l'ensemble des mises en correspondance entre des objets (\mathcal{F}_O) en s'appuyant sur les descripteurs d'apparence. Il est à noter que pour la mise en correspondance entre des blobs, nous réutilisons les mises en correspondance propres

de chaque descripteur. Notre contribution est une nouvelle mise en correspondance au niveau objets en se basant sur celles au niveau blobs (images).

5.4.1 Mise en correspondance entre des blobs basée sur les descripteurs d'apparence

Avant de présenter les mises en correspondance entre des blobs par descripteurs d'apparence, nous définissons quelques notions :

Soit f_B un élément de l'ensemble \mathcal{F}_B . L'élément f_B devient :

- f_B^{DC} si les descripteurs utilisés sont les couleurs dominantes ;
- f_B^{CM} si les descripteurs utilisés sont les matrices de covariance ;
- f_B^{EH} si les descripteurs utilisés sont les histogrammes de contours ;
- f_B^{IP} si les descripteurs utilisés sont les points d'intérêt.

Soit Q, P deux blobs, la mise en correspondance entre Q et P par les couleurs dominantes, les matrices de covariance et les histogrammes des contours et les points d'intérêt sont $f_B^{DC}(B, Q)$, $f_B^{CM}(Q, P)$, $f_B^{EH}(Q, P)$, et $f_B^{IP}(Q, P)$ respectivement.

5.4.1.1 Mise en correspondance entre des blobs par les couleurs dominantes

Soit $DC(Q), DC(P)$ les couleurs dominantes extraites sur Q et P en appliquant l'algorithme 1 (cf. page 77-78), $DC(Q), DC(P)$ peuvent être exprimées :

$$DC(Q) = \{(c_i^Q, p_i^Q)\}, i = 1, \dots, N^Q \text{ et } DC(P) = \{(c_j^P, p_j^P)\}, j = 1, \dots, N^P$$

où c_i, p_i la couleur dominante et son poids qui sont déterminés par l'algorithme 1.

La mise en correspondance entre des blobs par les couleurs dominantes est définie par [Deng 2001] :

$$f_B^{DC}(P, Q) : (\mathcal{D}_{DC} \times \mathcal{D}_{DC}) \rightarrow \mathcal{R} \stackrel{\text{def}}{=} \sqrt{\sum_{i=1}^{N^Q} (p_i^Q)^2 + \sum_{j=1}^{N^P} (p_j^P)^2 - \sum_{i=1}^{N^Q} \sum_{j=1}^{N^P} 2a_{ij} p_i^Q p_j^P} \quad (5.1)$$

tel que :

$$a_{ij} = \begin{cases} 1 - \frac{d_{ij}}{d_{max}} & d_{ij} \leq T_d \\ 0 & d_{ij} > T_d \end{cases} \quad (5.2)$$

où $d_{ij} = \|c_i^Q - c_j^P\|$ et $d_{max} = \alpha * T_d$. La valeur de α est 1.2 tandis que T_d est la distance maximale de deux couleurs dominantes. La valeur de T_d est déterminée avant de l'extraction des couleurs dominantes. Plus la valeur de T_d est petite, plus le nombre de couleurs dominantes détectées est élevé.

5.4.1.2 Mise en correspondance entre des blobs par les matrices de covariance

Soit $CM(Q)$, $CM(P)$ deux matrices de covariance extraites sur Q et P , La mise en correspondance entre des blobs par les matrices de covariance est définie [Forstner 1999] par :

$$f_B^{CM}(P, Q) : (\mathcal{D}_{CM} \times \mathcal{D}_{CM}) \rightarrow \mathcal{R} \stackrel{\text{def}}{=} \sqrt{\sum_{k=1}^d \ln^2 \alpha_k(CM(Q), CM(P))} \quad (5.3)$$

où $\alpha_k(CM(Q), CM(P))$ sont des valeurs propres généralisées de $CM(Q)$ et $CM(P)$. Les valeur propres généralisées sont déterminées par :

$$\alpha_k CM(Q)u_k - CM(P)u_k = 0 \quad \forall k \quad (5.4)$$

5.4.1.3 Mise en correspondance entre des blobs par les histogrammes des contours

Soit $EH(Q)$, $EH(P)$ les histogrammes des contours de Q and P déterminés par l'algorithme 2. Il existe quatre types d'histogrammes des contours : l'histogramme local, l'histogramme semi-local, l'histogramme global et l'histogramme composé. Nous pouvons les préciser :

$EH^{local}(Q) = \{H_i^Q\}$ et $EH^{local}(P) = \{H_i^P\}$ où $i = 1, \dots, 80$: les histogrammes locaux de Q et P .

$EH^{semi-local}(Q) = \{H_i^Q\}$ et $EH^{semi-local}(P) = \{H_i^P\}$, $i = 1, \dots, 65$: les histogrammes semi-locaux de Q et P .

$EH^{global}(Q) = \{H_i^Q\}$ et $EH^{global}(P) = \{H_i^P\}$, $i = 1, \dots, 5$: les histogrammes globaux de Q et P .

$EH^{com}(Q) = \{H_i^Q\}$ et $EH^{com}(P) = \{H_i^P\}$, $i = 1, \dots, 150$: les histogrammes composés Q et P .

Correspondant au type de l'histogramme, nous déterminons :

$$f_B^{EH} = \{f_B^{EH,local}, f_B^{EH,semi-local}, f_B^{EH,global}, f_B^{EH,com}\}.$$

La mise en correspondance entre des blobs par leurs histogrammes locaux :

$$f_B^{EH,local}(P, Q) : (\mathcal{D}_{EH^{local}} \times \mathcal{D}_{EH^{local}}) \rightarrow \mathcal{R} \stackrel{\text{def}}{=} \sum_{i=1}^{80} \|H_i^Q - H_i^P\|_2 \quad (5.5)$$

La mise en correspondance entre des blobs par leurs histogrammes semi-locaux :

$$f_B^{EH,global}(P, Q) : (\mathcal{D}_{EH^{global}} \times \mathcal{D}_{EH^{global}}) \rightarrow \mathcal{R} \stackrel{\text{def}}{=} \sum_{i=1}^5 \|H_i^Q - H_i^P\|_2 \quad (5.6)$$

La mise en correspondance des blobs par leurs histogrammes globaux :

$$f_B^{EH,semi-local}(P, Q) : (\mathcal{D}_{EH^{semi-local}} \times \mathcal{D}_{EH^{semi-local}}) \rightarrow \mathcal{R} \stackrel{\text{def}}{=} \sum_{i=1}^{65} \|H_i^Q - H_i^P\|_2 \quad (5.7)$$

La mise en correspondance entre des blobs par leurs histogrammes composés :

$$f_B^{EH,com}(P, Q) : (\mathcal{D}_{EHcom} \times \mathcal{D}_{EHcom}) \rightarrow \mathcal{R} \stackrel{\text{def}}{=} \sum_{i=1}^{150} \|H_i^Q - H_i^P\|_2 \quad (5.8)$$

où $\|H_i^Q, H_i^P\|_2$ est la norme L2.

5.4.1.4 Mise en correspondance entre des blobs par les points d'intérêt

Soit $IP(Q) = \{(p_i^Q, d_i^Q)\} | i = 1, \dots, N^Q$, $IP(P) = \{(p_j^P, d_j^P)\} | j = 1, \dots, N^P$ deux ensembles de points d'intérêt associés au descripteur SIFT détectés sur les images Q et P. Les points d'intérêt sont les points de Harris Affine, MSER et DoG. Le descripteur SIFT est également extrait sur ces points.

$$f_B^{IP}(P, Q) : (\mathcal{D}_{IP} \times \mathcal{D}_{IP}) \rightarrow \mathcal{R} \stackrel{\text{def}}{=} 1 - \frac{nb_corres(Q, P)}{N^Q} \quad (5.9)$$

où $nb_corres(Q, P)$ est le nombre de points correspondants de Q et P.

$$nb_corres(Q, P) = \sum_{i=1}^{N^Q} corres((p_i, d_i), IP(P)) \quad (5.10)$$

$corres((p_i, d_i), IP(P))$ détermine si un point d'intérêt (p_i, d_i) de Q est correspondant à l'ensemble des points d'intérêt de P. Nous employons la méthode de Lowe [Lowe 2004] : pour un point d'intérêt (p_i, d_i) de Q, la méthode trouve deux points de P les plus proches du point d'intérêt (p_i, d_i) de Q (les distances Euclidiennes de leurs descripteurs SIFT sont les plus petites) et le rapport de deux distances est inférieur à un seuil.

5.4.2 Mise en correspondance entre des objets par les descripteurs d'apparence

5.4.2.1 Introduction

Comme nous l'expliquons dans le chapitre 4, l'apparence d'un objet évolue au fur et à mesure dans le temps, il est donc nécessaire d'utiliser un ensemble de ses blobs pour qu'on puisse représenter les différents aspects de son apparence. Les deux méthodes de détection des blobs représentatifs présentées dans le chapitre 4 nous permettent d'avoir cet ensemble. Dans l'état de l'art, nous avons montré le travail de Ma et al. [Ma 2007] qui est, à notre connaissance, le seul travail dédié à la mise en correspondance entre des objets en se basant sur celle entre leurs blobs. Pour cela, la distance de Hausdorff a été choisie. Étant donné deux objets Q, P, leurs blobs représentatifs sont : $B_Q = \{(B_{Q,i}^r, w_{Q,i}^r)\}, i \in (1, K_Q)$ et $\{(B_{P,j}^r, w_{P,j}^r)\}, j \in (1, K_P)$. La distance $d(Q, P)$ entre deux objets basée sur la distance de Hausdorff est définie par :

$$d(Q, P) = \max_{B_{Q,i}^r \in B_Q} \min_{B_{P,j}^r \in B_P} (d(B_{Q,i}^r, B_{P,j}^r)) \quad (5.11)$$

où $d(B_{Q,i}^r, B_{P,j}^r)$ est la distance entre deux matrices de covariance de deux blobs. Cette distance n'est cependant pas appropriée à la mise en correspondance entre des objets de vidéosurveillance car elle n'est pas robuste à l'imperfection de la détection et du suivi d'objets. Si deux ensembles de blobs représentatifs des objets Q et P qui sont parfaitement appariés sauf un seul blob de Q qui est semblable à aucun blob de P , la distance de Hausdorff sera déterminée par cette paire de blobs. Ce problème est fréquemment rencontré en indexation et recherche de vidéos de vidéosurveillance en raison de l'imperfection de la détection et du suivi d'objets. De plus, la méthode de Ma et al. n'emploie pas de poids associé à chaque blob.

Afin de travailler avec les bases bruitées, la mise en correspondance entre des objets doit avoir les trois caractéristiques suivantes :

- elle peut calculer la distance entre des objets ayant un nombre différent de blobs ;
- elle prend en compte la distance entre des blobs par les descripteurs d'apparence ;
- elle permet d'apparier partiellement des objets.

La distance EMD (Earth Movers Distance) possède ces trois caractéristiques. Il est à noter que cette distance a été utilisée en indexation et recherche d'images [Rubner 1998] et de vidéos télévisées [Peng 2005]. Cependant, on ne peut pas appliquer exactement le travail présenté en indexation et recherche d'images et de vidéos télévisées à l'indexation et à la recherche de vidéos pour la vidéosurveillance. En indexation et recherche d'images [Rubner 1998], une image est représentée par un ensemble des régions qui sont calculées par la segmentation d'images. Le nombre de régions n'est pas élevé. Il n'est donc pas nécessaire de détecter des régions représentatives. Le poids de chaque région est déterminé en fonction de sa taille. En indexation et recherche de vidéos télévisées [Peng 2005], la détection de frames clés est nécessaire. En effet, un segment est un ensemble de frames clés et une vidéo est un ensemble des plans de vidéo. La distance entre deux vidéos est la distance EMD entre deux ensembles de plans de vidéo. L'indexation et la recherche de vidéos pour la vidéosurveillance est plus fine que celles pour les vidéos télévisées : nous travaillons avec les objets mobiles au lieu d'avec les frames globaux. Nos contributions sont d'analyser comment cette distance est appliquée à l'indexation et à la recherche de vidéos de vidéosurveillance et de montrer quelle est la capacité de cette distance pour résoudre des problèmes dans ce domaine.

Nous présentons dans les prochaines sections la distance EMD et la mise en correspondance entre des objets basée sur cette distance.

5.4.2.2 La distance EMD

La distance EMD modélise le problème de la même manière que celui rencontré sous le nom de problème des transports. Un ensemble de trous ayant chacun une certaine profondeur représente la première distribution, alors qu'un ensemble de tas de terre ayant chacun une certaine hauteur représente la deuxième distribution. La

distance EMD entre les deux, est le transport de terre le plus efficace qui puisse être trouvé en terme de travail à fournir, pour remplir les trous. Si on note $P = \{(p_1, w_{p_1}), \dots, (p_m, w_{p_m})\}$ la première distribution et $Q = \{(q_1, w_1), \dots, (q_n, w_{q_n})\}$ la deuxième distribution et $F = (f_{ij})$ le flux de terre à transporter de p_i vers q_j , d_{ij} le travail demandé de transporter une unité de terre de p_i et q_j , le programme linéaire devient :

$$\min_F \sum_{i=1}^m \sum_{j=1}^n f_{ij} d_{ij} \quad (5.12)$$

sous les contraintes :

$$f_{ij} \geq 0, 1 \leq i \leq m, 1 \leq j \leq n. \quad (5.13)$$

$$\sum_{j=1}^n f_{ij} \leq w_{p_i}, 1 \leq i \leq m \quad (5.14)$$

$$\sum_{i=1}^m f_{ij} \leq w_{q_j}, 1 \leq j \leq n \quad (5.15)$$

$$\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min\left(\sum_{i=1}^m w_{p_i}, \sum_{j=1}^n w_{q_j}\right) \quad (5.16)$$

La première contrainte impose un transport uniquement de P vers Q . Les deux contraintes suivantes limitent les transports à la quantité disponible. Quand le flux F^* optimal est trouvé, la distance EMD entre P et Q est définie comme le travail normalisé par le flux optimal total :

$$EMD(P, Q) = \frac{\sum_{i=1}^m \sum_{j=1}^n f_{ij}^* d_{ij}}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}^*} \quad (5.17)$$

Dans cette thèse, nous appliquons la méthode proposée par Hillier et al. [Hillier 1990] afin de calculer la distance EMD. Le temps de calcul est $O(n^3 \log n)$ selon Rubner et al. [Rubner 2000] où n est le nombre d'éléments de la distribution.

5.4.2.3 La mise en correspondance entre des objets basée sur la distance EMD

Grâce au travail présenté dans le chapitre 4, nous obtenons la représentation d'un objet par K blobs représentatifs : $\{(B_j^r, w_j^r)\}$ où $j \in (1, K)$. Étant donné deux objets Q, P , leurs blobs sont :

$$R_{Q,ap} = \{(B_{Q,i}^r, w_{Q,i}^r)\}, i \in (1, K_Q) \text{ et } R_{P,ap} = \{(B_{P,j}^r, w_{P,j}^r)\}, j \in (1, K_P)$$

Soit f_O un élément de l'ensemble des mises en correspondance entre des objets par les descripteurs d'apparence \mathcal{F}_O , il est défini en appliquant la distance EMD :

$$f_O(Q, P) : (\mathcal{D}_{R_{ap}} \times \mathcal{D}_{R_{ap}}) \rightarrow \mathcal{R} \stackrel{\text{def}}{=} EMD(R_{Q,ap}, R_{P,ap}) \quad (5.18)$$

où la distance $EMD(R_{Q,ap}, R_{P,ap})$ est définie par :

$$EMD(R_{Q,ap}, R_{P,ap}) = \frac{\sum_{i=1}^{K_Q} \sum_{j=1}^{K_P} f_{ij}^* f_B(B_{Q,i}, B_{P,j})}{\sum_{i=1}^{K_Q} \sum_{j=1}^{K_P} f_{ij}^*} \quad (5.19)$$

où $f_B(B_{Q,i}, B_{P,j})$ est la mise en correspondance entre $B_{Q,i}$ et $B_{P,j}$. Correspondant à quatre types de f_B qui sont déterminés dans la section 5.4.1, f_O inclut quatre types différents :

- la mise en correspondance par les couleurs dominantes f_O^{DC} : les descripteurs utilisés sont les couleurs dominantes ;
- la mise en correspondance par les matrices de covariances f_O^{CM} : les descripteurs utilisés sont les matrices de covariance ;
- la mise en correspondance par les histogrammes de contour f_O^{EH} : les descripteurs utilisés sont les histogrammes de contours ;
- la mise en correspondance par les points et régions d'intérêt f_O^{IP} : les descripteurs utilisés sont les points d'intérêt.

La mise en correspondance entre deux objets Q et P par les couleurs dominantes $f_O^{DC}(Q, P)$, par les matrices de covariances $f_O^{CM}(Q, P)$, par les histogrammes de contour $f_O^{EH}(Q, P)$ et par les points d'intérêt $f_O^{IP}(Q, P)$ sont déterminés par l'équation 5.19 en remplaçant $f_B(B_{Q,i}, B_{P,j})$ par $f_B^{DC}(B_{Q,i}, B_{P,j})$, $f_B^{CM}(B_{Q,i}, B_{P,j})$, $f_B^{EH}(B_{Q,i}, B_{P,j})$ et $f_B^{IP}(B_{Q,i}, B_{P,j})$ respectivement.

Afin de montrer la différence entre la méthode de Ma et al. et notre méthode pour la mise en correspondance entre des objets, nous donnons un exemple : la mise en correspondance entre deux objets dont les étiquettes sont 1064 et 1065 respectivement. Puisque la différence entre notre mise en correspondance et celle de Ma et al. est la distance utilisée, dans cette thèse, nous utilisons la distance EMD et notre méthode de mise en correspondance de façon interchangeable. La distance de Hausdorff et la méthode de mise en correspondance de Ma et al. sont également utilisées de façon interchangeable.

En appliquant la méthode de détection des blobs représentatifs, nous obtenons 4 et 5 blobs représentatifs pour l'objet 1064 et 1065 respectivement. Les blobs représentatifs avec les poids associés sont montrés dans la figure 5.1. Afin de mesurer la distance entre deux objets, nous calculons les distances de chacune des paires des blobs. Leurs distances par les matrices de covariance (cf. équation 5.3) sont données dans le tableau 5.1. La méthode de Ma et al. et notre méthode sont montrées (voir figure 5.2). Comme nous l'expliquons la distance de Hausdorff a deux limitations : la distance entre deux objets est décidée par celle de la paire de blobs dont leur distance est la plus grande ; elle n'emploie pas les poids des blobs. Il est à noter que le poids d'un blob exprime le degré d'importance de ce blob. Plus les poids sont grands, plus les blobs sont significatifs. La personne 1064 est mal détectée et suivie dans certains frames. Cela est exprimé par le blob représentatif 1 dont le poids est 0.052.

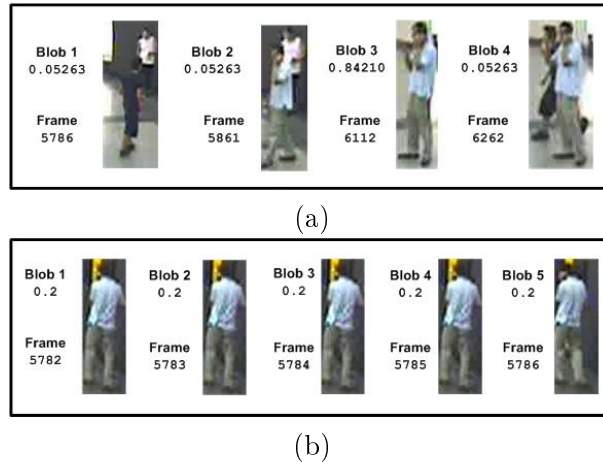


FIG. 5.1 – (a) blobs représentatifs de l’objet #1064; (b) et ceux de l’objet #1065. L’objet 1064 n’est pas bien détecté et suivi pendant certains frames. Le blob représentatif 1 n’est pas pertinent.

TAB. 5.1 – Distances entre des paires de blobs en utilisant les matrices de covariance, les colonnes sont les blobs de l’objet 1064, les lignes sont les blobs de l’objet 1065. La distance de Hausdorff entre deux objets est déterminée par la distance entre le blob 1 de l’objet 1064 et le blob 4 de l’objet 1065.

		Objet #1064				
		blob 1	blob 2	blob 3	blob 4	blob 5
Objet #1065	blob 1	3.873128	3.873128	3.873128	3.873128	3.361022
	blob 2	2.733361	2.733361	2.733361	2.733361	2.161412
	blob 3	2.142512	2.142512	2.142512	2.142512	1.879587
	blob 4	2.193842	2.193842	2.193842	2.193842	2.048116

L’exemple montre l’efficacité de la distance EMD en indexation et recherche de vidéos de vidéosurveillance. Une comparaison quantitative de performance de deux méthodes (notre méthode et celle de Ma et al.) est présentée dans le chapitre 6.

5.4.3 Discussions

Une nouvelle méthode de mise en correspondance entre des objets basée sur la distance EMD dans les vidéos de vidéosurveillance est introduite. La distance EMD a trois précieuses caractéristiques : les objets ayant un nombre différent d’aspects d’apparence peuvent être comparés ; les distances de descripteur d’apparence présentées dans l’état de l’art sont considérées ; le poids de chaque blob représentatif est explicitement prise en compte.

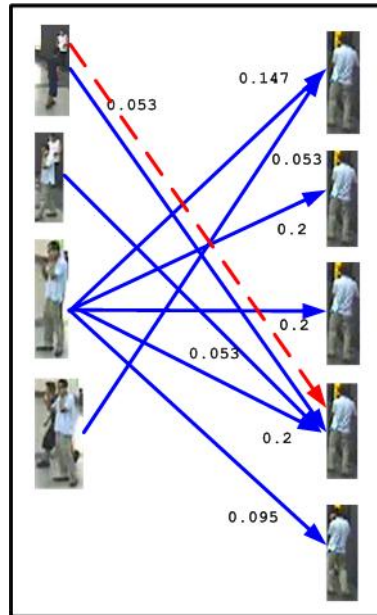


FIG. 5.2 – Mise en correspondance entre des objets en utilisant la distance EMD (en bleu) et la distance de Hausdorff (en rouge). La distance EMD tient compte de la contribution de chaque blob en fonction de son poids. Le chiffre associé à chaque paire de blobs montre la participation de ces blobs dans la mise en correspondance basée sur la distance EMD (en bleu) Plus le poids du blob est élevé, plus la contribution du blob est impacte. Le poids du blob 3 de l’objet 1064 est le plus élevé (0.842), ce blob a un rôle important dans la mise en correspondance de cette objet alors que le rôle du blob 1 (non pertinent) n’est pas considérable.

5.5 Mise en correspondance des objets basée sur les descripteurs temporels

Grâce aux méthodes présentées dans le chapitre 4, une trajectoire peut être représentée aux niveaux numérique et symbolique. Il est à noter que les trajectoires dans notre travail sont représentées en utilisant leurs points de contrôle. L’ensemble des mises en correspondance entre des objets en s’appuyant sur leurs trajectoires \mathcal{F}_T inclut deux éléments : la mise en correspondance entre deux trajectoires au niveau numérique (f_T^{num}) et celle au niveau symbolique (f_T^{sym}). Nous réutilisons deux distances entre des trajectoires proposées par Chen et al. [Chen 2004] basées sur la distance d’édition (ED) entre deux séquences de caractères. Les prochaines sections visent à présenter la distance d’édition et les deux mises en correspondance.

5.5.1 La distance d’édition

La distance d’édition a été beaucoup utilisée pour définir les appariements approximatifs sur les séquences de caractères. Nous rappelons simplement quelques

principes de base.

Une édition est une transformation élémentaire qui remplace un caractère dans une séquence par un autre. Une substitution est une édition qui transforme un caractère d'une séquence en un autre à la même position. Une insertion (respectivement une destruction) qui insère (respectivement détruit) un caractère dans une séquence est aussi une édition.

Une distance d'édition entre deux séquences correspond au nombre minimal d'opérations d'édition qui transforment une séquence de caractères dans une autre.

Soient deux séquences $A(N)$, $B(M)$ contenant N et M caractères respectivement, $A[i]$, $B[j]$ les caractères à la position i et j de A et B , la distance ED est définie :

$$ED(A(N), B(M)) = \begin{cases} N & \text{si } M = 0 \\ M & \text{si } N = 0 \\ ED(A(N-1), B(M-1)) & \text{si } A[N] = B[M] \\ \min \begin{cases} ED(A(N-1), B(M-1)) + sub(A[N], B[M]) \\ ED(A(N-1), B(M)) + des(A[N]) \\ ED(A(N), B(M-1)) + ins(B[M]) \end{cases} & \text{sinon} \end{cases} \quad (5.20)$$

où $sub(A[N], B[M])$, $des(A[N])$, $ins(B[M])$ sont les coûts de faire la substitution, la destruction et l'insertion un caractère de A à B .

Supposons que ces coûts sont égaux 1, la distance ED (cf. équation 5.20) devient :

$$ED(A(N), B(M)) = \begin{cases} N & \text{si } M = 0 \\ M & \text{si } N = 0 \\ ED(A(N-1), B(M-1)) & \text{si } A[N] = B[M] \\ \min \begin{cases} ED(A(N-1), B(M-1)) + 1 \\ ED(A(N-1), B(M)) + 1 \\ ED(A(N), B(M-1)) + 1 \end{cases} & \text{sinon} \end{cases} \quad (5.21)$$

5.5.2 Mise en correspondance entre des trajectoires au niveau numérique

Soit Q et P deux objets, leurs trajectoires au niveau numérique :

$$T^{num}(Q) = [(\theta_1^Q, \delta_1^Q), \dots, (\theta_n^Q, \delta_n^Q)], T^{num}(P) = [(\theta_1^P, \delta_1^P), \dots, (\theta_m^P, \delta_m^P)]$$

La distance entre deux trajectoires de Q et P au niveau numérique $EDR(Q, P, n, m)$ est définie en se basant sur la distance ED (voir équation 5.21) :

$$EDR(Q, P, n, m) = \begin{cases} n & \text{si } m = 0 \\ m & \text{si } n = 0 \\ EDR(Q, P, n-1, m-1) & \text{si } match((\theta_{a,n}, \delta_{a,n}), (\theta_{b,m}, \delta_{b,m})) \\ \min \begin{cases} EDR(Q, P, n-1, m-1) + 1 \\ EDR(Q, P, n-1, m) + 1 \\ EDR(Q, P, n, m-1) + 1 \end{cases} & \text{sinon} \end{cases} \quad (5.22)$$

où $match((\theta_{a,i}, \delta_{a,i}), (\theta_{b,j}, \delta_{b,j}))$ est défini par :

$$match((\theta_{a,i}, \delta_{a,i}), (\theta_{b,j}, \delta_{b,j})) = \begin{cases} vrai & \text{si } |\theta_{a,i} - \theta_{b,j}| \leq \varepsilon_{dir} \text{ et } |\delta_{a,i} - \delta_{b,j}| \leq \varepsilon_{dis} \\ faux & \text{sinon} \end{cases} \quad (5.23)$$

La mise en correspondance entre Q et P au niveau numérique (f_T^n) :

$$f_T^{num}(Q, P) : (\mathcal{D}_{T^{num}} \times \mathcal{D}_{T^{num}}) \rightarrow \mathcal{R} \stackrel{\text{def}}{=} EDR(Q, P, n, m) \quad (5.24)$$

5.5.3 Mise en correspondance entre des trajectoires au niveau symbolique

Les trajectoires de Q et P au niveau symbolique : $Tsym(Q) = [A_1^Q, \dots, A_n^Q]$ et $Tsym(P) = [A_1^P, \dots, A_m^P]$. La distance entre deux trajectoires de Q et P au niveau symbolique $EDM(Q, P, n, m)$:

$$EDM(Q, P, n, m) = \begin{cases} n & \text{si } m = 0 \\ m & \text{si } n = 0 \\ EDM(Q, P, n-1, m-1) & \text{si } ap_match(A_n^Q, A_m^P) \\ \min \begin{cases} EDM(Q, P, n-1, m-1) + 1 \\ EDM(Q, P, n-1, m) + 1 \\ EDM(Q, P, n, m-1) + 1 \end{cases} & \text{sinon} \end{cases} \quad (5.25)$$

où $ap_match(A_i^Q, A_j^P)$ est défini par :

$$ap_match(A_i^Q, A_j^P) = \begin{cases} vrai & \text{si } A_i^Q = A_j^P \text{ ou } A_i^Q \text{ est voisin de } A_j^P \\ faux & \text{sinon} \end{cases} \quad (5.26)$$

La relation de voisinage de deux symboles est déterminée par la position des symboles dans l'espace défini par la direction de mouvement et l'incrément relatif (voir figure 4.12 du chapitre 4).

La mise en correspondance entre deux trajectoires de Q et P au niveau symbolique :

$$f_T^{sym}(Q, P) : (\mathcal{D}_{T^{sym}} \times \mathcal{D}_{T^{sym}}) \rightarrow \mathcal{R} \stackrel{\text{def}}{=} EDM(Q, P, n, m) \quad (5.27)$$

5.5.4 Discussions

Deux mises en correspondance entre des trajectoires reprises de Chen et al. [Chen 2004] sont présentées. La différence de notre travail et celui de Chen et al. est que les mises en correspondance sont effectuées sur les points de contrôle des trajectoires dans notre travail. En plus de la distance d'édition, d'autres distances peuvent être étudiées.

5.6 Relations temporelles

L'ordre des événements reconnus dans le temps joue un rôle important dans les systèmes de vidéosurveillance. La présence des événements qui se sont passés dans un ordre permet de définir des événements plus complexes. Par exemple, l'événement "une personne abandonne une valise dans une station de métro" est reconnu si deux événements "la personne est à côté de la valise ou porte la valise" et "la personne s'éloigne de la valise" sont reconnus et le premier événement s'est passé avant le deuxième événement.

Nous utilisons la représentation de temps par l'intervalle et les relations temporelles proposées par Allen [Allen 1983].

Soit un intervalle de temps I , I est représenté par : $I = [I^l, I^r]$ où I^l et I^r sont deux points limites de I , $I^l < I^r$. La durée de I est définie : $|I| \cong I^r - I^l$.

En se basant sur cette représentation, Allen [Allen 1983] a introduit 13 relations temporelles entre deux intervalles de temps dont 7 relations de base et leurs inverses.

Soit I_1 et I_2 deux intervalles de temps, les sept relations ($=$, $<$, $>$, o , m , d , s , f) temporelles de I_1 et I_2 sont listées dans le tableau 5.2. À partir de cinq relations (o , m , d , s , f), cinq relations inverses (oi , mi , di , si , fi) sont également définies.

TAB. 5.2 – 7 relations temporelles entre deux intervalles de temps I_1 et I_2 .

Nom	Anglais	Notation	Définition
égal	equal	$I_1 = I_2$	$(I_1^l = I_2^l) \wedge (I_1^r = I_2^r)$
avant	before	$I_1 < I_2$	$(I_1^r < I_2^l)$
recouvrement	overlap	$I_1 o I_2$	$(I_1^l < I_2^l) \wedge (I_1^r > I_2^r) \wedge (I_1^l < I_2^r)$
rencontre	meets	$I_1 m I_2$	$(I_1^r = I_2^l)$
pendant	during	$I_1 d I_2$	$(I_1^l > I_2^l) \wedge (I_1^r < I_2^r)$
départ	starts	$I_1 s I_2$	$(I_1^l = I_2^l) \wedge (I_1^r < I_2^r)$
fin	finishes	$I_1 f I_2$	$(I_1^l > I_2^l) \wedge (I_1^r = I_2^r)$

Soit f_{Temps} un élément de l'ensemble des relations temporelles \mathcal{F}_{Temps} , Q et P deux objets ou deux événements, leurs intervalles de temps sont I^Q , I^P respectivement. La mise en correspondance des objets et des événements en se basant sur les relations temporelles devient :

$$f_{Temps}(P, Q) : (\mathcal{D}_I \times \mathcal{D}_I) \rightarrow Boolean \stackrel{\text{def}}{=} I^Q \alpha I^P \quad (5.28)$$

où α est une des 13 relations temporelles dont 7 relations définies dans le tableau 5.2 ($=$, $<$, $>$, o , m , d , s , f) et 5 relations inverses (oi , mi , di , si , fi).

5.7 Langage de requêtes

5.7.1 Introduction

L'indexation et la recherche d'images n'ont pas besoin d'avoir un langage de requêtes car la recherche commence avec une région ou une image entière. Les résultats sont toujours des images, les descripteurs sont fixés ou choisis par l'utilisateur en cliquant sur le bouton approprié. La figure 5.3 montre l'interface du système d'indexation et de recherche d'images proposé par le projet IMEDIA, INRIA ¹. Les requêtes dans ce système sont simplement soit une image fournie par l'utilisateur soit une image choisie par l'utilisateur en naviguant dans la base de données soit quelques mots clés (si l'annotation est disponible) (voir les zones en vert dans la figure 5.3).

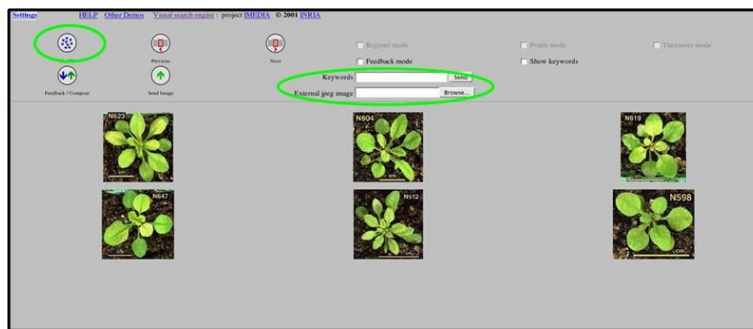


FIG. 5.3 – Interface du système d'indexation et de recherche d'images proposé par le projet IMEDIA, INRIA. Trois types de requêtes (en vert) sont possibles : une image fournie par l'utilisateur, une image choisie par l'utilisateur en naviguant dans la base de données, quelques mots clés (si l'annotation est disponible).

En indexation et recherche de journaux télévisés, les requêtes sont des images d'exemple ou des plans de vidéo recherchés. Si les concepts sont détectés dans la phase d'indexation, les requêtes peuvent être un ou plusieurs concepts prédéterminés (p. ex. "aircraft").

Pour l'indexation et la recherche de vidéos de vidéosurveillance, un langage de requêtes est nécessaire car le contenu est riche. De nombreux types d'objets et d'événements avec de nombreuses relations sont disponibles. Dans l'état de l'art (cf. chapitre 2), nous avons introduit deux travaux concernant le langage de requêtes : l'un de Tian et al. [Tian 2008] et l'autre de Ghanem et al. [Ghanem 2004]. Le travail de Tian et al. est dédié à un système commercial de IBM [Tian 2008] qui comprend d'une part l'analyse de vidéos telle que la détection, le suivi, la classification d'objets et la reconnaissance d'événements et d'autre part la recherche d'objets et d'événements. La figure 5.4 montre l'interface du système de IBM. Un ensemble limité de requêtes est à gauche. La requête "Find Cars" est activé, des petites vidéos à droite sont des résultats retrouvés pour cette requête. Les requêtes prises grâce à

¹<http://www-rocq.inria.fr/cgi-bin/imedia/circario.cgi/demos>

l'interface vont être exprimées dans le langage SQL (Structured Query Language). La recherche d'objets et d'événements se base sur les méta-données comme dans les bases de données relationnelles. Le langage de requêtes basé sur le réseau de Pétri est proposé par Ghanem et al. [Ghanem 2004]. Ce langage permet d'exprimer les relations entre des événements et celles entre des objets. Il se base également sur les méta-données. Ces deux langages ont deux inconvénients.(1) Les requêtes sont aux niveaux objets et événements. Ils ne permettent pas d'exprimer la requête contenant une image d'exemple. (2) La mise en correspondance entre des données indexées est la mise en correspondance exacte. Cela n'est pas suffisant pour l'indexation et la recherche de vidéos de vidéosurveillance en raison de l'imperfection d'analyse. Les techniques de mise en correspondance basées sur les descripteurs visuels ne peuvent pas être appliquées.

Notre contribution est un nouveau langage de requêtes qui dépasse ces limitations. Dans les sections suivantes, nous présentons le syntaxe de ce langage et une comparaison de notre langage de requêtes et celui de Ghanem et al. [Ghanem 2004].

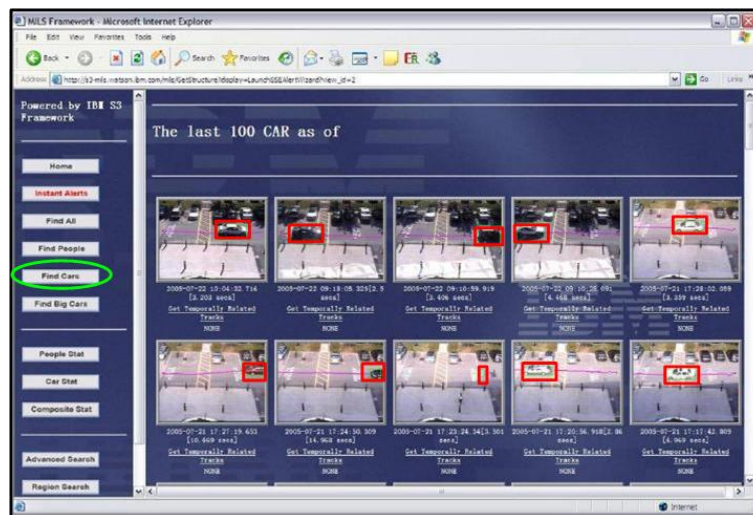


FIG. 5.4 – Interface du système d'indexation et de recherche de vidéos de vidéosurveillance. Un ensemble limité de requêtes est à gauche. La requête “Find Cars” (en vert) est activé, des petites vidéos à droite sont des résultats de cette requête ([Tian 2008]).

5.7.2 Syntaxe

Les requêtes exprimées par le langage sont sous la forme :

$$SELECT \langle Select\ list \rangle FROM \langle Database \rangle WHERE : ENTITIES \langle Entities\ list \rangle \\ > CONDITIONS \langle Conditions\ list \rangle$$

où **SELECT**, **FROM**, **WHERE :**, **ENTITIES**, et **CONDITIONS** sont les mots clés.

Select list : identifie quelle partie des résultats l'utilisateur souhaite recevoir.

Database : ce champ permet à l'utilisateur de choisir les vidéos sur lesquelles la recherche s'effectue.

Entities list : ce champ identifie le type de données avec lequel il veut travailler.

Ce champ est une séquence des $(v : type)$ séparés par une virgule.

v est une variable, $type$ est un des concepts définis par le modèle de données. Il inclut *Physical_objects* (pour les objets), *Events* (pour les événements), et *SubImage* (pour les images d'exemple). Par exemple : $(p : Physical_objects)$ déclare une variable p du type de *Physical_objects*, donc la valeur de p sont les objets dans la vidéo.

Conditions list : Avant de définir ce champ, nous déterminons le terme : la fonction d'accès.

La fonction d'accès : $(u's)$ est la fonction qui retourne la valeur d'un attribut. Si l'on écrit : $(u's X)$ c'est-à-dire qu'on prend l'attribut X de u où u est une variable définie dans **Entities list**.

La liste des fonctions d'accès correspondantes à chacun des types des données est identifiée dans l'annexe A.

Une constante c est une valeur numérique ou une chaîne de caractères. Pour constante contenant une chaîne des caractères : les caractères sont entourés par "".

Nous définissons deux types de conditions :

$$e_1 : (u's X \theta v's Y) \quad (5.29)$$

$$e_2 : (u's X \theta c) \quad (5.30)$$

où u, v sont deux variables, θ est un opérateur.

Le champ **Conditions list** est une séquence des e_1 et e_2 .

Nous définissons trois types d'opérateurs : les opérateurs de base (θ_{base}), les opérateurs temporels (θ_{temp}) et les opérateurs d'apparence (θ_{ap}).

- Les opérateurs de base sont des opérateurs de comparaison qui incluent $\theta_{base} = \{=, <, >, >=, = <, !=\}$ et l'opérateur de relation entre les objets et les événements incluant *involved_in*, p.ex. $p \text{ involved_in } e$ identifie l'objet p impliqué dans l'événement e .
- Les opérateurs temporels θ_{temp} qui incluent les mises en correspondance des trajectoires et les relations temporelles : $\theta_{temp} \in \mathcal{F}_T, \mathcal{F}_{Temps}$;
- Les opérateurs d'apparence qui incluent les mises en correspondance d'apparence : $\theta_{ap} \in \mathcal{F}_B, \mathcal{F}_O$.

5.7.3 Exemples de requêtes exprimées par le langage proposé

La requête : "Trouver les événements *Close_to_Gates* qui sont reconnus dans les vidéos" est exprimée dans le langage proposé :

```
SELECT e FROM * WHERE : ENTITIES ((e : Events)) CONDITIONS ((e's
Name = "Close_to_Gates"))
```

où e est une variable du type *Events*, e 's *Name* est une fonction d'accès de l'attribut *Name* de e , "Close_to_Gates" est une chaîne des caractères.

Afin de permettre à l'utilisateur de se familiariser avec le langage, nous nommons les opérateurs temporels et d'apparence. Dans ce cas, nous pouvons enlever la fonction d'accès. La liste de noms des opérateurs est donnée en annexe A.

La requête : "Trouver les personnes dans les vidéos qui sont semblables à une image requête" devient :

```
SELECT p FROM * WHERE : ENTITIES ((p : Physical_objects), (i : SubImage)) CONDITIONS ((p's Class = "Person"), (i DoG_matching p))
```

où DoG_matching est le nom de l'opérateur d'apparence qui se base sur la mise en correspondance des objets par les points d'intérêts DoG et le descripteur SIFT.

5.7.4 Vers une comparaison avec l'approche de Ghanem et al.

L'approche de Ghanem et al. vise à définir un événement composé à partir des événements simples en se basant sur un réseau de Pétri [Ghanem 2004]. Cette section vise à donner une comparaison de notre langage avec le travail de Ghanem et al.

Nous reprenons deux exemples de requêtes introduits par Ghanem et al. et les exprimons dans notre langage de requêtes.

La première requête consiste à compter le nombre de voitures qui sont garées dans un zone pendant une période. Soit P une voiture, la requête vérifie l'ensemble des contraintes :

- l'événement E1 : la voiture P apparaît (nommé "appears");
- l'événement E2 : la voiture P entre dans la zone A0 (nommé "enters_regions_A0");
- l'événement E3 : la voiture P s'arrête (nommé "stops");
- l'événement E4 : la voiture P sort de la zone A0 (nommé "leaves_regions_A0");
- E1 se passe avant E2;
- E2 se passe avant E3;
- E3 se passe avant E4.

Cette requête est définie en utilisant le réseau de Pétri (voir figure 5.5). Elle est exprimée dans notre langage :

```
SELECT COUNT(P) FROM * WHERE : ENTITIES ((P : Physical_objects), (E1 : Events), (E2 : Events), (E3 : Events), (E4 : Events)) CONDITIONS ((P's Class = "Car") (E1's Name = "appears") (E2's Name = "enters_regions_A0") (E3's Name = "stops") (E4's Name = "leaves_regions_A0") (P involved_in E1) (P involved_in E2) (P involved_in E3) (P involved_in E4) (E1 before E2) (E2 before E3) (E3 before E4))
```

La deuxième requête consiste à détecter une personne qui passe d'une voiture à l'autre. Soit P une personne, $c0$ et $c1$ deux voitures, la requête vérifie :

- E1 : la voiture C0 s'arrête (nommé "parks");
- E2 : la voiture C1 s'arrête (nommé "parks");

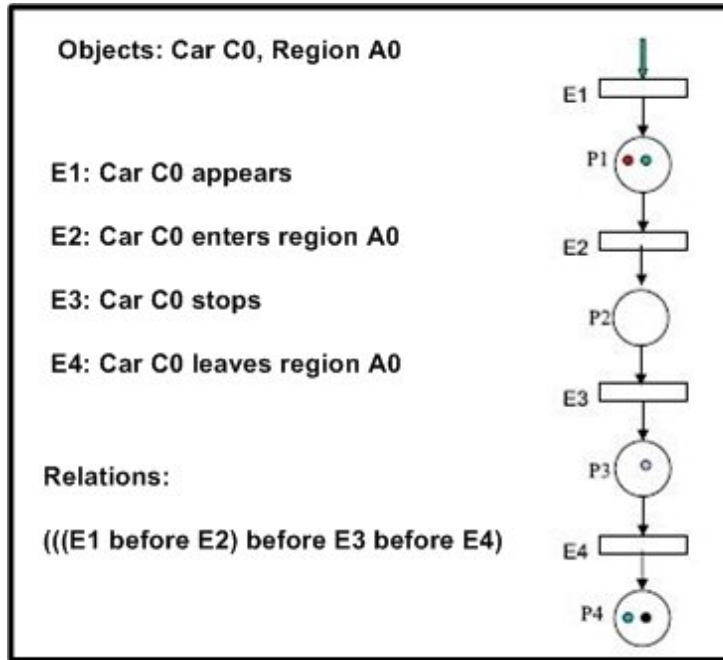


FIG. 5.5 – Réseau de Pétri pour la requête “compter le nombre de voitures qui sont garées dans un zone pendant une période” ([Ghanem 2004]).

- E3 : la personne P sort de la voiture C0 (nommé “exits”) ;
- E4 : la personne P entre à la voiture C1 (nommé “enters”) ;
- E5 : la voiture C1 part (nommé “leaves”) ;
- E1 se passe avant E2 ;
- E2 se passe avant E3 ;
- E3 se passe avant E4 ;
- E4 se passe avant E5.

La figure 5.6 montre l’expression de cette requête par le réseau de Pétri. La requête exprimée dans notre langage :

```
SELECT p FROM * WHERE : ENTITIES ((P : Physical_objects), (C0 : Physical_objects), (C1 : Physical_objects), (E1 : Events), (E2 : Events), (E3 : Events), (E4 : Events), (E5 : Events)) CONDITIONS ((P's Class = "Person") (C0's Class = "Car")(C1's Class = "Car") (E1's Name = "parks") (E2's Name = "parks") (E3's Name = "exits") (E4's Name = "enters") (E5's Name = "leaves") (C0 involved_in E1) (C1 involved_in E2) (C0 involved_in E3) (P involved_in E3) (C1 involved_in E4) (P involved_in E4) (C1 involved_in E5) (E1 before E2) (E2 before E4) (E3 before E4) (E4 before E5))
```

Notre approche permet d’exprimer autant de requêtes que l’approche de Ghanem et al. L’approche de Ghanem et al. suppose que : les résultats des modules d’analyse sont parfaits, la mise en correspondance donc se base sur les étiquettes.

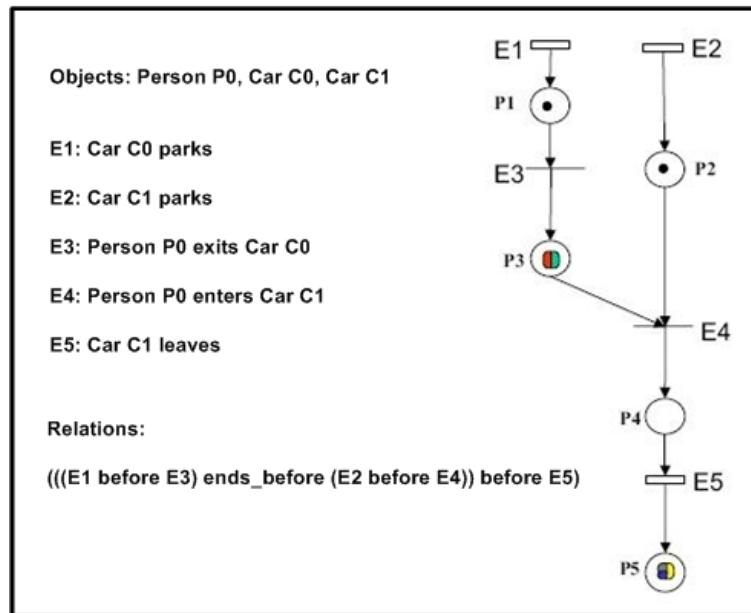


FIG. 5.6 – Réseau de Pétri pour la requête “trouver une personne qui passe d’une voiture à l’autre” ([Ghanem 2004]).

Elle ne permet pas d’exprimer les requêtes telles que : “trouver les personnes qui sont proches de la porte et semblables à une image par les couleurs dominantes”. Cette requête exprimée par notre langage de requêtes :

```
SELECT p FROM * WHERE : ENTITIES ((p : Physical_objects), (i : SubImage), (e : Events))
CONDITIONS ((p's Class = "Person")
(e's Name = "Close_to_Gate") (p involved_in e) (i DoCo_matching p))
```

Notre langage a une limitation par rapport au langage de Ghanem et al. : il n’est pas hiérarchique. Une requête complexe ne peut donc pas être exprimée en combinant quelques requêtes simples prédéfinies.

5.7.5 Discussions

Un langage de requêtes est un outil qui permet de faire le lien entre l’utilisateur et le système d’indexation et de recherche. La performance d’un langage de requêtes est prouvée par son expressivité et sa facilité. Le langage de requêtes proposée donne un moyen à l’utilisateur pour accéder à l’information préparée par les tâches dans la phase d’indexation de manière qui est décidée par les méthodes de mise en correspondance dans la phase de recherche. Nous présentons l’utilisation du notre langage de requêtes dans le chapitre 6.

5.8 Recherche interactive et Retour de pertinence

5.8.1 Introduction

Nous remarquons que dans l'état de l'art en indexation et recherche de vidéos pour la vidéosurveillance, peu de travaux sont dédiés au retour de pertinence (voir chapitre 2). Nous pouvons mentionner le travail de Messen et al. [Meessen 2007] consistant à retrouver des frames d'intérêt dans les vidéos de vidéosurveillance, celui de Chen et al. [Chen 2007] pour les trajectoires en utilisant des réseaux de neurones.

Nous proposons deux méthodes de retour de pertinence à court terme pour la recherche d'objets dans les vidéos de vidéosurveillance : l'une basée sur plusieurs images d'exemple et l'autre basée sur les SVM à une classe. Ces deux méthodes ne travaillent qu'avec les exemples positifs. Cette contribution vient de deux contributions concernant la détection des blobs représentatifs et la mise en correspondance entre des objets.

Ces deux méthodes visent à exploiter la connaissance de l'utilisateur provenant des retours des itérations antérieures et de ceux de l'itération en cours.

Soit :

- \mathcal{O} : l'ensemble d'objets ;
- $\mathcal{R}^t \in \mathcal{O}$, $\mathcal{R}^t = \{O_i^t\} | i = 1, \dots, N_{obj}$: les résultats de recherche à l'itération t . Les résultats de \mathcal{R}^t sont ordonnés de manière croissante selon leurs distances avec la requête ;
- O_i^t devient $O_i^{P,t}$ s'il est choisi par l'utilisateur comme exemple positif ;
- $\mathcal{R}^0 \in \mathcal{O}$: les résultats de recherche sans retour de pertinence ;
- M : le nombre de résultats que l'utilisateur veut recevoir à chaque itération. $\mathcal{R}^t(M)$ sont les M premiers résultats de l'ensemble \mathcal{R}^t .

La recherche démarre avec une image d'exemple I ou un objet recherché O_Q . La performance du retour de pertinence est évaluée par deux mesures :

- Le nombre de résultats pertinents de $\mathcal{R}^t(M)$ est supérieur à celui de $\mathcal{R}^{t-1}(M)$;
- Le nombre d'itérations est le plus petit possible.

La première mesure montre que le retour de pertinence permet de trouver le plus de résultats pertinents possibles dans les premiers résultats. La deuxième mesure montre qu'il demande le moins d'échanges possible avec l'utilisateur.

Le processus commun des deux méthodes est montré par l'algorithme 11. Ces deux méthodes se différencient par les étapes (2), (3) et (4). Nous les détaillons dans les sections suivantes.

5.8.2 Retour de pertinence basé sur plusieurs images d'exemple

Cette méthode se base sur une observation : plus l'utilisateur fournit d'aspects sur l'objet recherché, plus l'apprenant arrive à savoir ce que l'utilisateur veut chercher.

Algorithme 11 : Le retour de pertinence**Input :** I : l'image d'exemple ou O_Q : l'objet recherché d : le descripteur choisi**Output :** t_{fini} est l'itération où l'utilisateur décide de finir la recherche $\mathcal{R}^{t_{fini}}(M)$: les M premiers résultats à l'itération t_{fini} **begin**1 trouver les résultats correspondants à la requête $\mathcal{R}^0(M)$ **while do**2 faire juger à l'utilisateur les exemples positifs ($R_i^{P,t-1}$)

3 appeler l'apprenant

4 appeler le sélectionneur pour avoir de nouveaux résultats $\mathcal{R}^t(M)$ **until** l'utilisateur décide de terminer la recherche**end**

Cette méthode demande à l'utilisateur de fournir à chaque itération quelques blobs qu'il trouve pertinents. Par conséquent, un ensemble de blobs contenant l'image de requête et les blobs rajoutés à chacune des itérations est créé.

Notre mise en correspondance entre des objets introduite dans la section 5.4.2 nous permet de calculer la similarité entre cet ensemble et des objets cibles.

À l'étape (2) de l'algorithme 11, parmi les M premiers résultats $\mathcal{R}^{t-1}(M)$ à l'itération $t-1$, l'utilisateur choisit M_P blobs positifs ou objets positifs. Dans le cas où les retours sont des objets positifs, pour chaque objet, le blob dont le poids est plus élevée sera utilisé.

À l'étape (3) de l'algorithme 11, l'apprenant de cette méthode crée simplement un objet intermédiaire O_I , sa représentation est $R_{O_I,ap} = \{(B_i, w_i)\} | i = 1, \dots, M_P$ où B_i est le blob jugé positif. Pour w_i , nous initions simplement $w_i = \frac{1}{M_P}$.

À l'étape (4) de l'algorithme 11, le sélectionneur fait la mise en correspondance entre l'objet O_I créé par l'apprenant à l'étape (3) et les objets cibles par le descripteur choisi. Cette mise en correspondance est présentée dans la section 5.4.2. Le sélectionneur rend ensuite à l'utilisateur les M résultats $\mathcal{R}^t(M)$ dont les distances avec l'objet O_I sont les plus petites.

5.8.3 Retour de pertinence par les machines à vecteurs de support

La méthode de retour de pertinence basée sur plusieurs images d'exemple est simple. Cependant, elle ne permet pas d'apprendre explicitement des retours de l'utilisateur. Pour apprendre des retours, nous choisissons les machines à vecteurs de support qui ont prouvés leurs performances pour le retour de pertinence en indexation et recherche de textes et d'images.

Parmi plusieurs types de SVM, les SVM à deux classes et celles à une classe sont

largement utilisées. Les SVM à deux classes consistent à trouver un hyperplan qui maximise la marge d'erreur, qui est la somme des distances entre l'hyperplan et les exemples positifs et négatifs les plus proches de cet hyperplan. Les SVM à une classe cherchent de trouver une région minimale qui contient le plus possible d'exemples positifs. L'application des SVM à deux classes au retour de pertinence présente deux inconvénients : les exemples négatifs sont divers, il est donc difficile de trouver leur distribution ; l'utilisateur s'intéresse aux exemples positifs et non pas aux exemples négatifs. Pour l'indexation et la recherche de vidéos de vidéosurveillance, l'utilisateur s'intéresse à trouver les objets qui sont semblables (objets positifs) d'un objet particulier. Cela justifie le choix de SVM à une classe pour notre méthode de retour de pertinence.

Il est intéressant d'analyser la différence entre cette méthode et celles en indexation et recherche d'images présentées dans l'état de l'art. Dans l'indexation et la recherche d'images, une image positive habituellement peut avoir des régions pertinentes ainsi que des régions non pertinentes. Si l'utilisateur juge une image positive sans indiquer les régions pertinentes, le retour de pertinence doit les déduire. En indexation et recherche de vidéos de vidéosurveillance au niveau objets, la détection des blobs représentatifs assure que les blobs choisis sont pertinents. La méthode de retour de pertinence donc peut utiliser tous les blobs des objets positifs. Avec cela, la base d'apprentissage est considérable, l'entraînement des SVM est stable.

Nous présentons tout d'abord les SVM à une classe qui ont été proposées par Schölkopf et al. [Schölkopf 1999]. Nous proposons ensuite notre méthode de retour de pertinence en combinant les SVM à une classe avec les blobs représentatifs.

Notons que $\{x_1, \dots, x_l\}$ est l'ensemble de l données et $f(x_i) \in \{1, -1\}$ est la classe de x_i . La valeur $f(x_i)$ est égal 1 si x_i est un exemple positif tandis que $f(x_i)$ est égal -1 si x_i est un exemple négatif. Pour un nouveau exemple x , les SVM à une classe visent à déterminer si l'exemple x appartient à la classe des exemples positifs.

Les SVM à une classe cherchent de trouver une région minimale qui contient le plus possible d'exemples positifs. Les SVM sont représentés par :

$$\min_{w, \rho, \xi} \frac{1}{2} \|w\|^2 + \frac{1}{\nu l} \sum_{i=1} \xi_i - \rho \quad (5.31)$$

sous les contraintes :

$$(w \cdot \phi(x_i)) \geq \rho - \xi_i \text{ où } \xi_i \geq 0 \quad (5.32)$$

où $\nu \in (0, 1]$ est un paramètre.

Si ce problème (voir équations 5.31 et 5.32) est résolu, les valeurs de w et ρ sont bien définies. La fonction de décision pour l'entrée x est :

$$f(x) = \text{sgn}((w \cdot \phi(x)) - \rho) \quad (5.33)$$

Nous précisons les étapes de notre méthode selon le processus (voir algorithme 11) :

À l'étape (2) de l'algorithme 11, parmi les M premiers résultats $\mathcal{R}^{t-1}(M)$ à l'itération $t - 1$, l'utilisateur choisit M_P exemples positifs $O_i^{P,t-1} | i = 1, \dots, M_P$.

À l'étape (3) de l'algorithme 11, l'apprenant de cet algorithme effectue :

- pour chaque objet positif, l'extraction de descripteurs choisis sur tous les blobs ;
- l'entraînement des modèles de SVM (voir équations 5.31 et 5.32).

La similarité de l'objet O_j avec la requête :

$$p(O_j) = w \cdot \phi(O_j) - \rho \quad (5.34)$$

À l'étape (4) de l'algorithme 11, le sélectionneur calcule la valeur de p (voir équation 5.34) pour tous les objets dans la base de données, les ordonne de manière décroissante et choisit M objets $\mathcal{R}^t(M)$ dont les valeurs de p sont les plus grandes pour afficher à l'utilisateur.

5.8.4 Discussions

Nous notons trois caractéristiques de nos méthodes de retour de pertinence :

- Le retour de pertinence proposé est au niveau objets ;
- La recherche abordée dans cette thèse est la recherche d'objets spécifiques (target search) qui est différente de la recherche de la classe des objets (category search). L'objectif de la recherche d'objets spécifiques est de trouver les objets qui sont semblables à un objet recherché tandis que celle de la classe des objets est de trouver les objets appartenant à la même classe de l'objet recherché. Comme présentée dans le chapitre 3, la classification des objets du module d'analyse vidéos est facultative. Si elle est présente, la classe de l'objet recherché peut être explicitement déterminée. Sinon, tous les objets appartiennent à une classe globale (objets mobiles). La recherche de la classe permet d'annoter les objets trouvés par la classe de l'objet recherché. Ceci peut être utilisé pour classer les objets détectés dans les vidéos de vidéosurveillance où la classification n'est pas disponible. Cela fait partie de notre travail futur.
- Le retour de pertinence est à court terme. Cependant les résultats obtenus par retour de pertinence peuvent être utilisés pour justifier le choix de descripteurs pour la mise en correspondance future. Cela fait également partie de notre travail futur.

5.9 Conclusion

Dans ce chapitre, nous définissons trois types de mise en correspondance entre des objets : celui des descripteurs d'apparence, celui des descripteurs temporels et les relations temporelles. En se basant sur les mises en correspondance des blobs par

les descripteurs d'apparence (couleurs dominantes, matrices de covariance, histogrammes de contour, points d'intérêt), nous proposons les mises en correspondance des objets contenant plusieurs blobs.

Nous résumons les contributions de notre travail de thèse dans ce chapitre :

- Une nouvelle mise en correspondance entre des objets en utilisant la distance EMD sur leurs blobs représentatifs est introduite. Cette distance nous permet de (1) prendre en compte la distance des blobs par les descripteurs d'apparence proposée dans l'état de l'art, (2) mettre en correspondance les objets contenant un nombre de blobs différent (3) apparier partiellement les objets ;
- Un nouveau langage de requêtes est présenté. Ce langage de requêtes se base sur notre modèle de données (voir section 4.2) et les mises en correspondance entre les informations indexées (voir section 5.3). Grâce à ce langage, des requêtes à différents niveaux de sémantique sont exprimées ;
- Deux méthodes de retour de pertinence à court terme sont proposées : l'une basée sur plusieurs images d'exemples et l'autre basée sur les SVM à une classe.

Implémentation et évaluation

Dans ce chapitre, nous présentons les résultats expérimentaux obtenus de l'approche proposée. Nous présentons tout d'abord l'implémentation de l'approche proposée et le module d'analyse vidéo choisi avec ses caractéristiques (section 6.2). Nous introduisons ensuite les bases de vidéos utilisées (section 6.3). Nous rappelons également les caractéristiques de la nature des vidéos que nous énonçons dans le chapitre 1. Les mesures d'évaluation des approches d'indexation et de recherche d'information habituellement utilisées sont présentées (section 6.4). Nous analysons l'utilisation du langage de requêtes proposé (section 6.6). Les résultats de deux méthodes proposées pour la détection des blobs représentatifs dont l'une basée sur le changement d'apparence et l'autre basée sur le regroupement des blobs sont présentés. Nous comparons notre méthode de détection des blobs représentatifs basée sur le regroupement des blobs avec celle de Ma et al. [Ma 2007] (section 6.7). La qualité de la recherche à différents niveaux est analysée. Pour la recherche au niveau objets, nous comparons notre méthode avec deux méthodes de l'état de l'art dont l'une de Ma et al. [Ma 2007] et l'autre de Calderara et al. [Calderara 2006]. Une évaluation des descripteurs d'apparence en indexation et recherche de vidéos de vidéosurveillance est également donnée (sections 6.8 et 6.9). Les premiers résultats obtenus avec deux méthodes de retour de pertinence (le retour de pertinence basé sur plusieurs images d'exemple et celui basé sur les SVM) sont présentés (section 6.10).

6.1 Implémentation

Nous avons implémenté un prototype en C++ pour l'indexation et la recherche de vidéosurveillance. Ce prototype divisé en 3 bibliothèques comprend les algorithmes que nous proposons dans cette thèse. Nous présentons en détail ce prototype en annexe B.

6.2 Modules d'analyse vidéo

Comme nous l'expliquons dans les hypothèses du chapitre 1, les vidéos doivent être prétraitées par un module d'analyse. Afin d'analyser des vidéos, nous choisissons un module d'analyse vidéo automatique et une annotation manuelle. L'annotation manuelle peut être considérée comme un résultat obtenu par un bon module d'analyse. L'utilisation de résultats provenant du module d'analyse automatique vidéo a pour objectif d'évaluer la qualité de la recherche de vidéos avec un module d'analyse

vidéo ayant une certaine qualité. Ce module peut analyser des vidéos à différents niveaux. L'objectif de l'utilisation des informations annotées manuellement est de mesurer la qualité de la recherche de vidéos sous des conditions différentes (p. ex. personnes observées par plusieurs caméras, changement d'illumination, occultation des personnes). Il est à noter que d'autres modules d'analyse peuvent être utilisés.

6.2.1 Plate-forme VSIP (Video Surveillance Interpretation Platform)

La plate-forme VSIP est une plate-forme d'analyse vidéo proposée par le projet PULSAR [Avanzi 2005], [Fusier 2007]. Cette plate-forme est composée d'algorithmes de détection, de suivi, de classification d'objets et de reconnaissance d'événements. Nous les décrivons brièvement. Nous rappelons les caractéristiques de la plate-forme en se basant sur les critères mentionnés dans le chapitre 1.

6.2.1.1 Détection et classification d'objets

Afin de détecter des objets, cette plate-forme [Avanzi 2005] utilise la technique nommée soustraction de l'arrière-plan (background subtraction). Cette technique [Zúniga 2006] consiste à calculer la différence d'intensité des pixels du frame traité et l'image de l'arrière-plan. L'image de l'arrière-plan est une image de la scène vide. Une marque binaire dont les pixels 1 (les pixels des objets) correspondant à l'objet et les pixels 0 (les pixels du fond) correspondant à l'arrière plan, est créée en déterminant si la différence d'intensité entre deux pixels est supérieure à un seuil. Les pixels des objets sont regroupés par les relations de voisinage. Chacun des groupes correspond à une région. Les dimensions et les positions des objets en 3D sont calculées. Les objets en 3D sont comparés avec les modèles prédéfinis. La classe d'un objet est la classe du modèle qui est le plus semblable à cet objet. Par exemple, la taille (170 de hauteur et 60 de largeur) est choisie pour la classe Person.

6.2.1.2 Suivi d'objets

Les objets 3D détectés et classifiés dans la section précédente sont ensuite suivis en fonction de leurs positions en 3D et de leurs dimensions en 3D. Dans cette plate-forme, l'algorithme de suivi [Avanzi 2001], [Cupillard 2002] construit un graphe temporel des objets connectés afin de les bien suivre. Les objets détectés sont connectés pour chaque paire de frames consécutifs (frame-to-frame tracker) Les liens des objets sont associés avec des poids qui sont calculés en fonction de la similarité de leurs classes, de leurs dimensions en 3D et de la différence de leurs distances en 3D. Ce graphe est ensuite analysé par un algorithme de suivi long terme afin de créer les chemins pour chacun des objets. Le meilleur chemin d'un objet est ensuite choisi comme sa trajectoire.

6.2.1.3 Reconnaissance d'événements

Les événements d'intérêt sont définis en utilisant un langage spécifique proposé par Vu et al. [Vu 2003]. Ce langage se base sur une ontologie contenant des concepts et des relations entre ces concepts. Deux concepts principaux sont les objets et les événements. Les objets sont les objets mobiles et les objets de contexte. Cette plateforme a quatre types d'événements : état primitif, état composé, événement primitif et événement composé.

Un état est une propriété spatio-temporelle d'un objet à un instant donné ou constante dans un intervalle de temps. Un état primitif est un état qui est directement déduit à partir des attributs visuels de l'objet. Un état composé est une combinaison d'états. Un événement est un changement d'états entre deux instants. Un événement primitif est un changement d'états primitifs. Un événement composé est une combinaison d'états et d'événements. L'approche proposée par Vu et al. [Vu 2003] permet de reconnaître des événements définis.

6.2.1.4 Caractéristiques de la plate-forme VSIP

- L'analyse est effectuée séparément pour chacune des vidéos ;
- Les modules utilisent deux types de connaissances a priori : celle du contexte et celle du domaine. La connaissance du contexte décrit tout ce qui présent dans la scène vide. La connaissance du domaine définit des événements d'intérêt pour le domaine considéré ;
- L'analyse vidéo est effectuée à différents degrés de granularité. Dans cette thèse, les vidéos sont analysées soit par tous les modules de cette plate-forme (la détection, le suivi et la classification d'objets et la reconnaissance d'événements) soit par la détection, le suivi et la classification d'objets ;
- La qualité des résultats des analyses est évaluée dans la section suivante.

6.2.2 Annotation manuelle

Le projet CAVIAR fournit une annotation manuelle des vidéos. Cette annotation est faite par des personnes dans ce projet à l'aide d'un outil. Pour la détection et le suivi d'objets, chaque objet a une étiquette et un ensemble de boîtes englobantes minimales déterminées sur les frames où l'objet est présent. Pour la classification d'objets, deux classes sont prédéfinies : Person et PersonGroup. D'autres informations sont disponibles dans cette annotation ¹. Nous utilisons l'étiquette, la classe et les boîtes englobantes minimales dans cette annotation pour fournir à la phase d'indexation.

¹http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/gt_file_format.txt

6.2.3 Mesures d'évaluation des performances des modules d'analyse vidéo

De nombreux algorithmes ont été proposés pour la détection, le suivi, la classification d'objets et la reconnaissance d'événements des modules d'analyse vidéo. Il est donc nécessaire d'évaluer ces algorithmes. L'évaluation des algorithmes consiste à préparer les bases de vidéos communes, à exécuter ces algorithmes sur ces bases, à calculer les mesures d'évaluation de chacun. Nous utilisons dans cette thèse les mesures présentées dans [Nghiem 2007] :

- Pour la détection d'objets, deux mesures dont le rapport entre le nombre d'objets détectés et le nombre d'objets de référence et le nombre de blobs bien déterminés pour un objet sont utilisées ;
- Le suivi d'objets est évalué par le temps de suivi, la persistance d'étiquette et la confusion d'étiquette. Le temps de suivi d'un objet de référence mesure le pourcentage de temps pendant lequel cet objet est détecté et suivi, par rapport au temps pendant lequel il apparaît dans la scène. La persistance d'étiquette mesure le nombre d'objets détectés et suivis qui sont associés à un seul objet de référence. La confusion d'étiquette mesure le nombre d'objets de référence qui sont présents dans un seul objet détecté et suivi (des objets différents ont la même étiquette) ;
- La performance de la classification d'objets est mesurée par le nombre d'objets détectés dont leurs types sont bien identifiés ;
- Pour la reconnaissance d'événements, la mesure de performance est le nombre d'événements bien reconnus dans un intervalle de temps.

Il est à noter que dans cette thèse, les termes "objet de référence" et "objet réel" sont utilisés de manière interchangeable.

Au niveau objets, les objets ne sont pas retrouvés par notre approche s'ils ne sont pas détectés par les modules d'analyse. Si l'objet est détecté et suivi, nous analysons la relation entre la qualité du suivi d'objets et celle de notre approche. La valeur obtenue pour le nombre de blobs bien déterminés d'une méthode de suivi d'objets nous indique la méthode de détection des blobs représentatifs utilisée. Si cette valeur est élevée, la méthode de détection des blobs représentatifs basée sur le changement d'apparence est utilisée. Sinon, la méthode basée sur le regroupement des blobs est préférée. La valeur de la persistance d'étiquette et celle de la confusion d'étiquette déterminent le choix de mise en correspondance : exacte ou similaire. Si elles sont élevées, la mise en correspondance par la similarité est primordiale. La valeur du temps de suivi d'un objet influence les relations temporelles.

Les résultats de la classification affecte ceux de la recherche si l'utilisateur indique explicitement le type d'objet qu'il veut chercher.

Au niveau événements, le type d'événements et l'occurrence jouent un rôle important pour la recherche au niveau événements ou au niveau hybride. Nous montrons les valeurs de ces mesures obtenues par la plate-forme VSIP sur les vidéos étudiées dans la section suivante.

6.3 Bases des données

Afin d'évaluer l'approche proposée, nous employons une base des trajectoires et deux bases de vidéos. Bien que les trajectoires des objets soient extraites à partir des vidéos, l'utilisation d'une base des trajectoires ayant une vérité terrain nous permet d'évaluer séparément la recherche d'objets basée sur les trajectoires.

6.3.1 Bases des trajectoires

La base des trajectoires utilisée comporte 2500 trajectoires divisées en 50 catégories². Chacune des catégories contient donc 50 trajectoires. La figure 6.1 montre quelques trajectoires de cette base.

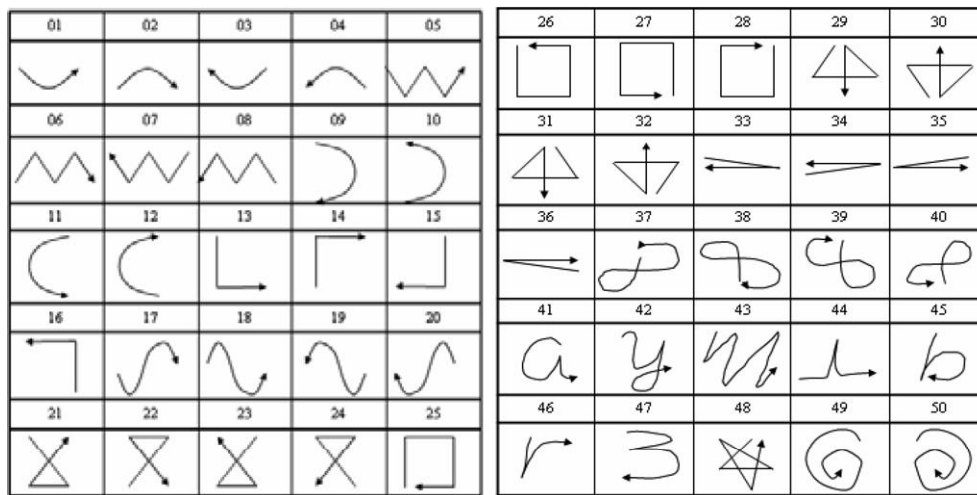


FIG. 6.1 – Trajectoires d'exemple dans la base des trajectoires.

6.3.2 Bases des vidéos

6.3.2.1 Vidéos provenant du projet CARETAKER

Le projet CARETAKER (Content Analysis and REtrieval Technologies to Apply Extraction to massive Recording)³ est un projet Européen avec la participation de 9 partenaires. L'objectif du projet est d'extraire des connaissances structurées dans les grandes bases de documents multimédia acquises par des réseaux de caméras et de microphones installés dans les zones publiques [Patino 2008]. Nous utilisons des vidéos acquises par des caméras installées dans deux stations de métro à Rome et à Turin, en Italie. Les vidéos utilisées sont montrées dans le tableau 6.1. Quelques frames dans ces vidéos sont illustrés dans la figure 6.2. Afin de distinguer les vidéos, nous les nommons. Les vidéos sont analysées en utilisant la plate-forme VSIP. Pour

²<http://mmplab.eed.yzu.edu.tw/trajectory/trajectory.rar>

³<http://www.ist-caremaker.org/>

chacune des vidéos, nous donnons le nombre de frames, la durée (en minutes) et le niveau auquel elle est analysée. Nous distinguons deux niveaux d'analyse : objets (la détection, le suivi et la classification d'objets) et événements (la reconnaissance d'événements).



FIG. 6.2 – Quelques frames dans les vidéos provenant du projet CARETAKER.

TAB. 6.1 – Vidéos provenant du projet CARETAKER.

Nom	Nombre de frames traités	Durée (minutes)	Niveau d'analyse
CARE_1	5454	$\simeq 20$	objets et événements
CARE_2	230250	$\simeq 120$	objets et événements
CARE_3	98980	$\simeq 330$	objets
CARE_4	3965	$\simeq 7$	objets et événements
CARE_5	51580	$\simeq 20$	objets
CARE_6	51450	$\simeq 20$	objets

Ces vidéos ont les caractéristiques suivantes :

- La qualité des images est faible ;
- Le changement d'illumination est considérable dans quelques vidéos ;
- Le contexte est complexe (p. ex. plusieurs objets de contexte et plusieurs types d'objets mobiles) ;
- L'activité humaine est très variée, il existe des occultations des personnes et des objets du contexte, et des interactions entre personnes.

Cinq classes d'objets sont prédéfinies pour les vidéos : Person (une personne), PersonGroup (un groupe de personnes), Crowd (une foule), Luggage (un bagage) et Unknown (un objet appartient à cette classe s'il n'appartient pas à l'une des quatre premières classes). Les résultats détaillés de l'analyse au niveau objets sont montrés dans le tableau 6.2.

Pour les vidéos de CARETAKER, nous avons les événements :

- inside_zone (o, z) : où un objet 'o' dans la zone 'z'

TAB. 6.2 – Résultats de l’analyse des vidéos provenant du projet CARETAKER au niveau objets.

Nom	Objets mobiles				
	Person	PersonGroup	Crowd	Luggage	Unknown
CARE_1	145	60	-	-	-
CARE_2	29	27	16	25	23
CARE_3	9655	-	-	-	102
CARE_4	2311	-	-	-	
CARE_5	777	-	-	-	
CARE_6	810	-	-	-	-

- `stays_inside_zone(o, z, T1)` : où l’événement `inside_zone(o, z)` est détecté pendant `T1` secondes
- `close_to(o, e, D)` : où la distance en 3D entre cet objet et l’équipement ‘e’ est inférieure à un seuil ‘D’.
- `stays_at(o, e, D, T2)` : où l’événement `close_to(o, e, D)` est détecté pendant `T2` secondes.

Au niveau événements, avec les seuils déterminés ($T1=60s$, $D=1.50m$, $T2=5s$), nous précisons :

- `"inside_zone_Platform"` et `"group_inside_zone_Platform"` pour l’événement `"inside_zone"` si la zone `z` est la plate-forme et l’objet `o` appartient aux classes `Person` et `PersonGroup` respectivement ;
- `"stays_inside_zone_Platform"` et `"group_stays_inside_zone_Platform"` pour l’événement `"stays_inside_zone"` si la zone `z` est la plate-forme et l’objet `o` appartient aux classes `Person` et `PersonGroup` respectivement ;
- `"close_to_Gates"` et `"group_close_to_Gates"` pour l’événement `"close_to"` si l’équipement `e` est les portes (`Gates`) et l’objet `o` appartient aux classes `Person` et `PersonGroup` respectivement ;
- `"stays_at_Gates"` et `"group_stays_at_Gates"` pour l’événement `"stays_at"` si l’équipement `e` est les portes (`Gates`) et l’objet `o` appartient aux classes `Person` et `PersonGroup` respectivement ;
- `"close_to_VendingMachine"` et `"group_close_to_VendingMachine"` pour l’événement `"close_to"` si l’équipement `e` est les machines de vente (`VendingMachine1` ou `Vending Machine2`) et l’objet `o` appartient aux classes `Person` et `PersonGroup` respectivement ;

Les résultats de l’analyse au niveau événements de vidéo `CARE_1` et `CARE_2` sont montrés dans les tableaux 6.3 et 6.4.

Il est à noter qu’il y a deux manières de transférer les résultats obtenus par la plate-forme VSIP au niveau événements vers le modèle de données. Comme l’événement est reconnu à chacun instant, la première manière est de considérer chaque

TAB. 6.3 – Résultats de l’analyse de la vidéo CARE_1 provenant du projet CARE-TAKER au niveau événements.

Nom de l’événement	Type d’objet	Nombre d’occurrences
inside_zone_Platform	Person	4460
close_to_Gates	Person	246
stays_at_Gates	Person	114
close_to_VendingMachine1	Person	549
stays_at_VendingMachine1	Person	313
stays_inside_zone_Platform	Person	1134
close_to_VendingMachine2	Person	43
group_close_to_VendingMachine1	PersonGroup	748
group_inside_zone_Platform	PersonGroup	3236
group_stays_at_VendingMachine1	PersonGroup	688
group_stays_inside_zone_Platform	PersonGroup	1360
group_close_to_VendingMachine2	PersonGroup	457
group_stays_at_VendingMachine2	PersonGroup	292
group_close_to_Gates	PersonGroup	314
group_stays_at_Gates	PersonGroup	201

instance reconnue comme un élément "Events" dans notre modèle de données. Cela crée plusieurs éléments "Events" pour un seul objet impliqué dans le même événement à différents moments. Par exemple, 5 instances de l’événement "inside_zone" d’une personne dont l’étiquette est 7 sont reconnues dans 5 frames consécutifs. Avec la première manière, 5 éléments "Events" dont la durée est 1 sont créés et stockés. La deuxième manière regroupe en un seul élément les instances du même événement concernant le même objet reconnues dans des frames consécutifs. Dans l’exemple précédent, la deuxième manière crée un élément "Events" dont la durée est 5. Nous employons les deux manières dans cette thèse. Les tableaux 6.3 et 6.4 sont les résultats de la première manière.

En utilisant les mesures d’évaluation des modules d’analyse vidéo, nous présentons les valeurs de ces mesures obtenues avec la plate-forme VSIP sur la vidéo CARE_6 du projet CARETAKER dans le tableau 6.5. La valeur de persistance d’étiquette est 2.06. Cette valeur montre qu’en moyenne deux objets détectés correspondent à un objet réel. La recherche exacte basée sur l’étiquette ne retrouve donc que la moitié des résultats. Avec la valeur de confusion, nous voyons que certains objets détectés correspondent à plus d’un objet réel (des blobs de l’objet détecté sont des blobs de plusieurs objets réels). La mise en correspondance entre des objets doit pouvoir tenir compte de cette caractéristique.

TAB. 6.4 – Résultats de l'analyse de la vidéo CARE_2 provenant du projet CARE-TAKER au niveau événements.

Nom de l'événement	Type d'objet	Nombre d'occurrences
inside_zone_Platform	Person	20231
stays_inside_zone_Platform	Person	9
stays_at_Gate1	Person	1069
stays_at_Gate2	Person	157
stays_at_Gate3	Person	200
stays_at_Gate4	Person	69
stays_at_Gate5	Person	12
stays_at_Gate6	Person	109
stays_at_Gate7	Person	175
stays_at_Gate8	Person	175
stays_at_Gate9	Person	149
close_to_Gate1	Person	835
close_to_Gate2	Person	1404
close_to_Gate3	Person	1678
close_to_Gate4	Person	761
close_to_Gate5	Person	324
close_to_Gate6	Person	1034
close_to_Gate7	Person	1474
close_to_Gate8	Person	1373
close_to_Gate9	Person	1041
group_inside_zone_Platform	PersonGroup	26390
group_stays_at_Gate1	PersonGroup	626
group_stays_at_Gate2	PersonGroup	1049
group_stays_at_Gate3	PersonGroup	1137
group_stays_at_Gate4	PersonGroup	1116
group_stays_at_Gate5	PersonGroup	1063
group_stays_at_Gate6	PersonGroup	1534
group_stays_at_Gate7	PersonGroup	2570
group_stays_at_Gate8	PersonGroup	2835
group_stays_at_Gate9	PersonGroup	2478
group_close_to_Gate1	PersonGroup	3861
group_close_to_Gate2	PersonGroup	5792
group_close_to_Gate3	PersonGroup	6239
group_close_to_Gate4	PersonGroup	6694
group_close_to_Gate5	PersonGroup	6539
group_close_to_Gate6	PersonGroup	7997
group_close_to_Gate7	PersonGroup	10828
group_close_to_Gate8	PersonGroup	11234
group_close_to_Gate9	PersonGroup	9538

TAB. 6.5 – Valeurs des mesures d’évaluation de la plate-forme VSIP pour la vidéo CARE_6. La valeur de persistance montre qu’en moyenne deux objets détectés correspondent à un objet réel alors que la valeur de confusion indique que certains objets détectés correspondent à plus d’un objet réel.

Vidéo	Objets	Nombre total de boîtes	Nombre de boîtes bien déterminées	Confusion	Persistance
CARE_6	810	35115	32836	1.057	2.06

6.3.2.2 Vidéos provenant du projet CAVIAR

Le projet CAVIAR ⁴ est financé par le projet IST 2001 37540 de *EC’s Information Society Technology*. L’objectif de ce projet est d’améliorer la reconnaissance en utilisant des descriptions locales d’images et des connaissances contextuelles. Ce projet aborde deux applications : l’une est la surveillance du centre ville et l’autre est l’analyse de comportements des clients dans les supermarchés. Deux bases de vidéos correspondant à deux applications dont l’une de l’INRIA à Grenoble en France et l’autre du Portugal sont construites. Les annotations manuelles sont également fournies. Pour ce projet, il y a deux types de classes d’intérêt : les personnes (Person) et les groupes de personnes (PersonGroup). Nous prenons 10 vidéos de la base provenant du Portugal. Les vidéos dans cette base décrivent des activités des personnes ou groupes des personnes dans un supermarché. Deux caméras sont installées : une caméra observe le couloir et l’autre est en face du magasin (voir figure 6.3). La description de ces vidéos est montrée dans le tableau 6.6. Au total, les 10 vidéos contiennent 53 personnes et 9 groupes de personnes. En supposant que l’analyse de ces vidéos est très bonne, nous employons les résultats de la détection et du suivi d’objets des annotations manuelles. L’utilisation de ces vidéos a pour objectif d’évaluer la mise en correspondance entre des objets observés par des caméras différentes, étant sous des conditions différentes (p. ex. changement de la lumière, occultation d’objets).

6.4 Mesures d’évaluation des performances de l’approche d’indexation et de recherche d’information

Nous présentons dans cette section les mesures habituellement utilisées pour l’évaluation des approches d’indexation et de recherche d’information. Avant de donner les définitions de ces mesures, nous introduisons les notions utilisées dans ces définitions. Soit q une requête, N la liste des résultats retrouvés pour la requête q . Les résultats dans la liste sont ordonnés par leurs distances avec la requête.

⁴<http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/>

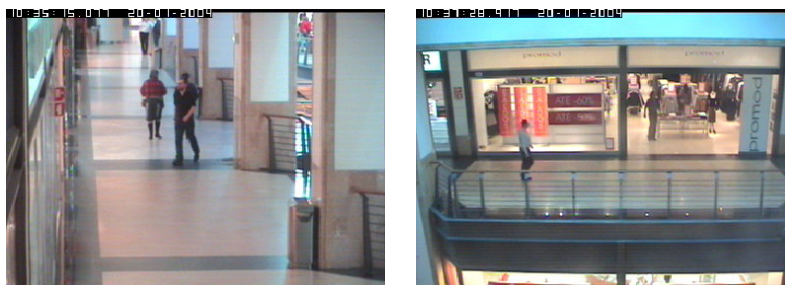


FIG. 6.3 – Supermarché dans le projet CAVIAR est observé par deux caméras : l'une est dans le couloir et l'autre est en face du magasin.

TAB. 6.6 – Dix vidéos provenant du projet CAVIAR.

Nom	Frames	Objets mobiles	
		Person	PersonGroup
CAVIAR_1	2360	17	3
CAVIAR_2	2360	4	1
CAVIAR_3	383	5	1
CAVIAR_4	383	4	1
CAVIAR_5	485	4	1
CAVIAR_6	485	3	0
CAVIAR_7	295	6	1
CAVIAR_8	295	3	0
CAVIAR_9	1119	5	1
CAVIAR_10	1119	1	0

- N_{rel} est l'ensemble des résultats pertinents de la base de données pour la requête q , $|N_{rel}|$ est la cardinalité de N_{rel} ;
- N_{retr} est l'ensemble des résultats retrouvés ;
- m est l'ensemble des premiers résultats dans la liste N de la requête q , $1 \leq |m| \leq |N|$;
- n est l'ensemble des résultats pertinents dans l'ensemble m ;
- R_i est le rang du résultat pertinent i dans la liste des résultats N .

6.4.1 Rappel et précision

Le rappel et la précision sont utilisés en indexation et recherche d'information [Kent 1955]. Le rappel pour une requête (cf. équation 6.1) mesure le pourcentage de résultats pertinents dans l'ensemble de résultats retrouvés par rapport au nombre de résultats pertinents de la base de données.

$$Rappel = \frac{|N_{rel} \cap N_{retr}|}{|N_{rel}|} \quad (6.1)$$

La précision (cf. équation 6.2) mesure le pourcentage de résultats pertinents de l'ensemble de résultats trouvés par rapport au nombre de résultats retrouvés.

$$Precision = \frac{|N_{rel} \cap N_{retr}|}{|N_{retr}|} \quad (6.2)$$

Correspondant à chaque ensemble des résultats retrouvés, une paire de rappel et précision est calculée. En changeant l'ensemble des résultats retrouvés, on peut obtenir une courbe de rappel et précision. Pour évaluer la performance d'une approche, on analyse la forme de la courbe et les valeurs de *Precision* correspondant à quelques valeurs spéciales de *Rappel*. Pour comparer les performances de deux approches, correspondant à une valeur de *Rappel*, l'approche qui obtient la valeur de *Precision* la plus élevée est la plus efficace.

6.4.2 Rang normalisé moyen

Le rang normalisé moyen a été proposé par Muller et al. [Müller 2002]. Il est défini par l'équation 6.3. L'idée du rang normalisé moyen est qu'une approche d'indexation et de recherche est efficace si elle trouve les résultats pertinents dans les premiers résultats.

$$\widetilde{Rang} = \frac{1}{|N| * |N_{rel}|} \left(\sum_{i=1}^{|N_{rel}|} (R_i) - \frac{(|N_{rel}| * (|N_{rel}| + 1))}{2} \right) \quad (6.3)$$

La valeur de \widetilde{Rang} est entre 0 et 1. Plus la valeur de \widetilde{Rang} est petite, plus l'approche est efficace.

6.4.3 Matrice de confusion

En plus des mesures de rappel et précision et de rang normalisé moyen, nous utilisons d'autres mesures d'évaluation provenant de la reconnaissance. La matrice de confusion est constituée de quatre mesures : TP (true positive), FP (false positive), TN (true negative) et FN (false negative).

À la différence de l'indexation et de la recherche d'images, la contrainte sur les rangs des résultats retrouvés en indexation et recherche en vidéosurveillance est plus forte en raison de deux facteurs. Le premier facteur est que les résultats de recherche dans ce domaine concernent la sécurité. Dans certains cas, la recherche est urgente (p. ex. le vol, la bagarre dans les stations des métros). Le personnel de sécurité doit recevoir les résultats pertinents dans un petit ensemble des premiers résultats. Il ne peut pas prendre son temps pour naviguer une grande liste de résultats. Le deuxième facteur est que le personnel de sécurité est impatient. Si les premiers résultats ne sont pas pertinents, il n'a pas la volonté ou le temps pour voir d'autres résultats retrouvés. Afin d'évaluer la performance des approches d'indexation et de recherche pour les situations urgentes, nous calculons les valeurs de TP et FP dans les m premiers résultats retrouvés. La valeur de TP mesure le nombre de résultats pertinents alors que la valeur de FP montre le nombre de résultats non pertinents dans les m premiers résultats. La valeur choisie de m est petite. Nous évaluons des résultats avec quatre valeurs de m (de 1 à 4).

Les mesures TN et FN peuvent être utilisées afin d'évaluer la performance des approches d'indexation et de recherche pour d'autres situations. Par exemple, dans un système de surveillance de bagages, la mesure FN compte le nombre de bombes qui ne sont pas détectées.

6.4.4 Discussions

Nous discutons dans ce paragraphe l'évaluation de performance des approches d'indexation et de recherche de vidéos de vidéosurveillance. Trois types de mesures d'évaluation sont présentés.

À partir des définitions de ces mesures, pour pouvoir évaluer les performances des approches d'indexation et de recherche d'information, il faut disposer la vérité terrain pour chacune des requêtes. Cette vérité terrain correspondant à une requête comporte des informations telles que les résultats possibles dont des résultats pertinents et non pertinents. Les vérités terrain sont faites manuellement. Cela demande un travail considérable lorsque les résultats possibles sont grands. De plus, le jugement d'un résultat pertinent ou non pertinent est subjectif. Heureusement, ce jugement pour la recherche de vidéos de vidéosurveillance est plus facile que celui pour la recherche d'images car les objets pertinents et l'objet recherché représentent le même objet réel observé à différents moments et/ou par différentes caméras.

Afin de comparer les approches d'indexation et de recherche de vidéos, il faut calculer les mesures d'évaluation sur des bases de vidéos communes et des requêtes communes associées aux vérités terrain. Des bases de vidéo communes sont déjà

disponibles pour l'indexation et la recherche de journaux télévisés. Jusqu'à présent, il n'existe pas encore de bases de vidéos communes pour l'indexation et la recherche de vidéos pour la vidéosurveillance. Cela nous rend difficile l'évaluation de notre approche et sa comparaison avec d'autres approches. Il est à noter que TRECVID en 2008 a fourni une base de vidéos de vidéosurveillance afin d'évaluer des modules d'analyse vidéo. Nous espérons qu'une base de vidéos permettant d'évaluer l'indexation et la recherche de vidéos pour la vidéosurveillance sera disponible dans un futur proche.

Le choix de la mesure dépend de l'objectif de l'évaluation. La courbe de rappel et précision présente la distribution de performance d'une approche en fonction du nombre de résultats retrouvés tandis que le rang normalisé moyen rend une seule valeur pour chaque requête. Le rang normalisé moyen permet donc d'avoir une comparaison globale des approches. Les mesures de TP et FP déterminées sur les m premiers résultats se focalisent sur la performance de l'approche sur un petit nombre des premiers résultats. Dans cette thèse, nous employons les trois types de mesures.

6.5 Analyse de performance de l'approche proposée

Les prochaines sections sont dédiées à l'évaluation de notre approche.

Nous présentons d'abord l'utilisation du langage de requêtes (cf. section 6.6). Nous décrivons l'expression du langage de requêtes. Nous mettons en évidence les points forts et aussi que les points faibles du langage proposé.

Ensuite, nous analysons les résultats obtenus avec les deux méthodes proposées pour la détection des blobs représentatifs (une basée sur le changement d'apparence et l'autre basée sur le regroupement des blobs) dans la section 6.7. La détection des blobs représentatifs basée sur le regroupement des blobs est une amélioration de la méthode présentée par Ma et al. [Ma 2007]. Nous la comparons avec celle de Ma et al.

Puis, comme notre approche permet de faire l'indexation et la recherche à trois niveaux (objets, événements et combinaison d'objets et d'événements), nous analysons séparément les performances à chacun des niveaux.

Concernant la recherche d'objets (cf. section 6.8), nous analysons les difficultés rencontrées en indexation et recherche d'objets dans les vidéos de vidéosurveillance. Nous comparons la performance de la méthode de mise en correspondance proposée avec deux méthodes de l'état de l'art dont l'une de Ma et al. [Ma 2007] et l'une de Calderara et al. [Calderara 2006]. Puis, nous évaluons la performance des descripteurs d'apparence en indexation et recherche d'objets dans les vidéos de vidéosurveillance. Les premiers résultats obtenus avec la mise en correspondance entre des trajectoires sont également montrés.

Concernant la recherche d'événements et celle d'objets et d'événements (cf. section 6.9), nous montrons comment l'approche proposée permet de retrouver de nouveaux événements ou d'objets impliqués dans un événement particulier ou d'événements d'un objet particulier.

Enfin, les résultats obtenus avec les deux méthodes de retour de pertinence dont l'une basée sur plusieurs d'images d'exemple et l'une basée sur les SVM sont analysés.

6.6 Utilisation du langage de requêtes

Nous visons dans cette thèse deux applications : la recherche d'objets et d'événements pour la vidéosurveillance et l'étude statistique (cf. chapitre 1). Les tâches de la phase d'indexation et de recherche (cf. chapitres 4 et 5) telles que la représentation d'objets, l'extraction de descripteurs, la mise en correspondance décident de la qualité de recherche de l'approche. Le langage de requêtes est un outil qui permet à l'utilisateur d'accéder au système. Un langage de requêtes est évalué par son expressivité et sa facilité. Cette section vise à analyser l'expressivité et la facilité du langage de requêtes.

6.6.1 Analyse d'effort demandé à l'utilisateur et expressivité du langage

Les requêtes exprimées dans ce langage sont à plusieurs niveaux (images, objets et événements). Les requêtes ont deux objectifs : la sécurité et l'étude statistique. Les relations entre les objets et les événements peuvent être exprimées par 13 opérateurs temporels, 7 opérateurs d'apparence (les couleurs dominantes, les matrices de covariance, les histogrammes de contours, les points d'intérêts MSER associés au descripteur SIFT, ceux de HarrisAffine et ceux de DoG), 6 opérateurs de comparaison des valeurs numériques et des chaînes de caractères ($=$, \neq , $>$, $<$, \geq , \leq), 4 opérateurs arithmétiques ($+$, $-$, $*$, $/$), 3 opérateurs d'appartenance et 18 fonctions d'accès qui sont présentés en annexe A.

Afin d'analyser l'effort demandé pour exprimer les requêtes et l'expressivité du langage proposé, nous divisons les requêtes en 5 catégories appelées C1, C2, C3, C4 et C5. Nous analysons dans les sections suivantes l'effort demandé pour chaque catégorie correspondant à trois niveaux d'analyse vidéo. Le premier niveau (N1) contient la détection et le suivi d'objets. Le deuxième niveau (N2) rajoute la classification d'objets. Le troisième niveau (N3) contient toutes les analyses (la détection, le suivi, la classification d'objets et la reconnaissance d'événements). Le tableau 6.7 résume ces analyses. Les analyses montrent que le langage de requêtes permet à l'utilisateur d'exprimer des requêtes des catégories C1, C2, C3, C4 et C5 avec peu ou beaucoup d'effort selon le niveau d'analyse.

6.6.2 Requêtes concernant des images d'exemple et des objets (C1)

Les requêtes de la catégorie C1 visent à retrouver des objets indexés par une image d'exemple. Les requêtes dans cette catégorie peuvent être classées en deux sous-catégories C11 et C12.

TAB. 6.7 – Analyse de l’effort demandé pour chaque catégorie de requête correspondant à 3 niveaux d’analyse vidéo. Le premier niveau (N1) contient la détection et le suivi d’objets. Le deuxième niveau (N2) rajoute la classification d’objets. Le troisième niveau (N3) contient toutes les analyses (la détection, le suivi, la classification d’objets et la reconnaissance d’événements). Le symbole \vee montre la présence d’un niveau d’analyse vidéo. Si un niveau d’analyse n’est pas disponible, nous utilisons le symbole \emptyset tandis que si ce niveau n’est pas concerné, nous employons le symbole $-$. L’effort demandé est divisé en trois niveaux : peu (la requête est facilement exprimée), beaucoup (il est faisable mais difficile d’exprimer la requête) et indéfini (c’est impossible à exprimer la requête). Le symbole \times indique le niveau de l’effort demandé.

Catégorie	Niveau d’analyse vidéo			Effort		
	N1	N2	N3	peu	beaucoup	indéfini
C11	\vee	\emptyset	-	\times		
	\vee	\vee	-	\times		
C12	\vee	\emptyset	-	\times		
	\vee	\vee	-	\times		
C21	\vee	\emptyset	-	\times		
	\vee	\vee	-	\times		
C22	\vee	\emptyset	-	\times		
	\vee	\vee	-	\times		
C31	\vee	\vee	\vee	\times		
	\vee	\vee	\emptyset		\times	
C32	\vee	\vee	\vee	\times		
	\vee	\vee	\emptyset		\times	
C41	\vee	\vee	\vee	\times		
	\vee	\vee	\emptyset		\times	
C42	\vee	\vee	\vee	\times		
	\vee	\vee	\emptyset		\times	
C51	\vee	\vee	\vee	\times		
	\vee	\vee	\emptyset		\times	
C52	\vee	\vee	\vee	\times		
	\vee	\vee	\emptyset		\times	

C11 : les requêtes concernant des images d'exemple et des objets appartenant à une classe déterminée

La requête : "Retrouver des personnes qui sont similaires à une image d'exemple par les histogrammes de contours" est exprimée par :

```
SELECT * FROM Video WHERE : ENTITIES ((s : SubImage), (p : PhysicalObjects)) CONDITIONS ((p's Class = "Person")(s EH_matching p))
```

L'opérateur *EH_matching* indique que l'image d'exemple *s* va être comparée avec les personnes *p* de la base de données par l'histogramme de contours. Cette requête peut être exprimée dans le langage. Le résultat de la recherche est cependant vide si la classification d'objets n'est pas disponible dans des modules d'analyse.

C12 : les requêtes concernant des images d'exemple et des objets, la classe d'objet n'est pas indiquée

Les requêtes dans cette catégorie sont équivalentes aux requêtes de la catégorie C11. Le résultat de ces requêtes n'est pas vide même si la classification n'est pas disponible. Cependant, le résultat peut être non pertinent car tous les types d'objets sont utilisés pour la mise en correspondance avec l'image d'exemple. La requête "Retrouver des objets qui sont similaires à une image d'exemple par les histogrammes de contours" est exprimée par :

```
SELECT * FROM Video WHERE : ENTITIES ((s : SubImage), (p : PhysicalObjects)) CONDITIONS ((s EH_matching p))
```

Les requêtes dans la catégorie C1 sont exprimées en nécessitant peu d'effort de l'utilisateur.

6.6.3 Requêtes concernant des objets (C2)

Les requêtes de la catégorie C2 retrouvent des objets ayant certaines caractéristiques. Elles peuvent être classées de la même manière que les requêtes de la catégorie C1.

C21 : les requêtes concernant des objets appartenant à une classe déterminée

La requête suivante retrouve les personnes qui sont présentes dans la scène pendant plus de 50 frames :

```
SELECT * FROM Video WHERE : ENTITIES ((p : PhysicalObjects)) CONDITIONS ((p's Class = "Person")(p's Duration >50))
```

C22 : les requêtes concernant des objets dont la classe n'est pas déterminée

Si la classification n'est pas disponible, la requête suivante donne tous les objets qui sont présents dans la scène plus de 50 frames :

```
SELECT * FROM Video WHERE : ENTITIES ((p : PhysicalObjects)) CONDITIONS ((p's Duration >50))
```

Nous voyons qu'il est facile d'exprimer des requêtes de la catégorie C2.

6.6.4 Requêtes concernant des événements (C3)

La catégorie C3 contient des requêtes permettant de retrouver des événements ayant certaines caractéristiques. Les requêtes de cette catégorie sont classées en deux sous-catégories C31 et C32.

C31 : les requêtes concernant un seul événement

Les deux requêtes suivantes vérifient s'il y a des personnes qui sont proches de la machine de vente 1 dans deux cas : avec ou sans reconnaissance d'événements. Dans le deuxième cas, il faut que l'utilisateur puisse définir l'événement *close_to_VendingMachine1* en se basant sur la distance entre l'objet et la machine de vente 1.

```
SELECT e FROM Video WHERE : ENTITIES ((e : Events)) CONDITIONS
((e's Name = "close_to_VendingMachine1"))
```

```
SELECT p FROM Video WHERE : ENTITIES ((z : PhysicalObjects), (p :
PhysicalObjects)) CONDITIONS ((z's Name = "VendingMachine1")(p's Class =
"Person")(z distance p <1.5))
```

C32 : les requêtes concernant plus d'un événement et leurs relations temporelles

La requête qui vérifie : "La personne va à la machine 1. Après certain moments (5 frames), elle s'approche de la machine 2." est définie dans le langage :

```
SELECT * FROM Video WHERE : ENTITIES ((e1 : Events), (e2 : Events))
CONDITIONS ((e1's Name = "close_to_VendingMachine1")
(e2's Name = "close_to_VendingMachine2") (e1 having_same_objects e2) (e2's
_i - e1's i_ >5))
```

où *_i* et *_i* sont des fonctions d'accès aux points limites de l'intervalle de temps. Cela ne nécessite pas besoin beaucoup d'effort de l'utilisateur si la reconnaissance d'événement est présente.

Si la reconnaissance d'événements n'est pas disponible, l'utilisateur doit définir deux événements *close_to_VendingMachine1* et *close_to_VendingMachine2* (voir les requêtes de la catégorie C31 quand la reconnaissance n'est pas disponible) et indiquer la contrainte sur leurs points limites. L'expression dans ce cas demande beaucoup d'effort de l'utilisateur.

6.6.5 Requêtes concernant des objets et des événements (C4)

Les requêtes appartenant à cette catégorie permettent de retrouver des objets impliqués dans un événement particulier ou des événements d'un objet particulier. Nous divisons également ces requêtes en deux sous-catégories C41 et C42.

C41 : les requêtes concernant un seul événement

La requête permettant retrouver des personnes indexées impliquées dans l'événement *inside_zone* est exprimée de deux manières selon la présence de la reconnaissance d'événements. Si la reconnaissance est disponible, la requête est :

```
SELECT p FROM Video WHERE : ENTITIES ((p : PhysicalObjects) (e :
Events)) CONDITIONS ((p's Class= "Person") (p involved_in e) (e's Name ="in-
```

side_zone"))

Sinon, cette requête devient :

```
SELECT p FROM Video WHERE : ENTITIES ((p : PhysicalObjects) (z : PhysicalObjects)) CONDITIONS ((p's Class = "Person") (z's Name = "Platform") (p in z))
```

L'utilisateur doit arriver à définir lui-même l'événement *inside_zone*.

C42 : les requêtes concernant plus d'un événement et leurs relations temporelles

Quand la reconnaissance d'événement est disponible, la requête "retrouver des personnes qui se déplacent de la machine de vente 1 à la machine de vente 2" est exprimée par :

```
SELECT p FROM Video WHERE : ENTITIES ((e1 : Events), (e2 : Events), (p : PhysicalObjects)) CONDITIONS ((e1's Name = "close_to_VendingMachine1") (e2's Name = "close_to_VendingMachine2") (p's Class = "Person") (p involved_in e1) (p involved_in e2) (e1 before e2))
```

Si la reconnaissance d'événements n'est pas disponible, l'utilisateur doit définir deux événements *close_to_VendingMachine1* et *close_to_VendingMachine2* (voir les requêtes de la catégorie C31 quand la reconnaissance n'est pas disponible) et indiquer leur relation temporelle. L'expression dans ce cas demande beaucoup d'effort de l'utilisateur.

6.6.6 Requêtes concernant des images d'exemple, des objet et des événement (C5)

Les composants des requêtes de cette catégorie contiennent des images d'exemple, des objets et des événements.

C51 : les requêtes concernant un seul événement

Dans le cas où plusieurs instances de l'événement "close_to_VendingMachine1" sont reconnues. Le personnel de sécurité peut s'intéresser à retrouver les personnes impliquées dans cet événement qui sont semblables à une image d'exemple par les matrices de covariance. Cette requête est aisément exprimée si les modules d'analyse ont la reconnaissance d'événements.

```
SELECT * FROM Video WHERE : ENTITIES ((p : PhysicalObjects), (i : SubImage), (e : Events)) CONDITIONS ((e's Name = "close_to_VendingMachine1") (p involved_in e) (p's Class = "Person") (i Visual_matching p))
```

Sinon, cette requête doit définir l'événement "close_to_VendingMachine1" en se basant sur la distance entre l'objet et la machine de vente 1. Dans ce cas, plus d'effort de l'utilisateur est demandé.

```
SELECT * FROM Video WHERE : ENTITIES ((p : PhysicalObjects), (i : SubImage), (z : PhysicalObjects)) CONDITIONS ((z's Name = "VendingMachine1") (z distance p < 1.5) (p's Class = "Person") (i Visual_matching p))
```

C52 : les requêtes concernant plus d'un événement et leurs relations temporelles

Les requêtes concernant plus d'un événement et leurs relations temporelles peuvent

être exprimées sans difficulté dans le langage si la reconnaissance d'événements est disponible. La requête "retrouver des personnes en se déplaçant de la machine de vente 1 à la machine de vente 2 sont similaires à une image d'exemple" est :

```
SELECT * FROM Video WHERE : ENTITIES ((p : PhysicalObjects),(i : SubImage), (e1 : Events), (e2 : Events)) CONDITIONS ((p's Class = "Person") (e1's Name = "close_to_VendingMachine1")(p involved_in e1) (p involved_in e2) (e2's Name = "close_to_VendingMachine2")(i Visual_matching p) (e1 before e2))
```

Comme nous l'expliquons dans les sections précédentes, l'expression des requêtes des catégories C32 et C42 quand la reconnaissance d'événements n'est pas disponible demande beaucoup d'effort de l'utilisateur. L'expression des requêtes de la catégorie C52 est équivalente à celle des requêtes des catégories C32 et C42. L'utilisateur doit définir des événements, indiquer leurs relations temporelles et déterminer la similarité entre des images d'exemple et des objets impliqués dans ces événements.

6.6.7 Facilité du langage de requêtes

Le syntaxe du langage de requêtes est similaire à celui du langage SQL. Ce langage est relativement facile pour l'utilisateur qui est familier avec le langage SQL. Afin d'aider l'utilisateur à exprimer ses requêtes, une liste de fonctions d'accès et d'opérateurs (apparence, temporel) est fournie. Pour chacune des vidéos, une liste de classes pour les objets (si la classification d'objets est disponible) et celle de noms pour les événements (si la reconnaissance d'événements est disponible) doivent être montrées à l'utilisateur.

6.6.8 Discussions

Nous montrons dans cette section que le langage de requêtes permet d'exprimer plusieurs requêtes. Cela montre son expressivité. Concernant la facilité du langage, une interface qui permet de compléter automatiquement quelques parties de la requête est nécessaire pour les utilisateurs novices. Cependant, les utilisateurs potentiels de ce langage sont les personnels de sécurité, une formation courte peut être fournie. Le langage de requête a une limitation. Il n'est pas hiérarchique. Une requête complexe ne peut pas être exprimée en combinant quelques requêtes simples prédéfinies.

6.7 Évaluation de la détection des blobs représentatifs

Dans cette section, nous présentons les résultats obtenus avec nos deux méthodes de détection des blobs représentatifs : l'une basée sur le changement d'apparence et l'autre basée sur le regroupement des blobs. La méthode basée sur le regroupement des blobs est une amélioration de la méthode de Ma et al. [Ma 2007]. Nous comparons la performance de ces deux méthodes.

6.7.1 Résultats de la détection des blobs représentatifs

Deux méthodes de détection des blobs représentatifs sont utilisées dans la tâche de représentation des objets de la phase d'indexation. Comme indiqué dans le chapitre 4, le choix de méthode se base sur la qualité des modules d'analyse vidéo. Si les modules d'analyse vidéos obtiennent des résultats fiables, la méthode basée sur le changement d'apparence est employée et inversement. Le nombre de blobs est fixé par le choix du seuil pour la méthode basée sur le changement d'apparence (voir algorithme 9 de la section 4.5 du chapitre 4) et le nombre de groupes pour celle basée sur le regroupement des blobs (voir algorithme 10 de la section 4.5 du chapitre 4). Le tableau 6.8 montre les résultats obtenus avec nos deux méthodes de détection des blobs représentatifs avec différents descripteurs (les points d'intérêt, les histogrammes de contours et les matrices de covariance). Les objets provenant du projet CAVIAR sont bien détectés et suivis, nous employons la détection basée sur le changement d'apparence tandis que la détection basée sur le regroupement des blobs est utilisée pour les vidéos provenant du projet CARETAKER qui sont analysées par la plate-forme VSIP. Les figures 6.4, 6.5, 6.6 montrent trois résultats obtenus.

Concernant le temps de calcul, la méthode de détection basée sur le changement d'apparence est plus rapide que celle basée sur le regroupement des blobs. Pour un objet contenant N blobs, la méthode basée sur le changement d'apparence demande $(N - 1)$ comparaisons des blobs consécutifs tandis que la méthode basée sur le regroupement des blobs fait $\frac{N*(N-1)}{2}$ comparaisons de blobs pour le regroupement agglomératif en plus du temps d'appeler les SVM pour les classifier en blobs avec objets et sans objet.

En résumé, la méthode de détection basée sur le changement d'apparence est en ligne, simple, rapide tandis que celle basée sur le regroupement des blobs est hors ligne, efficace à l'imperfection de la détection et du suivi d'objets mais coûteuse surtout si le nombre de blobs d'un objet est élevé. La propriété en ligne de la méthode de détection basée sur le changement d'apparence est précieuse dans certaines applications. Nous les analysons dans le chapitre 7.

La performance des deux méthodes dépend beaucoup des descripteurs d'apparence utilisés. L'évaluation des descripteurs d'apparence en indexation et recherche de vidéos pour la vidéosurveillance qui sera présentée dans la section suivante nous permet de choisir le descripteur approprié.

6.7.2 Comparaison de la méthode de détection des blobs représentatifs basée sur le regroupement des blobs et celle de Ma et al.

La méthode de détection des blobs représentatifs basée sur le regroupement des blobs est une amélioration de celle de Ma et al. [Ma 2007], [Cohen 2008]. Afin de comparer les deux méthodes, nous effectuons deux expérimentations. La première expérimentation vise à comparer les résultats obtenus par les deux méthodes où les

TAB. 6.8 – Résultats obtenus de la détection des blobs représentatifs, les méthodes 1 et 2 sont la méthode basée sur le changement d'apparence et celle basée sur le regroupement des blobs respectivement.

Vidéos	Méthode	Nombre d'objets	Nombre de blobs	Nombre de blobs représentatifs
CAVIAR	1	9	5352	85
CARE_1	2	145	4920	537
CARE_2	2	29	29740	45
CARE_4	1	2311	64915	8426
	2	2311	64915	5792
CARE_5	2	777	14909	8874
CARE_6	2	810	35115	2512



FIG. 6.4 – Une personne détectée et suivie pendant 905 frames de la vidéo CARE_5, 4 blobs représentatifs sont identifiés pour cette personne. Les images en haut sont les images de la scène avec la présence de la personne. Les images en bas sont les blobs représentatifs de la personne. Tous les quatre blobs représentatifs sont pertinents. Ils représentent des aspects différents de la personne.



FIG. 6.5 – Une personne détectée et suivie pendant 95 frames de la vidéo CARE_5, 3 blobs représentatifs sont identifiés pour cette personne. Les images en haut sont les images de la scène avec la présence de la personne. Les images en bas sont les blobs représentatifs de la personne. Tous les trois blobs représentatifs sont pertinents. Les blobs représentent des aspects différents (avec ou sans présence d'autres personnes).



FIG. 6.6 – Une personne détectée et suivie pendant 175 frames de la vidéo CARE_6, 4 blobs représentatifs sont identifiés pour cette personne. Les images en haut sont les images de la scène avec la présence de la personne alors que les images en bas sont les blobs représentatifs de la personne. Tous les quatre blobs représentatifs sont pertinents. Ils représentent les aspects différents de la personne. Cependant, la détection de la personne n'est pas bonne, la personne est entièrement présente dans un seul blob parmi 4 blobs représentatifs.

machines à vecteurs de support sont entraînées sur des exemples provenant de la même vidéo sur laquelle deux méthodes s'effectuent. La deuxième expérimentation évalue les deux méthodes en réutilisant les machines à vecteurs de support entraînées dans la première expérimentation sur une autre vidéo.

Nous rappelons que dans le chapitre 5, nous présentons deux mesures d'évaluation pour les méthodes de détection des blobs représentatifs. Pour un objet O , ces deux mesures sont définies par :

$$\begin{aligned} F(O) &= \frac{n*100\%}{N} \\ P(O) &= \frac{na*100\%}{n} \end{aligned} \quad (6.4)$$

où n est le nombre de blobs représentatifs déterminés pour l'objet O , N est le nombre de blobs de l'objet O avant de détecter les blobs représentatifs, na est le nombre de blobs pertinents parmi les n blobs. La mesure $F(O)$ exprime la capacité de réduire les informations à stocker et à calculer tandis que la mesure $P(O)$ montre la capacité à corriger les erreurs produites par la détection et le suivi d'objets. Un algorithme de détection des blobs représentatifs est efficace s'il obtient une petite valeur de F et une grande valeur de P .

Nous implémentons la méthode de Ma et al. exactement comme présentée dans [Ma 2007]. La matrice de covariance choisie est une matrice de 11×11 contenant : les positions x , y , les couleurs r , g , b , les valeurs des gradients de r , g , b (cf. équations 2.45 et 2.46). Les matrices de covariance sont extraites en appliquant l'algorithme présenté dans [Porikli 2006a] et [Porikli 2006b] qui permet de les calculer rapidement. Cet algorithme se base sur les images d'intégral [Viola 2004].

Pour les deux expérimentations, si le nombre de blobs d'un objet est petit (5 est choisi), nous gardons tous les blobs de cet objet comme blobs représentatifs. Sinon, le nombre de groupes choisi est 5 pour les deux expérimentations. Pour chacun des groupes, un blob représentatif est déterminé. Nous calculons les valeurs de F et P (cf. équation 6.4) pour chacun des objets en utilisant la vérité terrain manuellement préparée.

La première expérimentation est effectuée sur la vidéo CARE_6 (cf. tableau 6.1) du projet CARETAKER contenant 810 objets avec 35115 blobs. Les machines à vecteurs de support sont entraînées par 100 blobs dont 50 blobs avec objets et 50 blobs sans objets. Les SVM nous permettent d'enlever 135 objets qui sont entièrement mal détectés (tous les blobs de cet objet sont déterminés comme blobs sans objet par les SVM). La méthode de Ma obtient 3250 blobs représentatifs tandis que la nôtre détecte 2512 blobs représentatifs. Les figures 6.7, 6.8 illustrent les valeurs de la mesure F et P obtenues avec les deux méthodes si elles sont différentes. Parmi 675 objets, notre méthode obtient de meilleurs résultats pour la mesure F sur 96 objets et pour la mesure P sur 157 objets cependant que la méthode de Ma et al. a de meilleurs résultats pour la mesure F sur 47 objets et pour la mesure P sur 49 objets. Pour d'autres objets, les valeurs de F et P obtenues par les deux méthodes sont égales. Dans certains cas, notre approche n'est pas plus performante que la méthode de Ma et al., la raison est que les objets dans ces cas sont mal détectés, les SVM enlèvent donc la majorité de leurs blobs. Le nombre de blobs de ces objets

après avoir appelé les SVM est inférieur à 5 (le seuil prédéterminé), notre méthode les prend comme blobs représentatifs sans appeler le regroupement agglomératif. Cependant la classification basée sur les SVM n'est pas toujours parfaite, ces blobs représentatifs peuvent être sans objet.

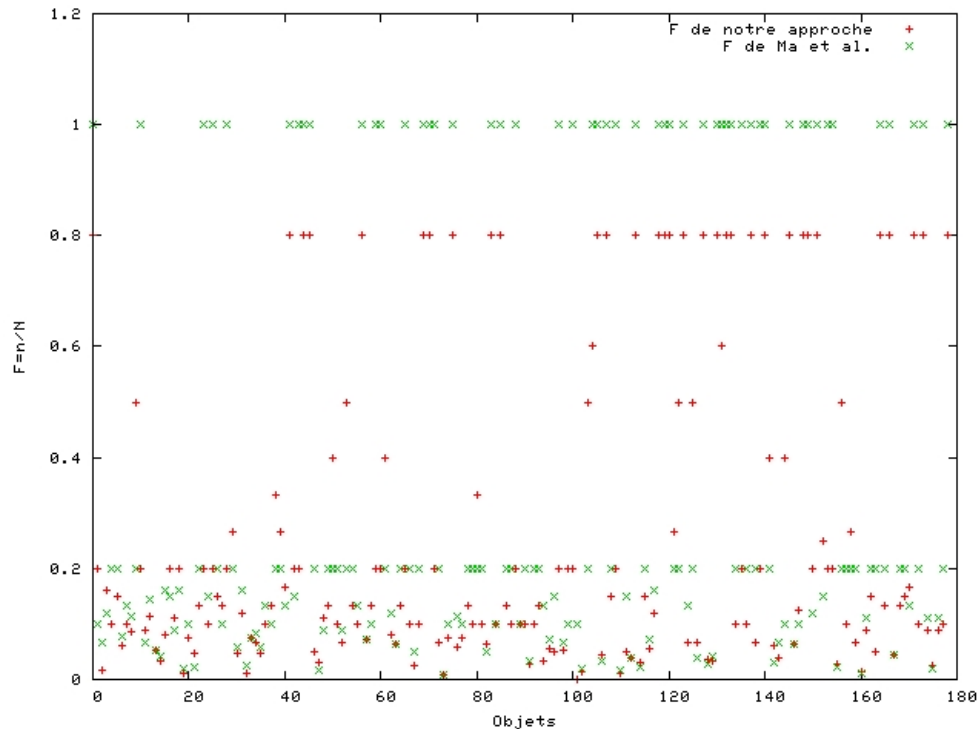


FIG. 6.7 – Valeurs de la mesure F obtenues par les deux méthodes dans la première expérimentation. Cette mesure exprime la capacité de réduire les informations à stocker. Une méthode est efficace si elle obtient une petite valeur de F . La valeur de F peut être de 0% à 100% (respectivement de 0 à 1 dans cette figure). Parmi 675 objets, la valeur de F obtenue par notre méthode est plus petite que celle obtenue par la méthode de Ma et al. sur 96 objets alors que la méthode de Ma et al. a de meilleurs résultats sur 47 objets.

La deuxième expérimentation est effectuée sur la vidéo CARE_5 du projet CARETAKER contenant 777 objets avec 14909 blobs. En utilisant les SVM entraînés dans la première expérimentation sur 14909 blobs, nous obtenons 661 objets contenant 8874 blobs. Les SVM nous permettent d'enlever 116 objets qui ne sont pas bien détectés et suivis. La méthode de Ma et al. nous donne 2060 blobs tandis que notre approche rend 1664 blobs. Parmi 661 objets, notre approche obtient de meilleurs résultats de la mesure F sur 131 objets et de la mesure P sur 142 objets (respectivement 41 et 75 objets pour la méthode de Ma et al.). Les figures 6.9, 6.10 illustrent les valeurs de la mesure F et P obtenues par les deux méthodes si elles sont différentes.

Les valeurs moyennes des mesures F et P obtenues par les deux méthodes dans

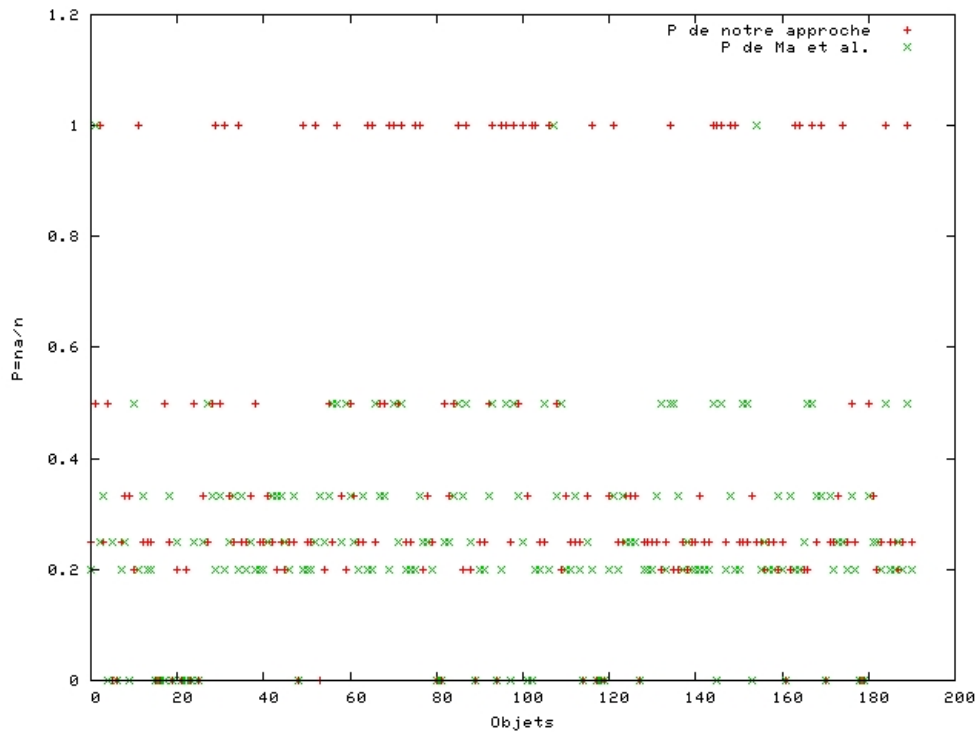


FIG. 6.8 – Valeurs de la mesure P obtenues par les deux méthodes dans la première expérimentation. La mesure P montre la capacité à corriger les erreurs produites par la détection et le suivi d'objets. Une méthode est efficace si elle obtient une grande valeur de P . La valeur de P peut être de 0% à 100% (respectivement de 0 à 1 dans cette figure). Parmi 675 objets, la valeur de P obtenue par notre méthode est plus élevée que celle obtenue par la méthode de Ma et al. sur 157 objets alors que la méthode de Ma et al. a de meilleurs résultats sur 49 objets.

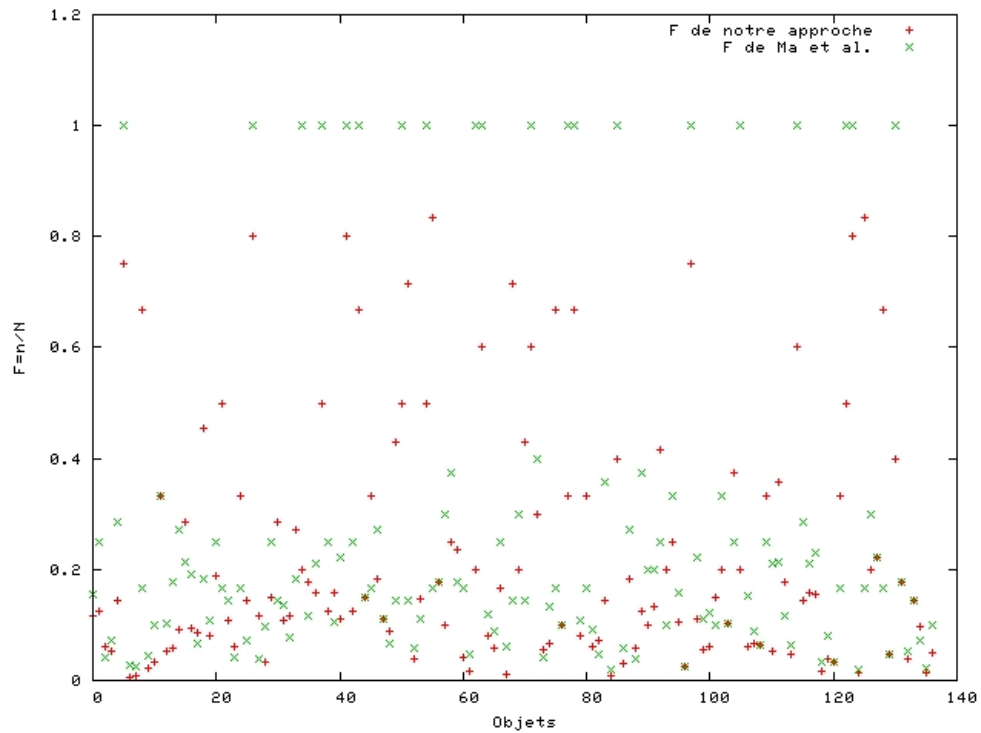


FIG. 6.9 – Valeurs de la mesure F obtenues par les deux méthodes dans la deuxième expérimentation. Cette mesure exprime la capacité de réduire les informations à stocker. Une méthode est efficace si elle obtient une petite valeur de F . La valeur de F peut être de 0% à 100% (respectivement de 0 à 1 dans cette figure). Parmi 661 objets, la valeur de F obtenue par notre méthode est plus petite que celle obtenue par la méthode de Ma et al. sur 131 objets alors que la méthode de Ma et al. a de meilleurs résultats sur 41 objets.

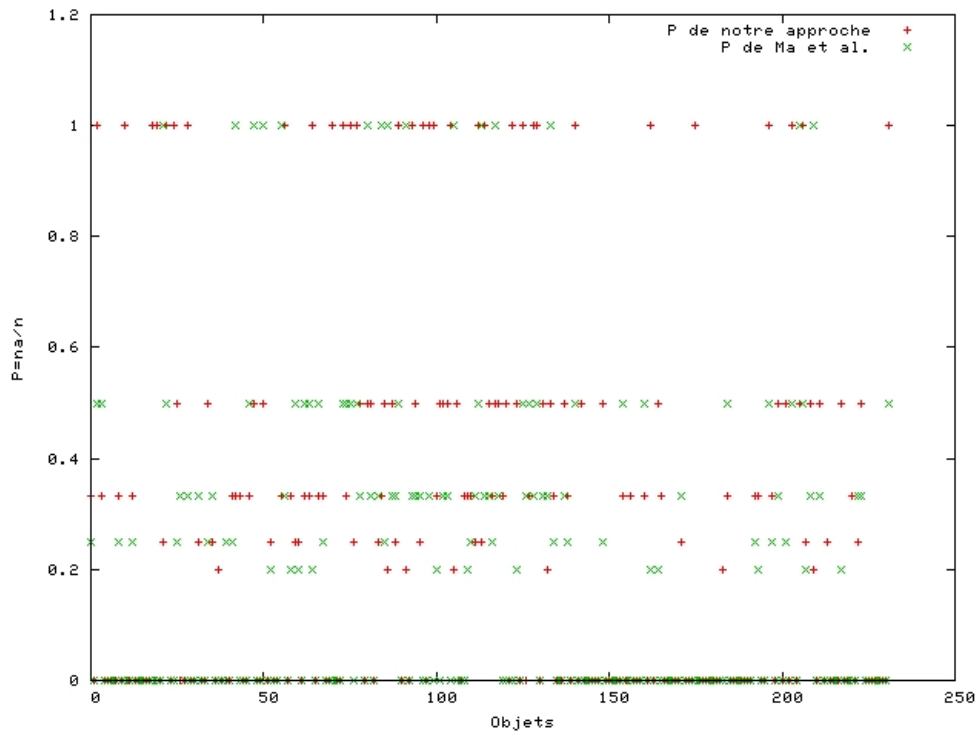


FIG. 6.10 – Valeurs de la mesure P obtenues par les deux méthodes dans la deuxième expérimentation. Cette mesure montre la capacité à corriger les erreurs produites par la détection et le suivi d’objets. Une méthode est efficace si elle obtient une grande valeur de P . La valeur de P peut être de 0% à 100% (respectivement de 0 à 1 dans cette figure). Parmi 661 objets, notre méthode obtient la valeur de P plus élevée que celle de la méthode de Ma et al. sur 142 objets alors que la méthode de Ma et al. a de meilleurs résultats sur 75 objets.

toutes les deux expérimentations sont montrées dans le tableau 6.9. Notre méthode a amélioré celle de Ma et al. Elle obtient en même temps de petites valeurs de F et de grandes valeurs de P, c'est-à-dire elle fournit une représentation compacte et efficace des objets. Cependant, la valeur de la mesure P est encore petite (dans l'expérimentation 2, 70% blobs représentatifs choisis sont pertinents). Cela va influencer la qualité de la recherche. La valeur de P peut être augmentée en améliorant la classification des blobs en blobs avec objets et ceux sans objet. D'autres descripteurs d'apparence peuvent être utilisés pour la classification. Nous analysons cette perspective dans le chapitre 7.

TAB. 6.9 – Valeurs moyennes des mesures F et P obtenues par les deux méthodes dans les deux expérimentations.

	$F = \frac{n*100\%}{N}$		$P = \frac{na*100\%}{n}$	
	notre méthode	Ma et al.	notre méthode	Ma et al.
expérimentation 1	7	9.2	95.7	83.9
expérimentation 2	11.1	13.8	74.8	59.1

La figure 6.11 montre deux résultats de détection des blobs représentatifs pour deux personnes détectées. Notre approche permet d'enlever des blobs non pertinents.

6.7.3 Discussions

Les résultats obtenus montrent les points forts des deux méthodes proposées : (1) elles réduisent énormément les informations à stocker (cf. tableau 6.8) ; (2) elles préparent des données pertinentes pour la tâche de mise en correspondance entre les objets dans la phase de recherche. Les caractéristiques de chacune des méthodes sont analysées. Grâce à cette analyse, le choix de la méthode utilisée peut être donné. La comparaison de la méthode basée sur le regroupement des blobs avec celle de Ma et al. montre que notre méthode est supérieure que celle de Ma et al. Dans les sections suivantes, les résultats de la recherche d'objets sont les résultats de mise en correspondance entre les objets par leurs blobs représentatifs.

6.8 Évaluation de la recherche d'objets

La recherche d'objets consiste à trouver des objets indexés dans la base de données. La requête peut être une imagerie ou un objet. Il est à noter que les objets sont les objets mobiles. La recherche d'objets pour la vidéosurveillance est différente que celle pour les images. Un objet dans une image contient en général une seule région. Dans cette section, nous analysons tout d'abord les difficultés rencontrées pour la recherche des objets (section 6.8.1). Deux comparaisons de notre approche avec deux approches dans l'état de l'art : approche de Ma et al. [Ma 2007] (section 6.8.2) et celle de Calderara et al. [Calderara 2006] (section 6.8.3) sont ensuite



(a) Cinq blobs représentatifs de la personne #1055



(b) Cinq blobs représentatifs de la personne #4507

FIG. 6.11 – Deux personnes avec leurs blobs représentatifs détectés par la méthode de Ma et al. . Les blobs en rouge qui ne sont pas appropriés sont enlevés par notre méthode.

présentées. Puis, nous présentons une évaluation de performance des descripteurs visuels en indexation et recherche de vidéos de vidéosurveillance (section 6.8.4). La recherche d'objets en se basant sur leurs trajectoires est montrée dans la section 6.8.5. Enfin, nous présentons des conclusions (section 6.8.6).

6.8.1 Analyse de difficultés rencontrées dans la recherche d'objets

La recherche d'objets dépend fortement de la qualité de la détection et du suivi d'objets. Nous analysons leurs influences pour l'indexation et la recherche de vidéos de vidéosurveillance selon trois mesures : le rapport du nombre de blobs bien déterminés et le nombre total de blobs, la confusion d'étiquette et la persistance d'étiquette.

Le blob des objets n'est pas toujours bien déterminé. Il existe en général quatre cas : (1) l'objet n'est pas présent dans le blob, (2) l'objet est partiellement présent dans le blob (3) plusieurs objets sont présents dans le blob, (4) la combinaison de (2) et (3). La figure 6.12 illustre des exemples dans lesquels les blobs ne sont pas bien déterminés. L'utilisation de ces blobs pour la mise en correspondance entre des objets peut produire des erreurs. Afin de remédier à cette difficulté, dans les chapitres 4 et 5, nous avons proposé deux méthodes de détection des blobs représentatifs des objets et une méthode de mise en correspondance entre objets par la distance EMD. Grâce aux méthodes de détection des blobs représentatifs, les blobs appartenant au premier cas sont enlevés avant de les mettre en correspondance. La mise en correspondance nous permet de travailler avec les trois derniers cas.

La confusion d'étiquette mesure le nombre d'objets réels qui sont détectés et suivis comme un seul objet. Cela pose problème lors de la mise en correspondance entre objets. La figure 6.13.a montre un exemple de cette difficulté. La distance EMD utilisée pour la mise en correspondance entre des objets permet d'apparier partiellement des objets.

La persistance d'étiquette mesure le nombre d'objets détectés et suivis pour un seul objet réel. Notre approche arrive à remédier à cette difficulté parce qu'au lieu d'apparier les objets par leurs étiquettes, elle compare les objets par leurs apparences. La figure 6.13.b donne un exemple de cette difficulté.

6.8.2 Comparaison de notre méthode de mise en correspondance avec celle de Ma et al.

Nous comparons notre méthode de mise en correspondance basée sur EMD et la mise en correspondance de Ma et al. [Ma 2007], [Cohen 2008] qui se base sur la distance de Hausdorff. Nous appliquons les deux méthodes sur les mêmes vidéos, les mêmes ensembles de requêtes et les mêmes descripteurs.

Dans la première expérimentation, nous travaillons avec une vidéo enregistrée par une caméra dans une scène. Les requêtes sont des personnes détectées. Cela correspond au scénario de recherche où le personnel de sécurité a une personne, il veut savoir si cette personne apparaît dans la scène à différents moments. Nous



FIG. 6.12 – (a) personne détectée n'est pas présente dans le blob ; (b) personne détectée est partiellement présente dans le blob ; (c) et (d) deux personnes sont présentes dans un seul blob.



FIG. 6.13 – (a) un exemple de confusion d'étiquette : trois personnes réelles sont détectées et suivies comme une seule personne ; (b) un exemple de persistance d'étiquette : une personne est détectée et suivie comme deux personnes différentes.

choisissons 247 personnes de la vidéo CARE_6 comme personnes recherchées dans la première expérimentation. Les requêtes sont mises en correspondance avec 810 personnes indexées provenant de la vidéo CARE_6. La figure 6.14 montre le résultat obtenu de la première expérimentation. Parmi 247 personnes recherchées, notre méthode est plus performante que celle de Ma et al. sur 187 requêtes.

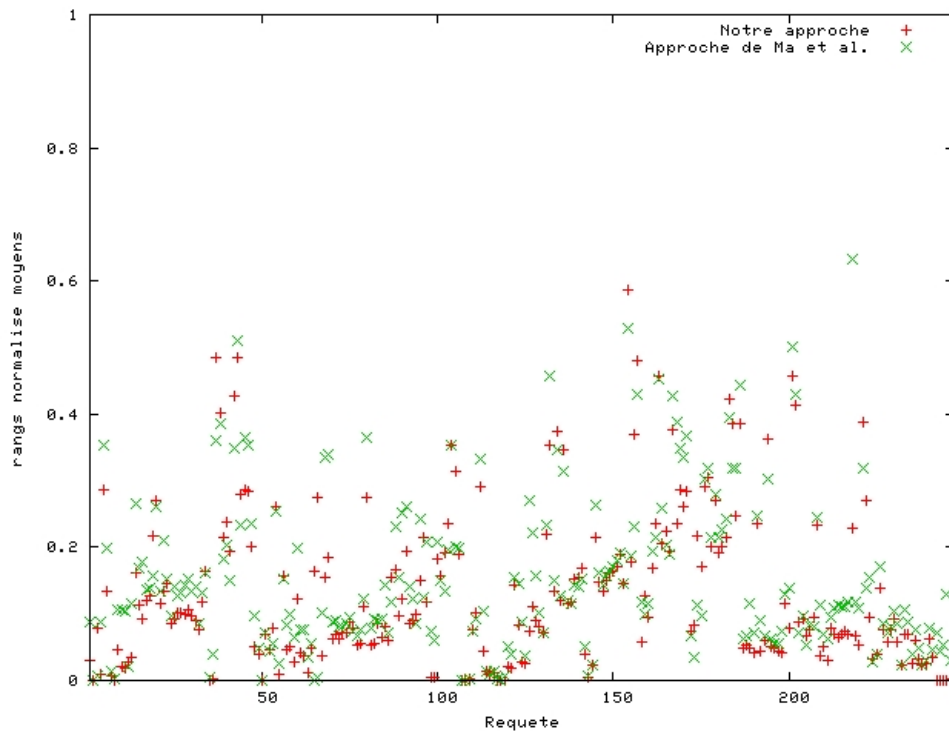


FIG. 6.14 – Rangs normalisés moyens obtenus par les deux méthodes sur 247 personnes recherchées. Plus les valeurs de rang normalisé moyen sont petites, plus la méthode est efficace. Notre méthode est plus performante que celle de Ma et al. sur 187 requêtes sur les 247.

La deuxième expérimentation est réalisée sur deux vidéos enregistrées par deux caméras observant la même scène. Les requêtes sont également des personnes détectées. Cela décrit le scénario de recherche où le personnel de sécurité a une personne observée par une caméra, il veut savoir si cette personne est aussi observée par d'autres caméras. Les 54 personnes provenant de la vidéo CARE_5 sont choisies comme personnes recherchées. Ces personnes sont mises en correspondance avec 810 personnes provenant de la vidéo CARE_6. Les rangs normalisés moyens obtenus par les deux méthodes sur 54 requêtes sont montrés dans la figure 6.15. Notre méthode obtient de meilleurs résultats avec 32 requêtes.

Les moyennes des rangs moyens obtenus sur 247 et 54 personnes recherchées dans deux expérimentations sont montrées dans le tableau 6.10. Notre méthode de mise en correspondance prouve son efficacité pour la recherche d'objets de vidéos. La qualité de la recherche est très bonne pour l'expérimentation 1 et acceptable

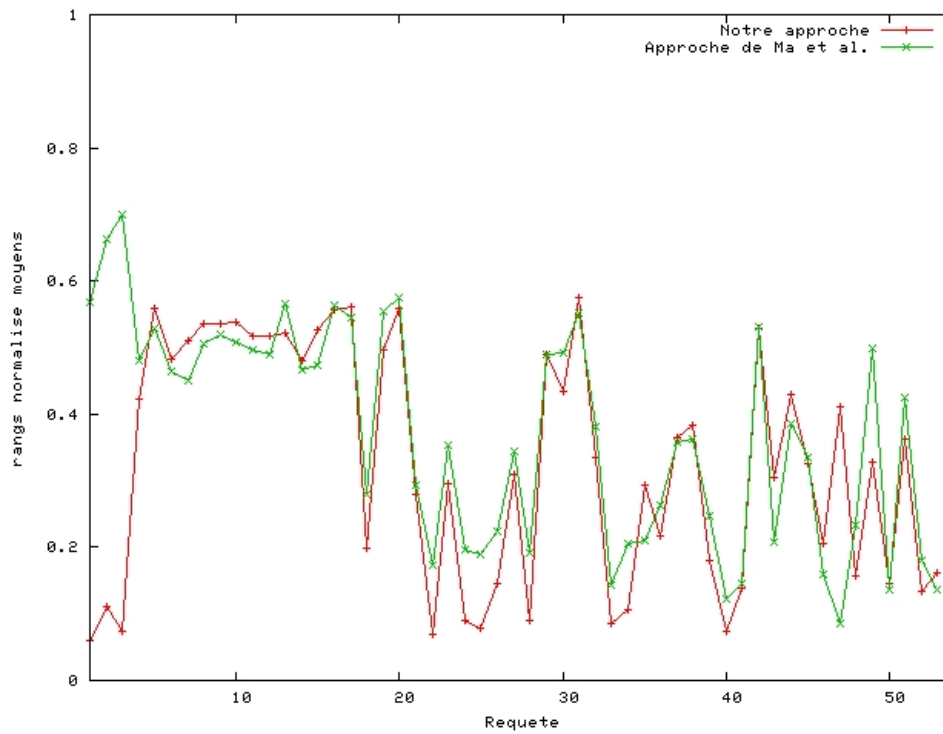


FIG. 6.15 – Rangs normalisés moyens obtenus par les deux méthodes sur 54 requêtes. Plus les valeurs de rang normalisé moyen sont petites, plus la méthode est efficace. Notre méthode obtient de meilleurs résultats sur 32 requêtes.

TAB. 6.10 – Moyennes des rangs moyens obtenus par les deux méthodes de deux expérimentations. Les rangs moyens obtenus par notre méthode sont plus petits que ceux de la méthode de Ma et al.

	notre méthode	Ma et al.
expérimentation 1	0.130	0.149
expérimentation 2	0.321	0.373

pour l'expérimentation 2. Parmi les 247 requêtes de l'expérimentation 1, les 246 (respectivement 225) requêtes obtiennent le rang qui est inférieur à 0.5 (respectivement 0.3). Pour la deuxième expérimentation, les 40 (respectivement 24) parmi 54 requêtes obtiennent le rang qui est inférieur à 0.5 (respectivement 0.3). Il est à rappeler que plus la valeur de rang est petite, plus l'approche est efficace.

La performance des deux méthodes pour les premiers résultats est analysée dans le tableau 6.11 en utilisant les mesures TP (true positive) et FP (false positive). Il est à noter que la valeur de TP mesure le nombre de résultats pertinents alors que la valeur de FP montre le nombre de résultats non pertinents dans les m premiers résultats. La valeur de m est de 1 à 4. Nous calculons les valeurs de TP et FP sur toutes les requêtes. N_m , G_m , TP_m , FP_m sont le nombre total des premiers résultats, le nombre total des résultats pertinents dans la vérité terrain, le nombre total des résultats pertinents et celui des résultats non pertinents dans les m premiers résultats de toutes les requêtes. Le nombre de requêtes est 247 et 54 pour les expérimentations 1 et 2 respectivement. L'expérimentation 1 correspond à la recherche de la même personne à différents moments tandis que l'expérimentation 2 cherche la même personne observée par différentes caméras. Les valeurs de TP obtenues avec notre méthode sont plus légèrement élevées que Ma et al. dans tous les deux expérimentations. La qualité de la recherche pour les premiers résultats obtenue dans l'expérimentation 1 est très bonne. Cependant, celle de l'expérimentation 2 n'est pas encore satisfaisante (les 6 sur les 54 requêtes obtiennent le résultat pertinent dans le premier résultat).

Nous analysons un exemple avec lequel notre méthode montre qu'elle est plus pertinente que celle de Ma et al. pour l'imperfection de la détection et du suivi d'objets. Une personne recherchée est la personne #2160 dont la confusion d'étiquette est 2 (c'est-à-dire deux personnes réelles sont suivies comme une seule personne). Quatre blobs représentatifs avec leurs poids sont montrés dans la figure 6.16. Le poids des blobs de la personne #2160 indique que la personne réelle est la personne qui est présente dans les deux premiers blobs. Les deux derniers blobs sont créés par erreurs de la détection et du suivi d'objets. Nous étudions la recherche avec les deux méthodes des deux personnes de la base de données #1518 et #1763 (cf. figures 6.17, 6.18). Si la distance calculée entre les deux personnes #2160 et #1518 est plus petite que celle entre les deux personnes #2160 et #1763, cela montre que la méthode arrive à travailler avec les vidéos ayant les valeurs de confusion élevées. En appelant notre méthode de mise en correspondance, la distance calculée entre les deux personnes #2160 et #1518 est 3.21 tandis que celle de les deux personnes #2160 et #1763 est 3.79. La méthode de Ma et al. n'est pas appropriée, car elle obtient 4.81 et 3.31 respectivement pour les deux distances. Nous les analysons en détail dans la figure 6.19. Notre mise en correspondance est basée sur le poids du blob. Elle permet donc de diminuer le rôle de blobs non pertinents car leurs poids sont petits.

En résumé, notre méthode est plus efficace que celle de Ma et al. pour l'analyse globale (les rangs normalisés moyens) et aussi pour l'analyse locale sur les premiers résultats (les mesures TP et FP). Les deux méthodes sont capables de retrouver

TAB. 6.11 – Comparaison de deux méthodes (notre méthode et celle de Ma et al.) basée sur les mesures TP (true positive) et FP (false positive). La valeur de TP mesure le nombre de résultats pertinents alors que la valeur de FP montre le nombre de résultats non pertinents dans les m premiers résultats. m a quatre valeurs (de 1 à 4). Le nombre de requêtes est 247 et 54 respectivement. Nous calculons les valeurs de TP et FP sur toutes les requêtes. N_m , G_m , TP_m , FP_m sont le nombre total des premiers résultats, le nombre total des résultats pertinents dans la vérité terrain, le nombre total des résultats pertinents et celui des résultats non pertinents dans les m premiers résultats de toutes les requêtes.

Méthode	m=1			m=2			m=3			m=4		
	G_1	TP_1	FP_1	G_2	TP_2	FP_2	G_3	TP_3	FP_3	G_4	TP_4	FP_4
Expérimentation 1												
	$N_m = 247$			$N_m = 494$			$N_m = 741$			$N_m = 988$		
Notre méthode	247	247	0	491	366	128	727	466	275	953	553	435
Ma et al.	247	247	0	491	355	139	727	440	301	953	516	472
Expérimentation 2												
	$N_m = 54$			$N_m = 108$			$N_m = 162$			$N_m = 216$		
Notre méthode	54	6	48	104	9	99	158	10	152	212	14	202
Ma et al.	54	1	53	104	3	105	158	5	157	212	8	208

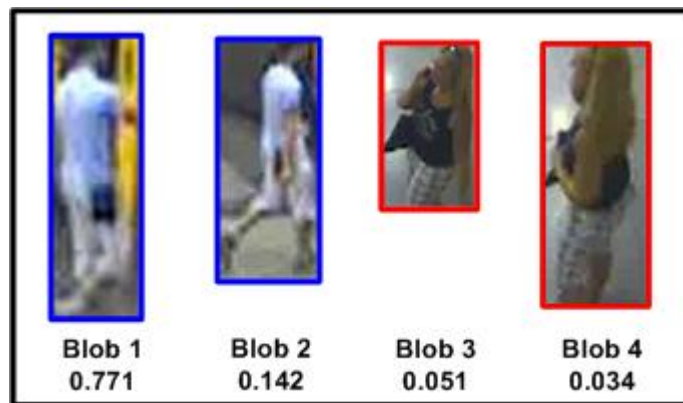


FIG. 6.16 – 4 blobs représentatifs et leurs poids pour la personne #2160



FIG. 6.17 – 4 blobs représentatifs et leurs poids pour la personne #1518



FIG. 6.18 – 5 blobs représentatifs et leurs poids pour la personne #1763

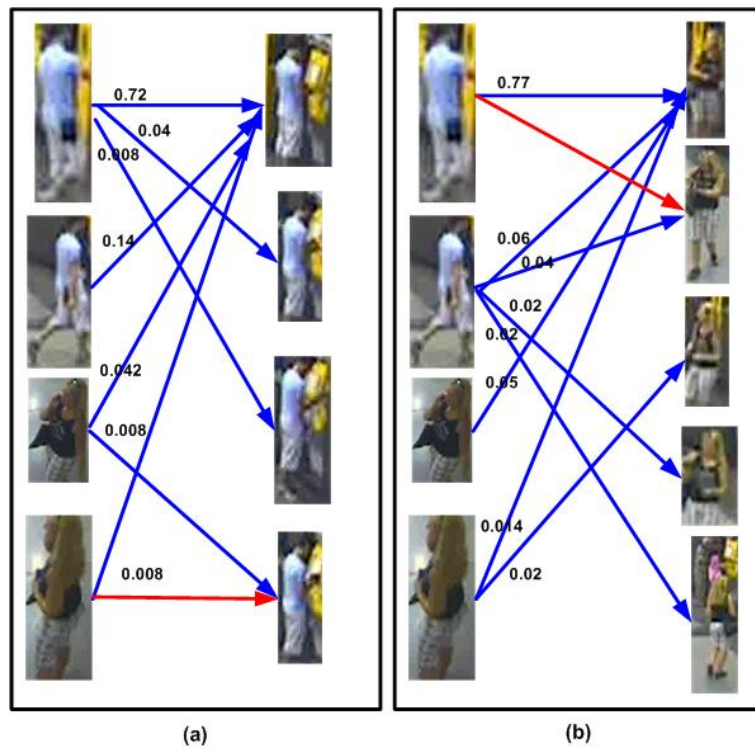


FIG. 6.19 – Mise en correspondance : (a) entre les deux personnes #2160 et #1518 ; (b) et entre les deux personnes #2160 et #1763. Les valeurs correspondantes montrent des parties que ce blob participe à la mise en correspondance dans notre méthode (en bleu). La méthode de Ma et al. détermine la distance entre deux ensembles de blobs par la distance de deux blobs (en rouge)

les objets dans les vidéos ayant la valeur de persistance d'étiquette élevée car elles appartiennent aux objets par la similarité et non pas par l'étiquette. Notre méthode est plus performante que celle de Ma et al. pour les vidéos ayant la valeur de confusion d'étiquette élevée.

Concernant le temps de calcul, notre méthode prend en plus du temps de celle de Ma et al. Afin de mettre en correspondance deux objets ayant N_1 et N_2 blobs respectivement, les deux méthodes doivent calculer $N_1 * N_2$ comparaisons des blobs. En plus, notre méthode doit trouver le flux F^* optimal pour le problème défini dans l'équation 5.12 du chapitre 5. Le temps pour cela est en $O(n^3 \log n)$ selon Rubner et al. [Rubner 2000]. Cependant, nous appelons la distance EMD à la mise en correspondance entre objets basée sur leurs blobs représentatifs. Le nombre de blobs représentatifs d'un objet est petit.

6.8.3 Comparaison de notre méthode avec celle de Calderara et al.

La méthode de Calderara et al. [Calderara 2006] tourne sur les MCATs c'est-à-dire l'ensemble des blobs d'une personne observée par plusieurs caméras. Les auteurs ont appelé SCAT l'ensemble de blobs de la personne correspondant à une caméra. Un SCAT est équivalent au concept objets dans notre modèle de données. La méthode de Calderara et al. consiste à mettre en correspondance la requête qui est une image d'exemple et les MCATs de la base de données. Nous comparons notre méthode avec celle de Calderara et al. au niveau SCAT. Nous avons implémenté l'algorithme de mise à jour des poids des gaussiennes présentées par Stauffer et al. [Stauffer 1999]. Les paramètres de l'algorithme de Calderara sont : $\sigma = 0.1$, $\alpha = 0.01$, le poids initialisé est 0.1. Dans la phase d'indexation, pour chacun des objets, dix gaussiennes sont créées et mises à jour en utilisant tous les blobs de l'objet.

Nous avons effectué trois expérimentations. Pour les trois expérimentations, dix gaussiennes sont entraînées de manière présentée dans [Calderara 2006].

La première expérimentation est réalisée sur une vidéo enregistrée par une caméra dans une scène. Les requêtes sont des images d'exemples. Cela correspond au scénario de recherche : le personnel de sécurité a une image d'exemple, il veut savoir si les personnes qui sont semblables à l'image d'exemple apparaissent dans la scène à différents moments. La figure 6.20 illustre les résultats obtenus pour 16 requêtes de la vidéo CARE_6 sur 810 personnes indexées. Les résultats de notre approche sont meilleurs que ceux de Calderara et al. dans la plupart des cas. Cependant, la méthode de Calderara et al. donne de meilleurs résultats avec deux requêtes (requêtes #2 et #8). Les personnes qui sont semblables aux images d'exemple de ces deux requêtes sont bien détectées et suivies sur de nombreux frames (de 40 à 905 frames). C'est pourquoi les gaussiennes de l'algorithme de Calderara et al. sont représentatives pour ces personnes.

La deuxième expérimentation est effectuée comme la première expérimentation mais les requêtes sont des personnes et non des images. La figure 6.21 montre les résultats obtenus par les deux méthodes avec 247 personnes recherchées sur 810 personnes de la vidéo CARE_6. L'approche de Calderara et al. prend le blob dont

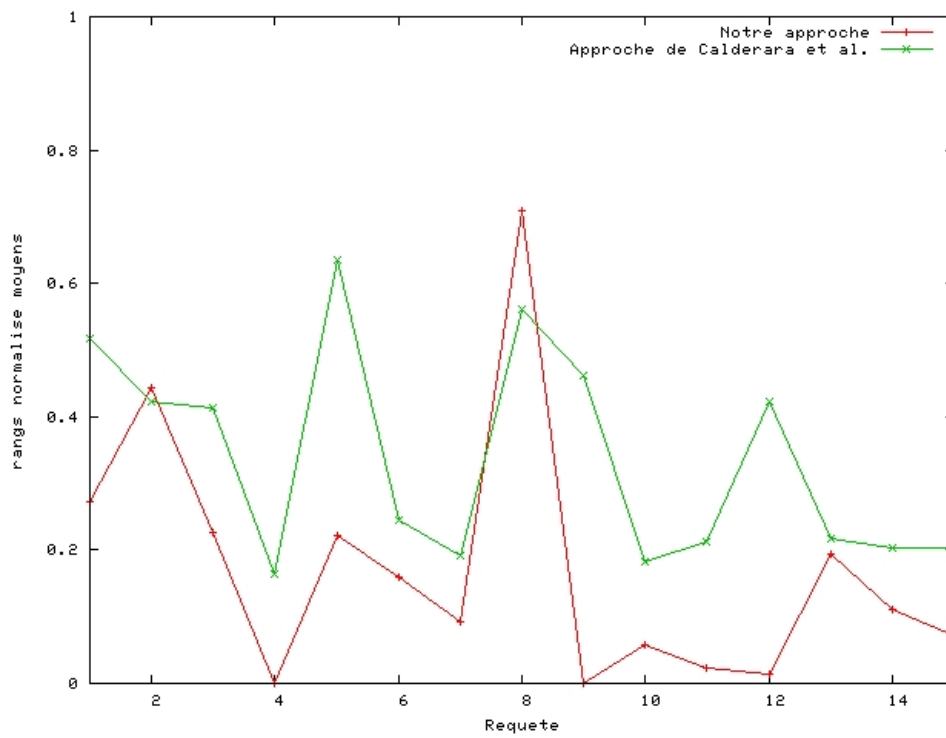


FIG. 6.20 – Rangs normalisés moyens obtenus par notre approche et celle de Calderara et al. avec 16 requêtes sur 810 personnes indexées de la vidéo CARE_6.

la variation de couleur est la plus élevée parmi les blobs de la personne recherchée comme une image d'exemple.

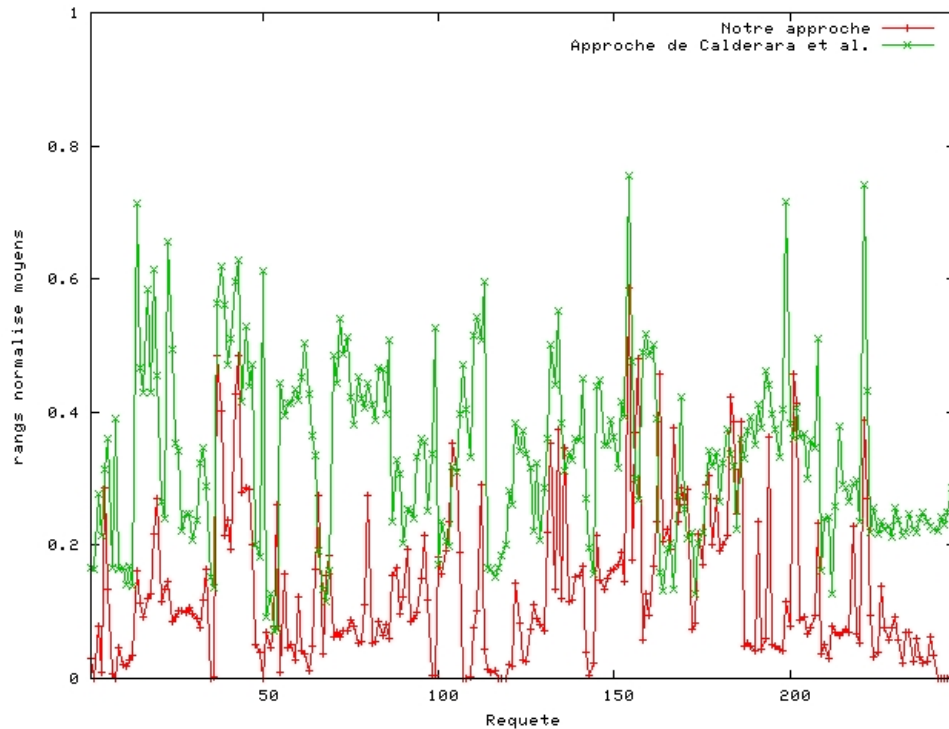


FIG. 6.21 – Rangs normalisés moyens obtenus par notre approche et celle de Calderara et al. dans la deuxième expérimentation avec 247 personnes recherchées sur 810 personnes indexées.

La troisième expérimentation est réalisée sur deux vidéos enregistrées par deux caméras. Les requêtes sont des personnes. Cela correspond au scénario de recherche : le personnel de sécurité a une personne observée par une caméra, il veut savoir si cette personne est observée par d'autres caméras. L'approche de Calderara et al. détermine l'image d'exemple à partir de la personne recherchée comme dans la deuxième expérimentation. La figure 6.22 illustre les résultats obtenus par les deux méthodes avec 54 personnes recherchées de la vidéo CARE_5 sur 810 personnes indexées de la vidéo CARE_6. Parmi 54 requêtes, notre approche obtient de meilleurs résultats que celle de Calderara et al. dans 36 requêtes. L'approche de Calderara et al. a de bons résultats avec 18 requêtes. Cela peut être expliqué : l'approche de Calderara et al. prend une seule image d'exemple de la personne recherchée tandis que notre approche utilise tous les blobs représentatifs de la personne recherchée requête. Dans certains cas, les blobs représentatifs ne sont pas pertinents pour l'objet (voir figure 6.23). La mise en correspondance en utilisant ces blobs n'est donc pas efficace.

Nous analysons les résultats de recherche des requêtes #6 et #40 de la troisième expérimentation. La méthode de Calderara et al. obtient de meilleurs résultats avec la requête #6 mais ce n'est pas le cas pour la requête #40. Pour la requête #6

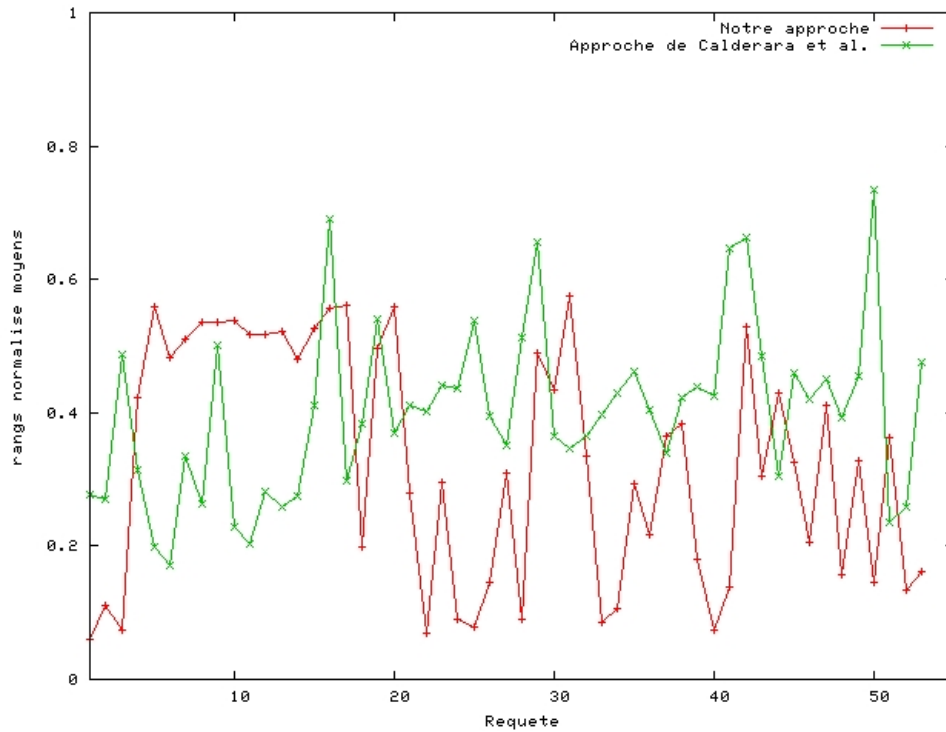


FIG. 6.22 – Rangs normalisés moyens obtenus par la troisième expérimentation avec 54 personnes recherchées de CARE_5 sur 810 personnes indexées de CARE_6

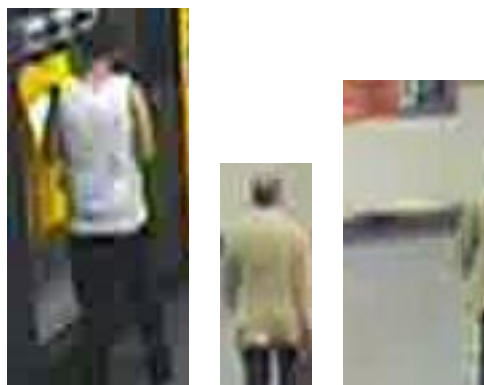


FIG. 6.23 – Trois blobs représentatifs pour une personne recherchée. Les blobs représentatifs ne sont pas toujours pertinents.

(voir figure 6.24), la détection et le suivi des personnes pertinentes sont relativement parfaits. Notre méthode retrouve des personnes non pertinentes en raison de la présence dominante de la machine de vente et d'autres personnes. Cependant, notre approche est appropriée pour la requête #40 (voir figure 6.25). Elle retrouve des résultats pertinents dans les premiers résultats. De plus, pour les résultats non pertinents, les personnes retrouvées sont très semblables à la requête. La méthode de Calderara et al. montre sa limitation. Les personnes pertinentes ne sont pas bien détectées et suivies (elles sont représentées par leurs ombres ou par les murs de la plate-forme). De plus, les gaussiennes se base sur les distributions des couleurs, ils perdent donc les informations spatiales.



FIG. 6.24 – Résultats de la requête #6 de la troisième expérimentation. Les images en haut sont les blobs représentatifs de la personne recherchée et les trois premiers résultats. Les images en bas sont les blobs de la personne recherchée et les premiers résultats obtenus par la méthode de Calderara et al. Les résultats en rouge sont des résultats pertinents.

TAB. 6.12 – Moyennes des rangs moyens obtenus par les deux méthodes (notre méthode et celle de Calderara et al.) dans les trois expérimentations. Plus les valeurs de rang normalisé moyen sont petites, plus la méthode est efficace.

	notre méthode	Calderara et al.
expérimentation 1	0.166	0.343
expérimentation 2	0.130	0.334
expérimentation 3	0.321	0.405

Les moyennes des rangs moyens obtenus par les deux méthodes (notre méthode

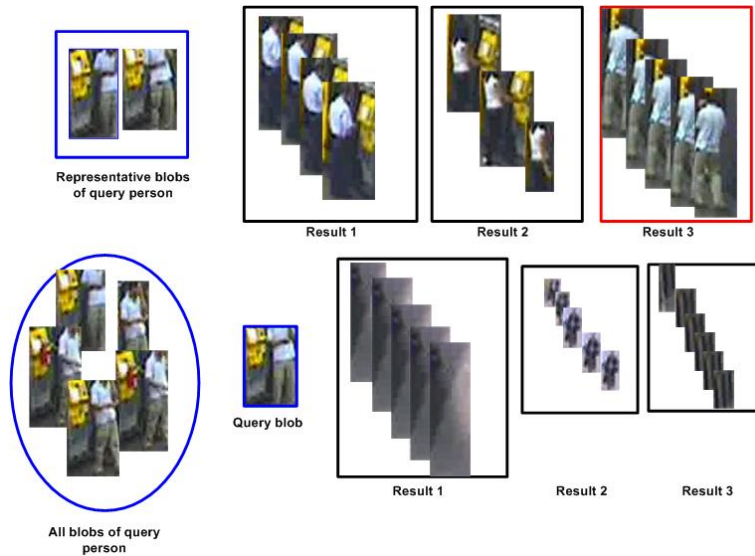


FIG. 6.25 – Résultats de la requête #40 de la troisième expérimentation. Les images en haut sont les blobs représentatifs de la personne recherchée et les trois premiers résultats. Les images en bas sont les blobs de la personne recherchée et les premiers résultats obtenus par la méthode de Calderara et al. Les résultats en rouge sont des résultats pertinents.

et celle de Calderara et al.) dans les trois expérimentations (voir tableau 6.12) montrent l'efficacité de notre approche en indexation et recherche de vidéos de vidéosurveillance. La qualité de la recherche est bonne. Dans la première expérimentation, les 15 (respectivement 14) sur 16 requêtes ont le rang qui est inférieur à 0.5 (respectivement 0.3). Dans la deuxième expérimentation, parmi les 247 requêtes, les 246 (respectivement 225) requêtes obtiennent le rang qui est inférieur à 0.5 (respectivement 0.3). Pour la troisième expérimentation, les 40 (respectivement 24) parmi 54 requêtes obtiennent le rang qui est inférieur à 0.5 (respectivement 0.3).

La performance des deux méthodes pour les premiers résultats est analysée dans le tableau 6.13 en utilisant les mesures TP (true positive) et FP (false positive). La valeur de TP mesure le nombre de résultats pertinents alors que la valeur de FP montre le nombre de résultats non pertinents dans les m premiers résultats. La valeur de m est de 1 à 4. Nous calculons les valeurs de TP et FP sur toutes les requêtes. N_m , G_m , TP_m , FP_m sont le nombre total des premiers résultats, le nombre total des résultats pertinents dans la vérité terrain, le nombre total des résultats pertinents et celui des résultats non pertinents dans les m premiers résultats de toutes les requêtes. Le nombre de requêtes est 16, 247, 54 pour l'expérimentation 1, 2 et 3 respectivement. Les expérimentations 1 et 2 correspondent à la recherche de la même personne à différents moments tandis que l'expérimentation 3 cherche la même personne observée par différentes caméras. Notre méthode obtient de très grande valeur TP pour les deux premières expérimentations. Les résultats pertinents sont retrouvés quasiment dans le premier résultat. Les valeurs de TP de la méthode

de Calderara et al. sont très petites (p. ex. parmi 16 requêtes, cette méthode retrouve des résultats pertinents à la première position de deux requêtes).

Il est intéressant d'analyser l'expérimentation 3. La performance globale (voir tableau 6.12) de notre méthode est meilleure que celle de Calderara et al. et pourtant sa valeur de TP est moins élevée que celle de Calderara et al. Cela montre que notre méthode retrouve des résultats pertinents avec les rangs plus élevés que 4. La méthode de Calderara et al. est plus efficace dans cette expérimentation car elle arrive à éviter l'erreur de la détection des blobs représentatifs pour la personne recherchée. Pour une personne recherchée, cette méthode choisit une seule image ayant la variation de la couleur la plus élevée tandis que notre approche emploie tous les blobs représentatifs de la personne recherchée. Cependant la méthode de Calderara et al. n'est pas appropriée si l'utilisateur veut chercher des personnes avec la personne recherchée ayant des aspects différents.

TAB. 6.13 – Comparaison de deux méthodes (notre méthode et celle de Calderara et al.) basée sur les mesures TP (true positive) et FP (false positive). La valeur de TP mesure le nombre de résultats pertinents alors que la valeur de FP montre le nombre de résultats non pertinents dans les m premiers résultats. m a quatre valeurs (de 1 à 4). Le nombre de requêtes est 16, 247, et 54 respectivement. Nous calculons les valeurs de TP et FP sur toutes les requêtes. N_m , G_m , TP_m , FP_m sont le nombre total des premiers résultats, le nombre total des résultats pertinents dans la vérité terrain, le nombre total des résultats pertinents et celui des résultats non pertinents dans les m premiers résultats de toutes les requêtes.

Méthode	m=1			m=2			m=3			m=4		
	G_1	TP_1	FP_1	G_2	TP_2	FP_2	G_3	TP_3	FP_3	G_4	TP_4	FP_4
Expérimentation 1												
	$N_m = 16$			$N_m = 32$			$N_m = 48$			$N_m = 64$		
Notre méthode	16	11	5	29	16	16	42	21	27	58	25	39
Calderara et al.	16	2	14	29	3	29	42	4	44	58	5	59
Expérimentation 2												
	$N_m = 247$			$N_m = 494$			$N_m = 741$			$N_m = 988$		
Notre méthode	247	247	0	491	366	128	727	466	275	953	553	435
Calderara et al.	247	9	238	491	19	475	727	25	716	953	33	955
Expérimentation 3												
	$N_m = 54$			$N_m = 108$			$N_m = 162$			$N_m = 216$		
Notre méthode	54	6	48	104	9	99	158	10	152	212	14	202
Calderara et al.	54	21	33	104	21	87	158	24	138	212	28	188

6.8.4 Évaluation des descripteurs visuels pour la recherche d'objets

Dans cette thèse, nous utilisons les descripteurs visuels suivants : les couleurs dominantes, les histogrammes de contours, les matrices de covariance, les points d'intérêt de Harris, MSER, et DoG associés au descripteur SIFT. L'objectif de cette expérimentation est d'évaluer la performance de ces descripteurs pour l'indexation et la recherche de vidéos de vidéosurveillance. Nous présentons quatre évaluations. Afin d'éviter l'influence de la qualité du suivi des objets, pour chacun des objets, un seul blob représentatif dont le poids est le plus élevé est employé pendant la mise en correspondance. Nous analysons la performance des descripteurs correspondant à deux cas : la détection d'objets manuelle et la détection automatique. Pour cela, nous utilisons deux vidéos dont une vidéo CARE_4 qui est automatiquement analysée par la plate-forme VSIP et une provenant du projet CAVIAR qui est annotée manuellement. La recherche d'objets avec la vidéo CARE_4 vise à retrouver des objets observés par une seule caméra à différents moments tandis que celle avec la vidéo provenant du projet CAVIAR consiste à retrouver des objets observés par deux caméras et à différents moments.

La première évaluation vise à évaluer quatre types d'histogrammes de contour : local, semi-local, global et composé. Quelques exemples de comparaison de l'histogramme local, semi-local et global sont présentés par Park et al. [Park 2000]. Une évaluation quantitative de ces histogrammes en indexation et recherche d'images est montrée par Won [Won 2004]. L'objectif de notre évaluation est d'analyser la performance de ces histogrammes en indexation et recherche de vidéos pour la vidéosurveillance. Nous effectuons la recherche d'objets avec 23 requêtes sur 2311 objets de la vidéo CARE_4. La figure 6.26 illustre quatre résultats obtenus avec quatre types d'histogrammes des contours. La performance de l'histogramme composé est stable, il compense l'inconvénient des histogrammes local et global. L'histogramme semi-local obtient de bons résultats dans la plupart de cas.

La performance de l'histogramme global et local varie selon le contenu de la requête. Nous analysons deux cas : un cas où l'histogramme local est meilleur que le global (voir figure 6.27) et un cas où la performance de l'histogramme global dépasse celle de l'histogramme local (voir figure 6.28). La figure 6.27 montre la requête #1 et deux objets retrouvés en utilisant l'histogramme global qui ne sont pas pertinents. Les histogrammes locaux des objets retrouvés (#1477 et #1170) sont différents de celui de la requête. L'histogramme global qui accumule 16 histogrammes locaux perd l'information spatiale. La figure 6.28 montre le résultat de la requête #21 en utilisant l'histogramme local. L'objet #2727 qui n'est pas pertinent mais retrouvé tandis que l'objet #219 pertinent n'est pas retrouvé car l'objet #2727 et l'objet de requête passent le même endroit. L'histogramme local ne réussit pas dans ce cas.

L'histogramme global est robuste au déplacement de l'objet dans le blob. Cependant si une grande partie du fond est présente ou si l'objet est occulté, la performance de l'histogramme global se dégrade. Contrairement à l'histogramme global, l'histogramme local est sensible à la position de l'objet dans le blob. Dans certaines

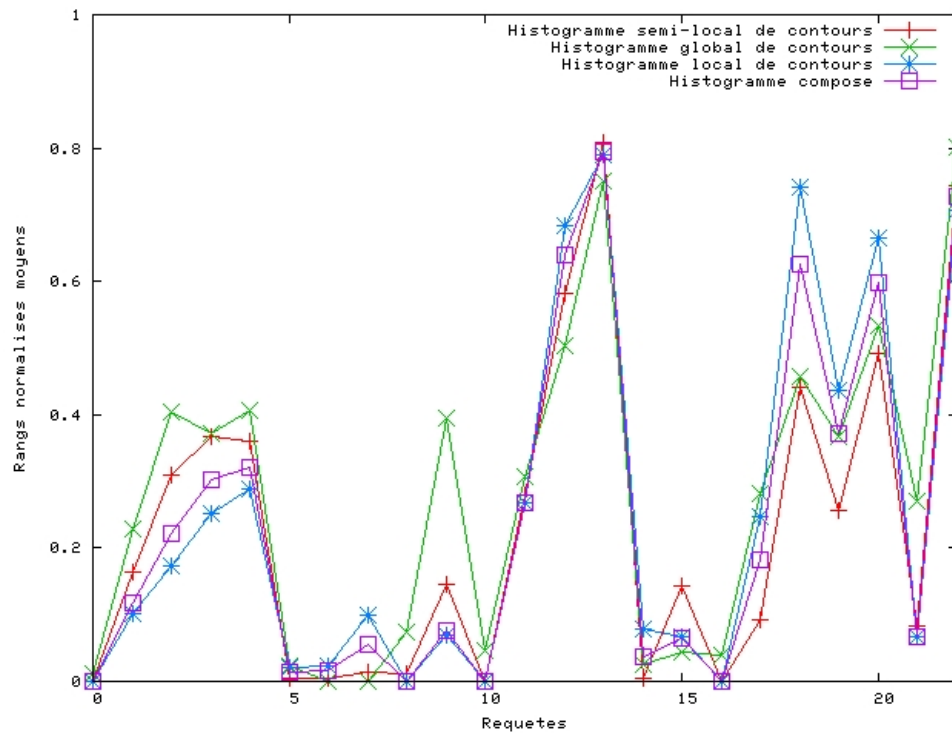


FIG. 6.26 – Rangs normalisés obtenus avec 4 types d’histogrammes des contours sur 2311 objets de la vidéo CARE_4 : l’histogramme local, semi-local, global et composé



FIG. 6.27 – Requête #1 et deux objets retrouvés en utilisant l’histogramme global qui ne sont pas pertinents



FIG. 6.28 – Requête #21 avec un objet qui n’est pas pertinent mais retrouvé (#2727) en utilisant l’histogramme local tandis que l’objet pertinent (#219) n’est pas retrouvé.

applications de vidéosurveillance, par exemple la station de métro, il y a des endroits où beaucoup de personnes y traversent (p. ex. portes, machines à vente), la présence des objets de contexte influence fortement les résultats de recherche d’objets en utilisant l’histogramme local.

En résumé, nous suggérons d’utiliser l’histogramme semi-local et celui composé pour caractériser le contour des objets. La performance de ces deux histogrammes est comparable. Cependant, la taille de l’histogramme composé est deux fois plus grande que celle de l’histogramme semi-local. Les histogrammes des contours n’utilisent pas d’information de couleur. Une combinaison d’histogramme des contours et de descripteurs de couleur tels que les couleurs dominantes peut améliorer la performance de la recherche.

La deuxième évaluation compare la performance des couleurs dominantes et des matrices de covariance. Cette évaluation est intéressante car elle complète l’évaluation des descripteurs de couleurs proposés dans MPEG-7 en indexation et recherche de vidéos pour la vidéosurveillance qui a été présentée par Annesley et al. [Annesley 2005]. Il est à noter que les couleurs dominantes ne prennent pas en compte l’information spatiale tandis que les matrices de covariance la prennent.

Les résultats obtenus avec 23 requêtes sur 2311 objets de la vidéo CARE_4 (voir figure 6.29) montre que les matrices de covariance sont meilleures que les couleurs dominantes dans la plupart de cas. Dans certains cas (p. ex. requête #4) où les matrices de covariance sont sensibles aux pixels différents entre deux blobs, les couleurs dominantes obtiennent de bons résultats.

Comme expliqué dans le chapitre 4, l’extraction des couleurs dominantes demande de déterminer 3 paramètres : le nombre de couleurs dominantes (N_{max}), deux seuils, l’un pour la distorsion (θ) et l’autre pour le regroupement des groupes proches (T_d). Une des raisons qui explique la faible performance des couleurs dominantes est que les paramètres ne sont pas bien définis. La figure 6.30 montre les couleurs dominantes déterminées pour un blob avec des valeurs différentes de T_d . Si la valeur de T_d est élevée, les pixels ayant la couleur assez différente ont la

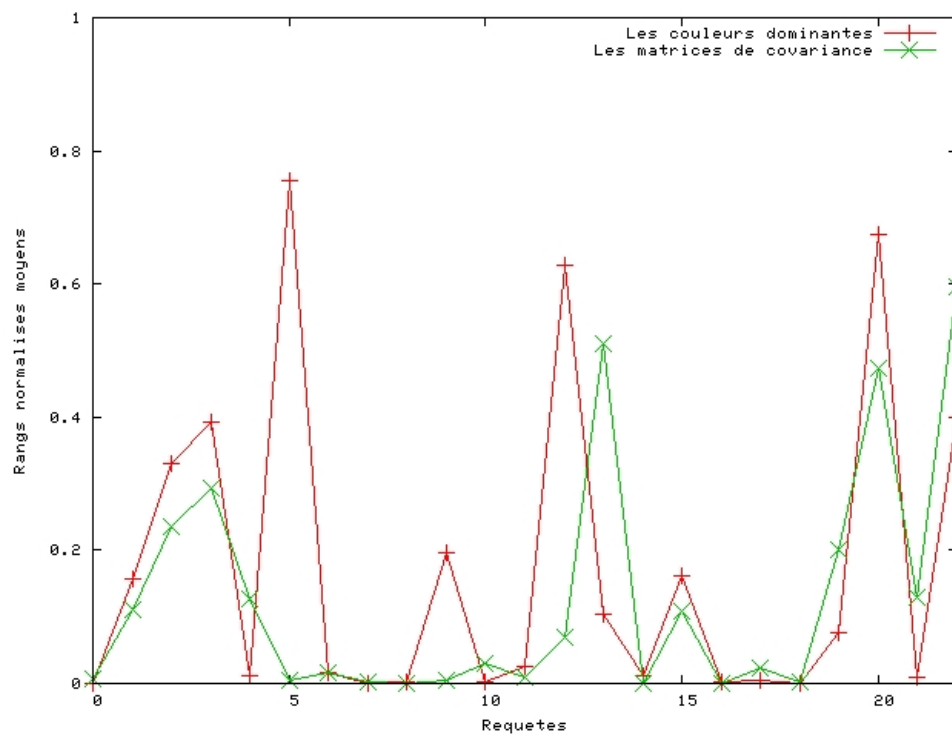


FIG. 6.29 – Rangs normalisés obtenus avec les couleurs dominantes et les matrices de covariance

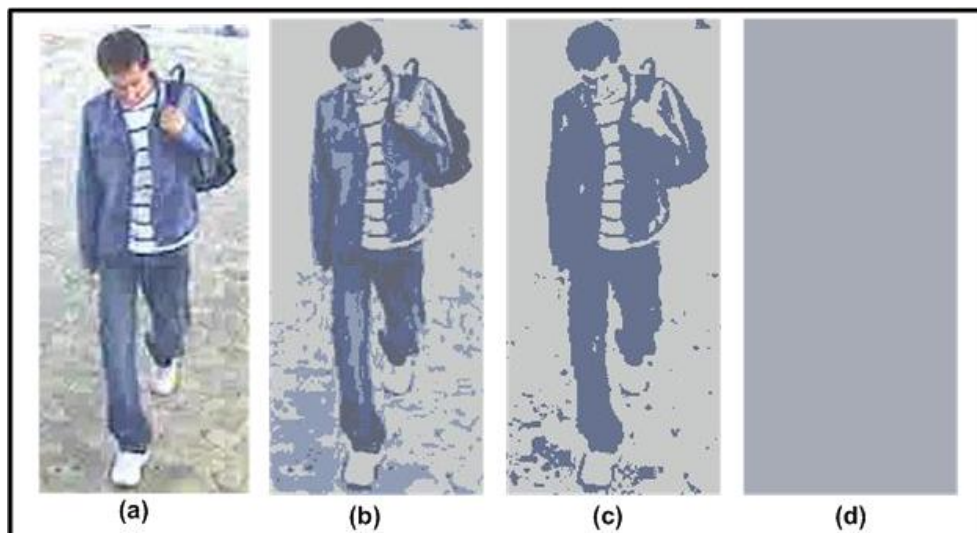


FIG. 6.30 – Les couleurs dominantes déterminées pour un blob avec les valeurs 10, 15, 25 de T_d respectivement.

même couleur dominante. Les valeurs de ces paramètres doivent être choisies selon le contenu visuel du blob. Une phase d'estimation qui consiste à prédéterminer les paramètres de l'extraction de couleurs dominantes d'un blob en fonction de son contenu visuel est nécessaire. Pour l'extraction de matrices de covariance, nous ne devons pas déterminer de paramètres.

Concernant le temps de calcul, l'extraction des couleurs dominantes est plus rapide que celle des matrices de covariance. Selon [Porikli 2006a], le temps d'extraction des matrices de covariance pour une image est $O(d^2 * W * H)$ où d^2 est la taille de la matrice de covariance ($d = 11$ dans cette évaluation), $W * H$ est la taille d'image. Le temps d'extraction des couleurs dominantes est $O(N_{max} * W * H)$ où N_{max} est le nombre maximal de couleurs dominantes ($N_{max} = 8$ dans cette évaluation). De plus, les couleurs dominantes demandent moins d'information stockée (N_{max} vecteurs de 4 éléments dont 3 pour les composantes de couleur et 1 pour son poids) que les matrices de covariance (une matrice de d^2 éléments). Cependant la taille des matrices de covariance des images ayant des tailles différentes est constante. Cette propriété est requise pour les techniques d'apprentissage automatiques.

La troisième évaluation permet de comparer la performance des points d'intérêt associés au descripteurs SIFT en indexation et recherche de vidéos pour la vidéosurveillance. Les points d'intérêt ont prouvé leurs performances en indexation et recherche d'images (voir évaluation montrée par Mikolajczyk et al. [Mikolajczyk 2004]). Il est intéressant d'analyser leurs performances en indexation et recherche de vidéos pour la vidéosurveillance. La figure 6.31 présente les rangs normalisés obtenus de 7 requêtes sur 53 objets provenant du projet CAVIAR pour trois types de points d'intérêt : DoG, MSER, HarrisAffine associés au descripteur SIFT. Les personnes de la base de données sont observées par deux caméras (l'une est en face du magasin et l'autre dans le couloir) et elles se sont déplacées dans la scène. Leurs apparences sont différentes d'une caméra à autre caméra et d'un moment à autre moment. Les points d'intérêt montrent qu'ils sont capables de retrouver ces personnes ayant différents changements si les points d'intérêt sont détectés.

Les points d'intérêts détectés pour un blob sont peu nombreux en raison de deux facteurs : (1) la taille du blob est habituellement petite, (2) les objets en vidéosurveillance peuvent être observés de loin, de plus la résolution de l'image est faible. Le nombre des points d'intérêt détectés dépend fortement des valeurs déterminées pour les paramètres. La recherche d'objets qui n'utilise que les points d'intérêts n'est pas efficace si aucun ou très peu de points sont détectés. Cependant, afin de profiter du point fort des points d'intérêts, la recherche d'objets en combinant des points d'intérêt et un descripteur global tel que les matrices de covariance fait partie de notre travail futur.

La quatrième évaluation donne une comparaison de tous les descripteurs. Les descripteurs utilisés pour la mise en correspondance entre objets sont les couleurs dominantes, les matrices des covariance, les histogrammes des contours et les points d'intérêt. Les figures 6.32 et 6.33 montrent les performances des descripteurs respectivement pour 23 requêtes sur 2311 objets de la vidéo CARE_4 et 15 requêtes sur 53 objets provenant du projet CAVIAR.

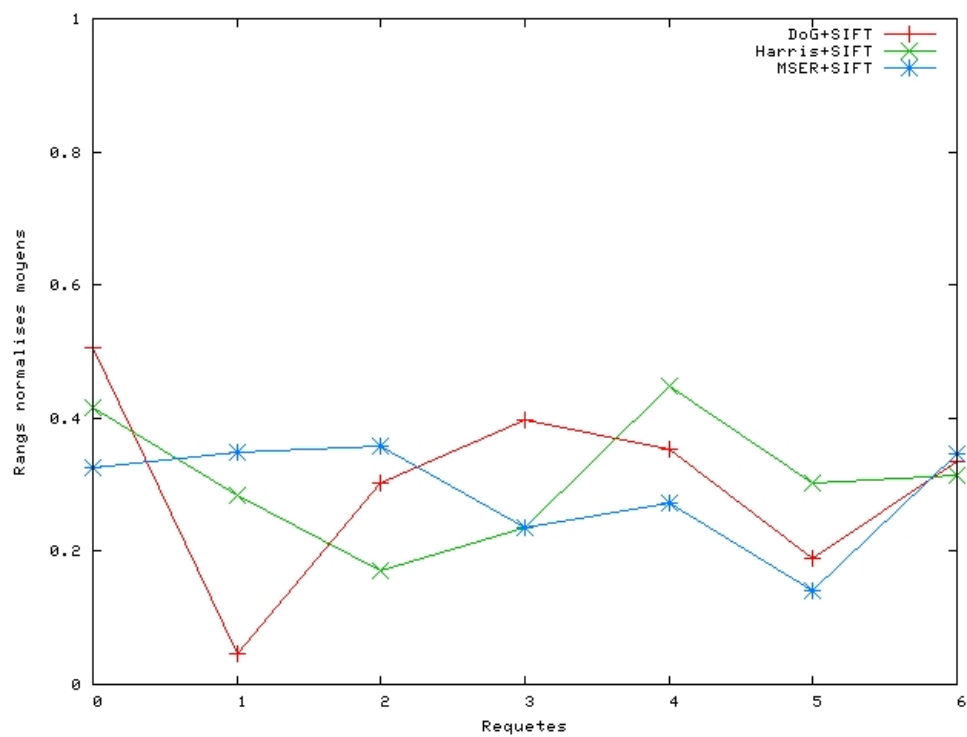


FIG. 6.31 – Rangs normalisés obtenus de 7 requêtes sur 53 objets provenant du projet CAVIAR en utilisant les points d'intérêt (DoG, MSER, HarrisAffine) associés au descripteur SIFT

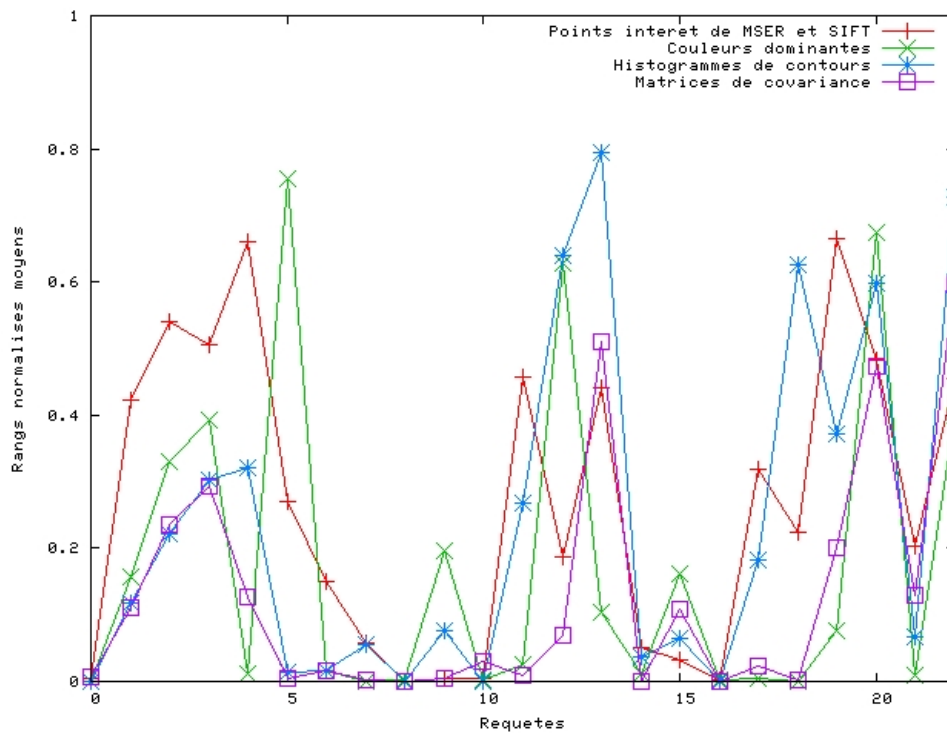


FIG. 6.32 – Résultats de recherche avec 23 requêtes sur 2311 objets de la vidéo CARE_4 avec les points d'intérêt de MSER et SIFT, les histogrammes des contours, les couleurs dominantes et les matrices de covariance.

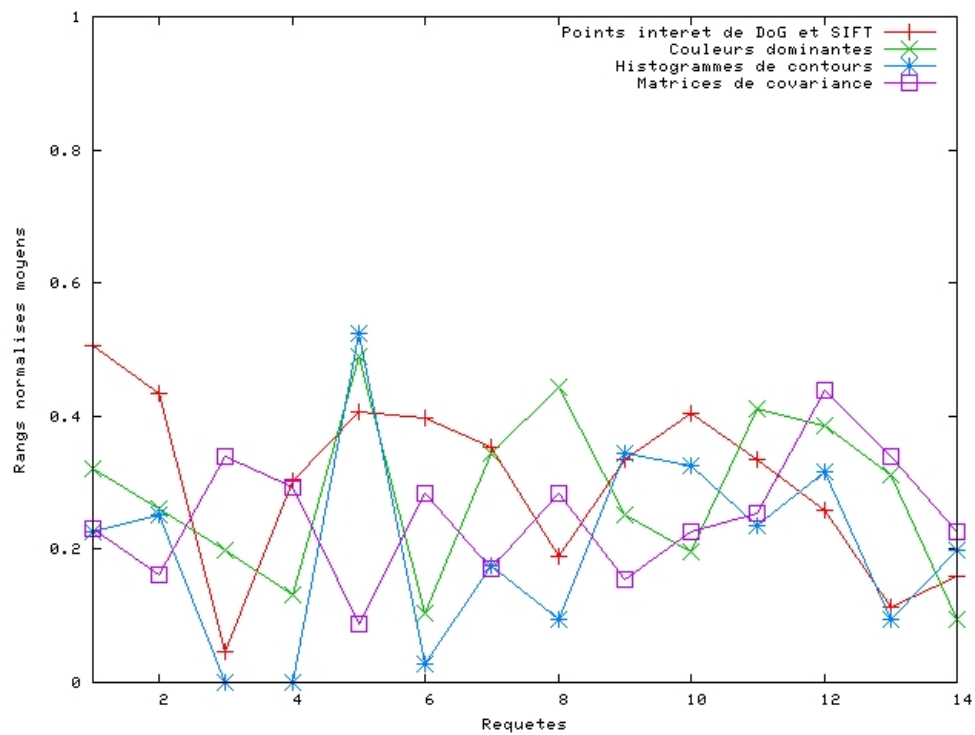
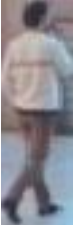






FIG. 6.33 – Résultats de recherche avec 15 requêtes sur 53 objets provenant du projet CAVIAR en utilisant les descripteurs : les points d'intérêt de DoG et SIFT, les histogrammes des contours, les couleurs dominantes et les matrices de covariance

TAB. 6.14 – Comparaison de la performance des 4 descripteurs d'apparence pour les premiers résultats sur les vidéos provenant du projet CAVIAR : les couleurs dominantes (DC), les histogrammes de contours (HC), les matrices de covariance (MC) et les points intérêt DoG+SIFT basée sur les mesures TP (true positive) et FP (false positive). La valeur de TP mesure le nombre de résultats pertinents alors que la valeur de FP montre le nombre de résultats non pertinents dans les m premiers résultats. m a quatre valeurs (de 1 à 4). Le nombre de requête est 15. Nous calculons les valeurs de TP et FP sur toutes les 15 requêtes. N_m , G_m , TP_m , FP_m sont le nombre total des premiers résultats, le nombre total des résultats pertinents dans la vérité terrain, le nombre total des résultats pertinents et celui des résultats non pertinents dans les m premiers résultats de toutes les 15 requêtes.

Descripteurs	m=1, $N_m = 15$			m=2, $N_m = 30$			m=3, $N_m = 45$			m=4, $N_m = 60$		
	G_1	TP_1	FP_1	G_2	TP_2	FP_2	G_3	TP_3	FP_3	G_4	TP_4	FP_4
CD	15	15	0	30	17	13	38	17	28	38	18	42
HC	15	15	0	30	17	13	38	19	26	38	21	39
MC	15	15	0	30	19	11	38	19	26	38	20	40
DoG+SIFT	15	15	0	30	15	15	38	16	29	38	16	44

TAB. 6.15 – Rangs obtenus par les descripteurs pour les personnes observées à différents moments ou par différentes caméras. Les matrices de covariance obtiennent les meilleurs résultats.

Descripteurs	Requête	Objets cibles				
						
Couleurs dominantes		1	36	31	25	49
Histogrammes de contours		1	21	45	35	25
Matrices de covariance		1	7	24	2	10
DoG+SIFT		1	39	31	28	29

Les résultats montrent que si les objets sont bien détectés et le fond et les objets de contexte ne sont présents dans le blob, tous les descripteurs d'apparence utilisés permettent de retrouver des objets avec un résultat acceptable (voir figure 6.33). Pour les autres cas, les matrices de covariance sont plus performantes que les autres descripteurs (voir figure 6.32). Le tableau 6.14 montre la performance des descripteurs pour les premiers résultats avec les vidéos provenant du projet CAVIAR. Les matrices de covariance ont les meilleurs résultats. Un résultat de la recherche est montré dans le tableau 6.15. Les personnes observées à différents moments ou par différentes caméras sont efficacement retrouvées en utilisant les matrices de covariance.

Il est intéressant de voir (cf. figure 6.33) que quand les matrices de covariance utilisent les informations de tous les pixels dans un blob, les points d'intérêt n'emploient que quelques pixels. Les couleurs dominantes et les histogrammes de contours utilisent l'information approximée de la couleur et du contour des pixels. Une de paires de descripteurs (matrices des covariance + couleurs dominantes) ou (matrices des covariance + histogrammes de contours) ou (matrices des covariance + points d'intérêts) peut être choisie comme descripteurs utilisés par défaut si l'utilisateur n'indique pas de descripteur préféré.

6.8.5 Recherche d'objets basée sur les trajectoires

Nous présentons les premiers résultats de l'analyse de trajectoire. Nous utilisons la base de trajectoires présentée dans la section 6.3.1. Cette base comprend 2500 trajectoires divisées en 50 catégories. Chacune des trajectoires est prise comme trajectoire de requête et est comparée avec les trajectoires de la base. Les trajectoires pertinentes sont des trajectoires appartenant à la même catégorie que la trajectoire de requête. Les moyennes des rappel et précision sont calculées.

Les valeurs de d_{min} et d_{max} dans l'équation 4.18 sont $n/20$ et $n/15$ respectivement où n est la longueur de T . Nous prenons $\varepsilon_{dir} = \pi/4$ et $\varepsilon_{dis} = 0.125$ pour la représentation symbolique. La figure 6.34 illustre les résultats de recherche obtenus en utilisant tous les points de trajectoires et les points de contrôles et les distances EDR, EDM pour la représentation numérique et symbolique respectivement. La distance EDR dans les deux cas donne de meilleurs résultats que ceux de EDM. L'utilisation des points de contrôle au lieu de tous les points permet de réduire l'information à stocker.

Le premier résultat est montré. Cependant, la recherche d'objets basée sur leurs trajectoires est difficile pour la vidéosurveillance. Les trajectoires des personnes dans la scène sont souvent très complexes. De plus, les trajectoires créées par la détection et le suivi d'objets sont incomplètes et erronées.

6.8.6 Discussions

Plusieurs évaluations de notre méthode au niveau objet sont données. La comparaison de notre méthode avec deux autres méthodes dans l'état de l'art prouve

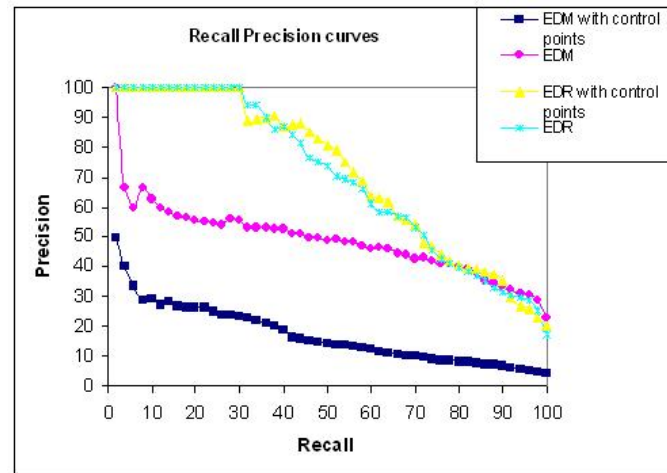


FIG. 6.34 – Courbes de rappel et précision obtenus pour EDR et EDM en utilisant tous les points de trajectoires et les points de contrôle

l'efficacité de notre méthode. Elle est supérieure aux deux autres méthodes pour travailler avec des vidéos qui ne sont pas bien analysées. Une évaluation des descripteurs d'apparence en indexation et recherche de vidéos pour la vidéosurveillance est montrée.

6.9 Évaluation de recherche d'objets et d'événements

La recherche aux niveaux objets et événements consiste à répondre à cinq catégories de requêtes :

- Catégorie 1 : retrouver des événements d'un objet particulier ;
- Catégorie 2 : retrouver des objets impliqués dans un événement particulier ;
- Catégorie 3 : retrouver un événement d'intérêt particulier où l'objet impliqué dans cet événement est semblable à une image d'exemple ;
- Catégorie 4 : retrouver une séquence d'événements d'un objet qui vérifient une relation temporelle ;
- Catégorie 5 : retrouver une séquence d'événements qui vérifient une relation temporelle et l'objet impliqué dans ces événements est semblable à une image d'exemple.

Il est à noter que les requêtes des catégories 1, 2 et 4 sont également abordées par Tian et al. [Tian 2008] et Ghanem et al. [Ghanem 2004]. Les requêtes des catégories 1, 2 et 4 sont résolues par les comparaisons exactes sur les méta-données tandis que celles des catégories 3 et 5 sont résolues en prenant en compte les similarités d'apparence. Les requêtes des catégories 3 et 5 sont des points forts de notre langage de requêtes.

6.9.1 Résultats de recherche avec les requêtes de la catégorie 1

Une requête de la catégorie 1 est exprimée dans le langage :

```
SELECT * FROM CARE_1 WHERE : ENTITIES ((p : PhysicalObjects) (e : Events)) CONDITIONS ((p's Class= "Person")(p involved_in e))
```

Cette requête retrouve tous les événements reconnus pour des objets appartenant à la classe "Person". La requête peut se focaliser sur des événements reconnus d'un objet ayant une étiquette déterminée. La requête suivante permet de retrouver des événements de la personne dont l'étiquette est 57.

```
SELECT * FROM CARE_1 WHERE : ENTITIES ((p : PhysicalObjects) (e : Events)) CONDITIONS ((p's Class= "Person")(p involved_in e)(p's Id=57))
```

Les résultats de cette requête comprennent 144 instances d'événements reconnues de la personne 57 dans la vidéo CARE_1. La qualité des résultats des requêtes dans cette catégorie dépend absolument de la qualité des modules d'analyse vidéo.

6.9.2 Résultats de recherche avec les requêtes de la catégorie 2

Les requêtes de la catégorie 2 consistent à retrouver des objets impliqués dans un événement particulier. Les deux requêtes suivantes appartiennent à la catégorie 2. La première requête cherche les objets impliqués dans l'événement "close_to_VendingMachine1". La deuxième requête vérifie si l'objet 57 est impliqué dans l'événement

"close_to_VendingMachine1". Les deux requêtes sont exprimées par :

```
SELECT * FROM CARE_1 WHERE : ENTITIES ((p : PhysicalObjects) (e : Events)) CONDITIONS ((p involved_in e)(e's Name = "close_to_VendingMachine1"))
```

```
SELECT * FROM CARE_1 WHERE : ENTITIES ((p : PhysicalObjects) (e : Events)) CONDITIONS ((p's Class= "Person")(p involved_in e)(p's Id=57) (e's Name = close_to_VendingMachine1))
```

La première requête retrouve 361 résultats tandis que la deuxième requête ne rend aucun résultat. Les résultats de la deuxième requête montre que l'objet 57 n'est pas à côté de la machine de vente. Comme les requêtes de la catégorie 1, la qualité des résultats des requêtes dans cette catégorie dépend absolument de la qualité des modules d'analyse vidéo.

6.9.3 Résultats de recherche avec les requêtes de la catégorie 3

Au lieu de retrouver des objets par leurs étiquettes comme les requêtes des catégories 1, 2, les requêtes de la catégorie 3 permettent de retrouver les objets par leurs apparences. Les requêtes de cette catégorie peuvent être exprimées en utilisant les opérateurs d'apparence. Ces opérateurs sont construits en se basant sur l'extraction des descripteurs d'apparence (cf. chapitre 4) et la mise en correspondance entre objets (cf. chapitre 5).

La requête "Retrouver des personnes qui sont à côté de la machine de vente et semblables à une image d'exemple" est une requête de la catégorie 3. Elle est exprimée dans le langage proposé :

```
SELECT * FROM CARE_1 WHERE : ENTITIES ((p : PhysicalObjects),(i :
SubImage), (e : Events)) CONDITIONS ((e's Name = "close_to_VendingMachine1")
(p involved_in e)(p's Class = "Person")(i visual_distance p))
```

Le descripteur utilisé dans cette requête est les matrices de covariance (cf. voir la liste d'opérateurs dans annexe A).

Nous recevons 359 résultats qui sont ordonnés par leurs distances avec la requête. Nous montrons dans la figure 6.35 trois résultats retrouvés.



FIG. 6.35 – Image d'exemple est à gauche, trois objets retrouvés dont les étiquettes sont 176, 162, 111. Ces trois objets impliqués dans l'événement "close_to_VendingMachine1" aux frames 5940, 5895, et 3825 respectivement.

La requête suivante retrouve les personnes dans la vidéo CARE_2 impliquées dans l'événement "close_to_Gate1" et semblables à une image d'exemple par la couleur :

```
SELECT * FROM CARE_2 WHERE : ENTITIES ((p : PhysicalObjects),(i :
SubImage), (e : Events)) CONDITIONS ((e's Name = "close_to_Gate1")
(p involved_in e)(p's Class = "Person")(i color_similarity p))
```

Nous comparons dans la figure 6.36 les résultats obtenus en utilisant le descripteur d'apparence (nous l'appelons similarity matching) avec ceux en se basant sur l'étiquette (nous l'appelons exact matching) sur les 15 images d'exemple et 19 événements de *close_to_Gate1*.

La recherche basée sur les descripteurs d'apparence est plus performante que celle basée sur les étiquettes. La raison est que la recherche basée sur les descripteurs arrive à corriger les erreurs de la détection et du suivi d'objets correspondant à une valeur élevée de persistance d'étiquette.

6.9.4 Résultats de recherche avec les requêtes de la catégorie 4

Les requêtes de la catégorie 4 permettent d'enrichir l'indexation. Les événements complexes peuvent être retrouvés en combinant des événements simples reconnus. La requête suivante définit un événement complexe à partir de deux événements reconnus "inside_zone_Platform" et "close_to_VendingMachine1".

```
SELECT * FROM CARE_1 WHERE : ENTITIES ((e1 : Events), (e2 : Events))
CONDITIONS ((e1's Name = "inside_zone_Platform")
```

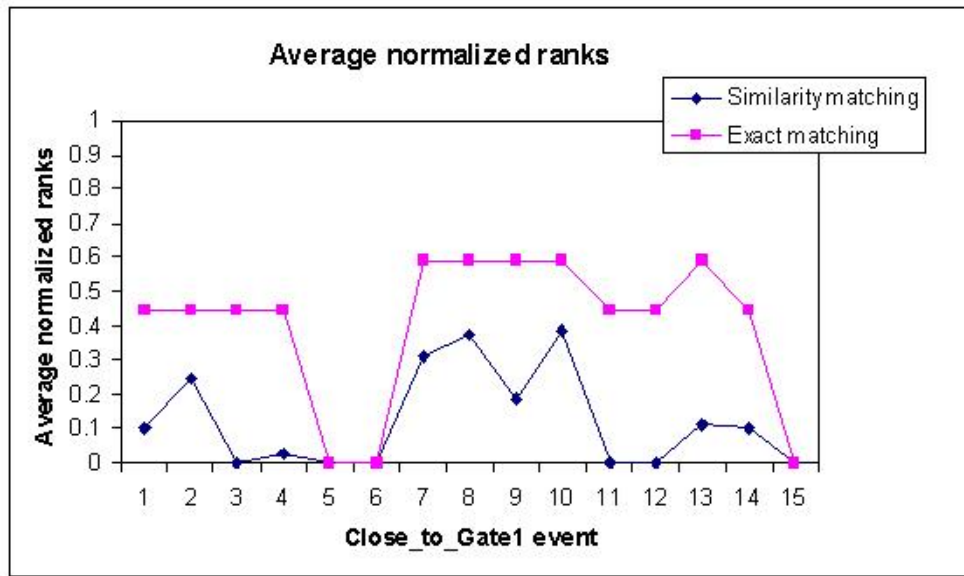


FIG. 6.36 – Rangs normalisés moyens obtenus de 15 requêtes sur 19 événements de *close_to_Gate1* pour la vidéo *CARE_2* du projet *CATETAKER*.

(e2's Name = "close_to_VendingMachine1")
(e1 having_same_objects e2) (e1 before e2)

La recherche avec cette requête nous rend 14412 séquences de deux instances dont une de l'événement "inside_zone_Platform" et l'autre de l'événement "close_to_VendingMachine1" qui ont vérifié la relation temporelle "before".

6.9.5 Résultats de recherche avec les requêtes de la catégorie 5

Il est à noter que les requêtes de la catégorie 4 utilisent les étiquettes. Les résultats de ces requêtes manquent les séquences des instances d'événements reconnus pour un seul objet réel, qui est cependant détecté et suivi comme plusieurs objets. Les requêtes de la catégorie 5 peuvent dépasser cette limitation. Une requête de cette catégorie est :

```
SELECT * FROM CARE_2 WHERE : ENTITIES ((e1 : Events), (e2 : Events),
(p1 : PhysicalObjects), (p2 : PhysicalObjects)) CONDITIONS ((e1 before e2) (e1's
Name = "inside_zone_Platform") (e2's Name = "close_to_Gate1") (p1 invol-
ved_in e1) (p2 involved_in e2) (p1 color_similarity p2))
```

Cette requête cherche des séquences d'événements *inside_zone_Platform* et *close_to_Gate1* qui vérifient la relation *before* et les objets impliqués dans les événements de *inside_zone_Platform* sont ressemblants aux ceux de *close_to_Gate1*. Nous comparons les résultats obtenus avec cette requête avec celle basée sur les étiquettes. La figure 6.37 montre les rangs obtenus.

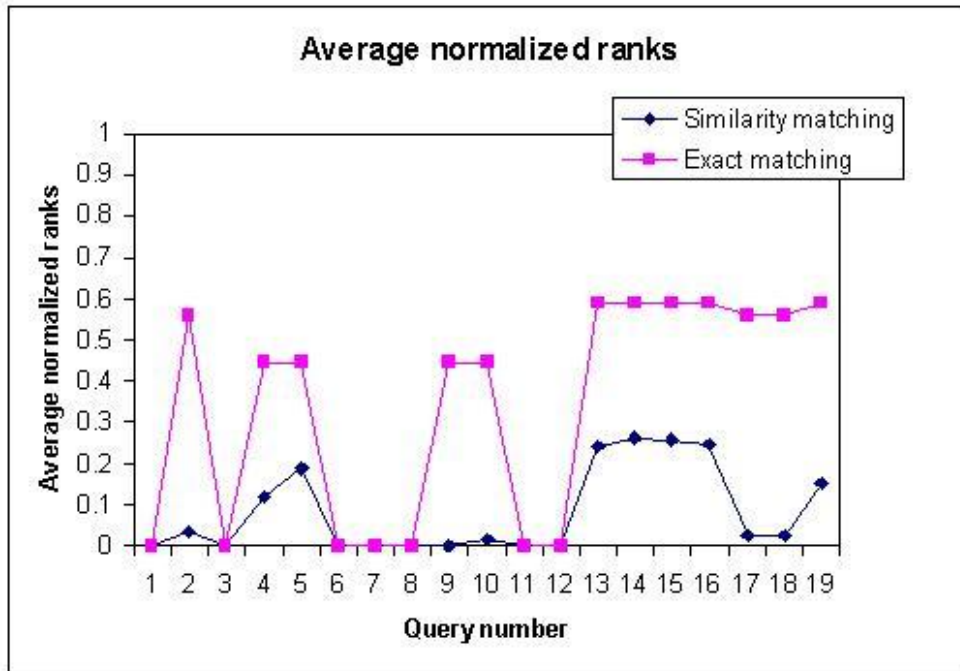


FIG. 6.37 – Rangs normalisés moyens obtenus pour 19 événements de *close_to_Gate1* et 29 événements de *inside_zone_Platform*.

6.9.6 Discussions

La relation entre l'objet et l'événement fournit une recherche riche et flexible. La recherche avec les requêtes des catégories 1, 2 et 3 vise à retrouver des objets avec l'information connue sur les événements ou des événements avec l'information connue sur les objets. La recherche correspondant aux requêtes des catégories 4 et 5 peut être utilisée pour enrichir l'indexation au niveau événements. L'indexation d'une vidéo qui est tout d'abord un ensemble des événements simples reconnus par les modules d'analyse vidéo peut être enrichie par les événements complexes retrouvés. Cela fait partie de notre travail futur. La comparaison entre la recherche basée sur les descripteurs et celle basée sur les étiquettes montre l'efficacité de la recherche basée sur les descripteurs.

6.10 Évaluation du retour de pertinence

Comme nous l'expliquons dans le chapitre 5, l'objectif du retour de pertinence est de trouver \mathcal{R}^t à partir de \mathcal{R}^0 qui vérifie deux propriétés :

- Le nombre de résultats pertinents de $\mathcal{R}^t(M)$ est supérieur à celui de $\mathcal{R}^{t-1}(M)$;
- Le nombre d'itérations est petit.

On peut donc évaluer des approches de retour de pertinence par les mesures d'évaluation à différentes itérations telles que le rang normalisé moyen. L'évaluation de chacune des approches de retour de pertinence est possible. Il est cependant difficile de comparer les performances des approches différentes même si elles ont été effectuées sur les mêmes bases de données et le même ensemble de requêtes. Puisque la recherche basée sur le retour de pertinence interagit avec l'utilisateur donc sa performance à chaque itération dépend fortement de deux facteurs : des résultats jugés par l'utilisateur et le nombre de résultats jugés par l'utilisateur à cette itération.

Dans certains cas, on peut fixer le nombre de résultats jugés à chaque itération. Cependant les M premiers résultats des approches différentes sont constitués par des éléments différents. L'utilisateur juge donc sur des données différentes. La comparaison des approches sur les mesures d'évaluation à quelques itérations n'exprime donc pas leur vraie performance. De plus, compte tenu des approches présentées dans l'état de l'art pour l'indexation et la recherche de vidéos de vidéosurveillance, aucune méthode de retour de pertinence n'est semblable à nos méthodes. À partir de cette analyse, nous présentons seulement des évaluations de notre méthodes.

Deux méthodes de retour de pertinence sont proposées (voir chapitre 5) : l'une se base sur plusieurs images d'exemple et l'autre se base sur les SVM. Les deux sections suivantes visent à présenter les résultats de ces deux méthodes.

6.10.1 Retour de pertinence basé sur plusieurs images d'exemple

Cette méthode fonctionne comme suit : parmi les M premiers résultats $\mathcal{R}^{t-1}(M)$ à l'itération $t - 1$, l'utilisateur choisit M_P blobs positifs. Un objet intermédiaire O_I est créé, sa représentation est $R_{O_I,ap} = \{(B_i, w_i)\} | i = 1, \dots, M_P$ où B_i est le blob positif. Pour w_i , nous initialisons simplement $w_i = \frac{1}{M_P}$. La méthode fait la mise en correspondance de l'objet O_I créé avec des objets de la base de données et rend à l'utilisateur de nouveaux résultats.

Afin d'évaluer automatiquement cette méthode avec un ensemble de requêtes, nous effectuons ce processus pour chacune des requêtes :

- nous identifions la vérité terrain en indiquant les objets pertinents pour la requête ;
- nous fixons la valeur des M premiers résultats considérés et le nombre d'itérations ;
- la méthode crée un objet de requête qui est initialisé par un seul blob qui est l'image d'exemple fournie par l'utilisateur ;
- la méthode met en correspondance l'objet recherché et les objets de la base de données, les ordonne de manière croissante avec leurs distances. La méthode se termine si un des trois critères sont vérifiés : (1) tous les objets pertinents sont dans les M premiers résultats ; (2) le nombre d'itérations est supérieur au nombre fixé ; (3) le nombre de résultats pertinents de $\mathcal{R}^t(M)$ n'est pas supérieur à celui de $\mathcal{R}^{t-1}(M)$. Sinon, la méthode choisit un blob dont le

poids est le plus élevé pour chaque objet pertinent qui est dans les M premiers résultats et met à jour la requête ;

- Les rangs moyens normalisés sont calculés à chacune des itérations.

Le premier critère montre que si la recherche est parfaite, aucune interaction n'est nécessaire. Le deuxième critère concerne l'utilisateur. Le nombre d'itérations doit être petit car l'utilisateur est habituellement impatient. Le troisième montre le cas où la qualité de recherche baisse après itérations. Comme nous n'utilisons que les exemples positifs, dans ce cas, aucun retour peut être donné par l'utilisateur pour la prochaine itération.

La figure 6.38 illustre le résultat obtenu par cette méthode sur 145 objets de CARE_1 avec 15 requêtes. M et le nombre d'itérations maximal autorisé sont 16 et 5 respectivement. La recherche est terminée après (0, 1, 2) itérations. Les résultats montrent que le retour de pertinence améliore la performance de recherche dans la plupart des cas. (les rangs normalisés moyens sont diminués après les itérations) sauf pour la requête #14 (voir figure 6.39). Nous expliquons ce cas. Dans la base de données, il existe un objet #1 dont les blobs représentatifs sont montrés dans la figure 6.40. Le suivi de cet objet n'est pas correct. Parmi les blobs représentatifs, un est semblable à la requête #14. Nous mettons donc cet objet dans la liste des objets pertinents de la requête #14. Lors de la première recherche (sans retour de pertinence), l'objet recherché #14 (voir figure 6.39.a) a un seul blob qui est l'image recherchée. La distance entre cet objet recherché et l'objet #1 n'est pas élevée, les rangs normalisés moyens sont petits. Pour l'itération #1, l'objet recherché contient 5 blobs (voir figure 6.39.b), dont la distance de cet objet avec l'objet #1 devient considérable. Cela augmente les rangs normalisés moyens.

Le résultat obtenu par cette méthode sur 810 objets de CARE_6 avec 50 requêtes provenant de la vidéo CARE_5 est montré dans la figure 6.41. Les valeurs de M et le nombre d'itérations maximal autorisé sont 100 et 5 respectivement. Le nombre d'itération est 0, 1, 2, 3. Nous affichons dans la figure 6.41 les résultats des requêtes dont le nombre d'itérations est supérieur à 1. Les résultats montrent que le retour de pertinence améliore considérablement la qualité de recherche d'objets.

Nous analysons le processus de recherche avec la requête #20. L'image d'exemple et les résultats sans retour de pertinence, ceux lors de la première itération et de la deuxième itération sont montrés dans la figure 6.42. Cette requête a huit objets pertinents. Les résultats sans retour de pertinence ne sont pas bons car les objets pertinents ont des rangs relativement élevés. Les objets #1086 et #1111 sont jugés comme des objets positifs. La requête lors de première itération contient 3 blobs. Les résultats lors de première itération sont considérablement améliorés (les rangs des objets pertinents sont diminués). Les objets pertinents dont le rang est inférieur à 100 ($M=100$) sont utilisés comme objets positifs pour la prochaine itération. Les rangs des objets pertinents lors de la deuxième itération sont progressivement diminués. Les rangs de la plupart des objets pertinents sont inférieurs à 10 après 2 itérations sauf l'objet #1393. L'apparence de cet objet est assez différente de ceux des autres. Cependant, son rang est diminué après le retour de pertinence (583, 355 et 201

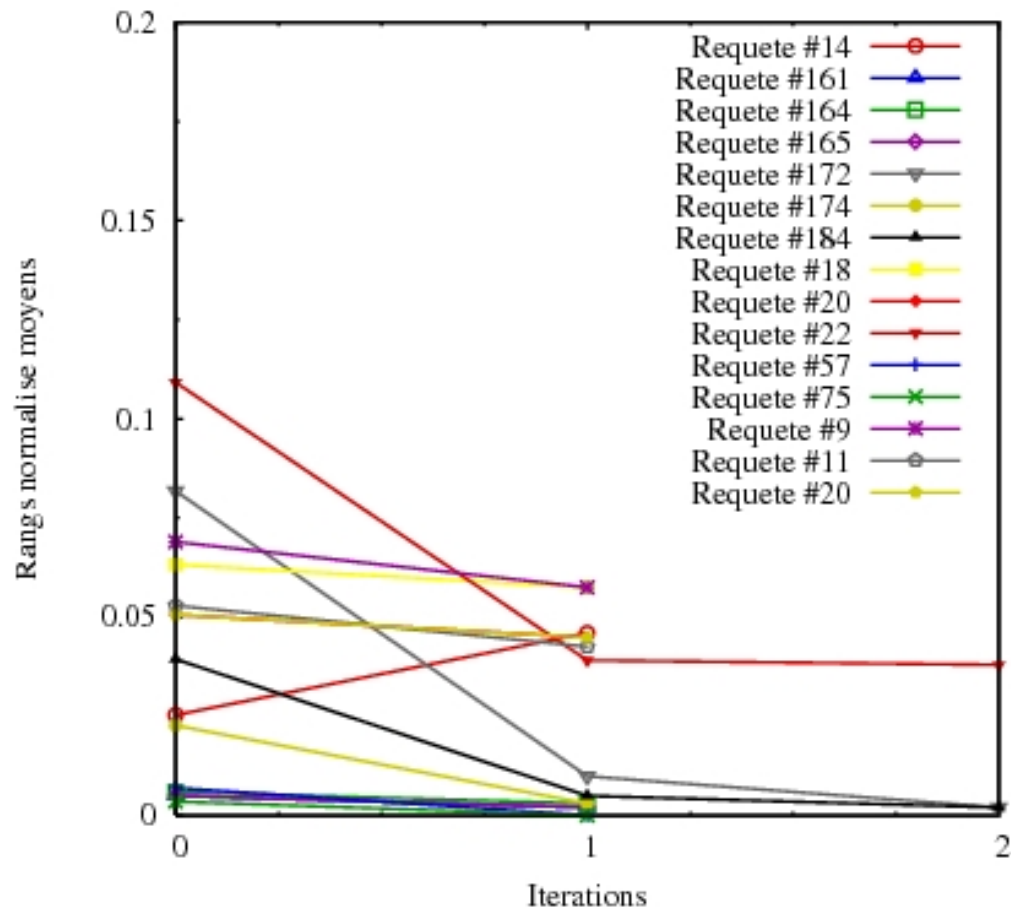


FIG. 6.38 – Rangs normalisés moyens pour la première méthode de retour de pertinence sur 145 objets de CARE_1 avec 15 requêtes. Les valeurs de M et le nombre d'itérations maximal autorisé sont 16 et 5 respectivement.



FIG. 6.39 – (a) requête #14 est initialisée par une image d'exemple ; (b) images positives dans la requête #14 lors de la première itération



FIG. 6.40 – Blobs représentatifs de l'objet #1.

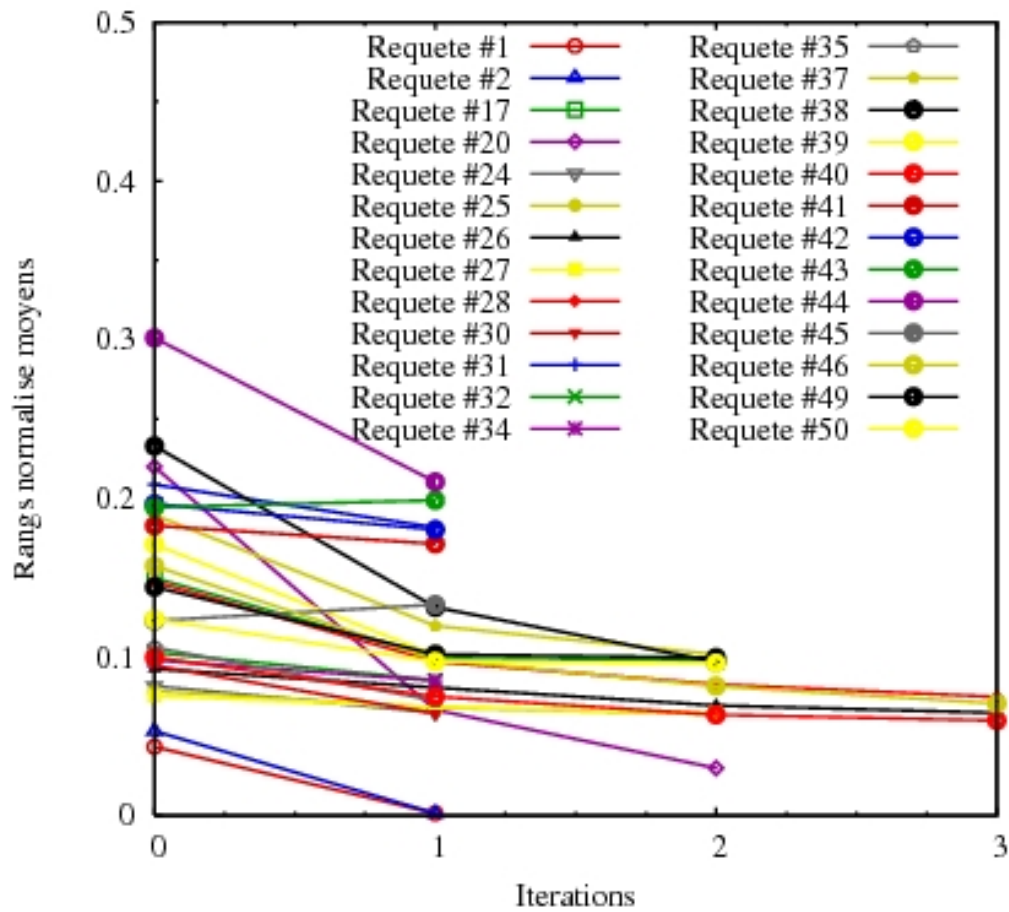


FIG. 6.41 – Rangs normalisés moyens pour la première méthode de retour de pertinence sur 810 objets de CARE_6 avec 50 requêtes provenant de la vidéo CARE_5. M et le nombre d'itérations maximal autorisé sont 100 et 5 respectivement.

respectivement sans retour de pertinence, après la première itération et la deuxième itération). Les objets jugés par l'utilisateur permettent au système de comprendre l'objet recherché par l'utilisateur ce qui n'est pas toujours évident avec une image d'exemple.

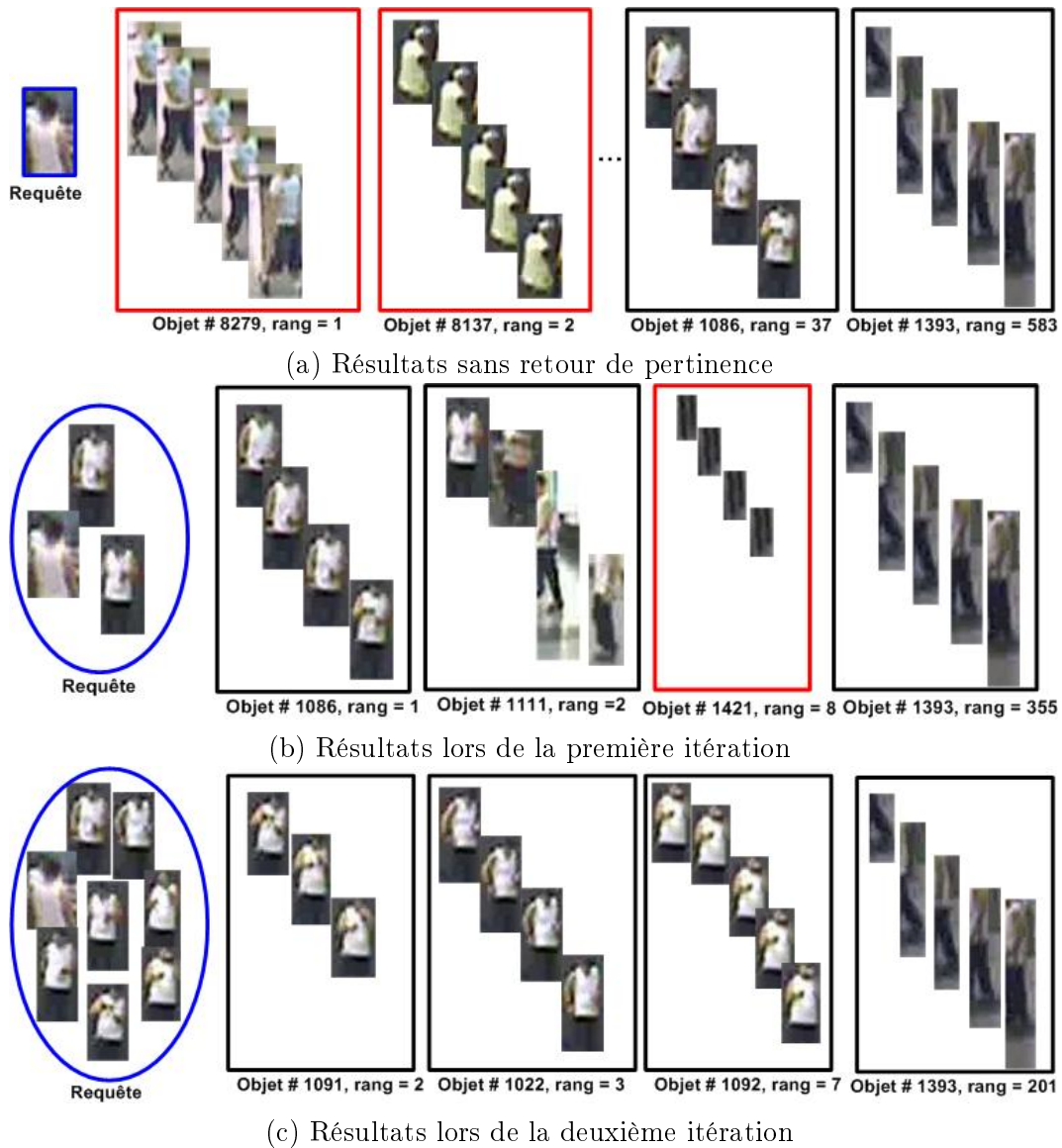


FIG. 6.42 – (a) résultats obtenus avec la requête #20 sans retour de pertinence ; (b) lors de la première itération ; (c) et de la deuxième itération ; (d) les requêtes en bleu, les objets non pertinents en rouge.

6.10.2 Retour de pertinence basé sur les SVM à une classe

Afin d'évaluer automatiquement le retour de pertinence basé sur les SVM avec un ensemble de requêtes, nous effectuons un processus semblable à celui de la méthode basée sur plusieurs images d'exemple sauf : si les trois critères ne sont pas vérifiés, les descripteurs sont calculés pour tous les blobs des objets positifs dans les M premiers résultats afin d'entraîner des SVM à une classe. Les objets dans la base de données sont ordonnés par leurs probabilités et rendus à l'utilisateur. La figure 6.43 montre le résultat obtenu avec cette méthode.

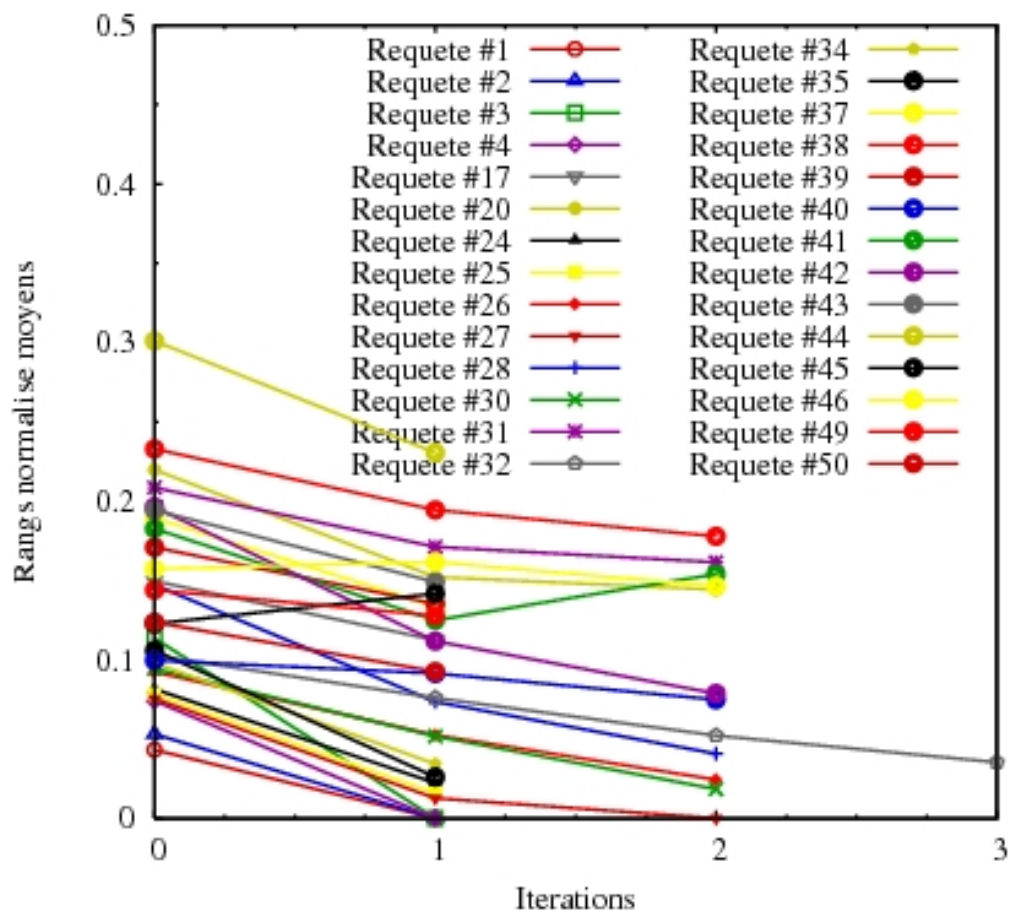


FIG. 6.43 – Rangs normalisés moyens pour le retour de pertinence basé sur les SVM à une classe sur 810 objets de CARE_6 avec 50 requêtes provenant de la vidéo CARE_5. M et le nombre d'itérations sont 100 et 5 respectivement.

Comme nous le voyons, les rangs normalisés moyens sont diminués après les (1,2,3) itérations.

Les résultats pertinents et donc les résultats jugés à chaque itération sont très peu nombreux. Cela peut poser problème pour l'entraînement des SVM. Nous utilisons

tous les blobs des objets positifs pour l'entraînement des SVM. La qualité du retour de pertinence basé sur les SVM à une classe dépend fortement de la qualité de la méthode de détection des blobs représentatifs. Pour cela, nous appliquons la méthode basée sur le regroupement des blobs qui est prouvée robuste à la détection et au suivi d'objets imparfaits.

Les SVM fournissent explicitement la connaissance apprise. Les SVM entraînés associés avec la requête d'un utilisateur peuvent être utilisés pour d'autres utilisateurs. Nous analysons cette possibilité dans nos perspectives.

6.10.3 Discussions

Les résultats obtenus avec retour de pertinence sont meilleurs que ceux sans retour de pertinence. L'information retournée par l'utilisateur peut être un blob ou un objet mobile.

Le retour de pertinence pour la vidéosurveillance peut être équivalent à celui de l'indexation et de la recherche d'images car une image (un ensemble de régions) est équivalent à un objet mobile (un ensemble de blobs). Cependant, il est à noter que pour l'indexation et la recherche d'images, une image jugée positive en général contient une (quelques) région(s) positive(s) tandis que pour la vidéosurveillance, un objet est positif, tous ses blobs sont positifs. Nous pouvons utiliser le blob le plus important (son poids est le plus élevé) ou tous les blobs des objets retournés.

Cependant, le retour de pertinence abordé dans cette thèse est à court terme. Le système oublie la connaissance apprise à chaque session de recherche. Dans le chapitre 7, nous présentons des pistes pour étendre notre approche en prenant en compte le retour de pertinence à long terme.

6.11 Conclusion

Après avoir analysé les résultats expérimentaux, nous résumons les caractéristiques suivantes de notre approche.

Le modèle de données proposé est général. Nous l'avons testé avec deux bases de vidéos différentes qui sont analysées par les deux modules d'analyse. Deux bases de vidéos : l'une provenant du projet CARETAKER et l'autre provenant du projet CAVIAR sont analysées à différents degrés. Toutes les 6 vidéos du projet CARETAKER sont analysées au niveau objets. Parmi ces 6 vidéos, 2 vidéos sont analysées aux niveaux objets et événements. Les 10 vidéos du projet CAVIAR sont traitées au niveau objets.

Concernant la détection des blobs représentatifs, les deux méthodes proposées sont complémentaires. Elles nous permettent de travailler avec des qualités différentes des modules d'analyse vidéo. La comparaison de notre méthode basée sur le regroupement des blobs et celle de Ma et al. [Ma 2007] montre que notre méthode améliore les résultats de la méthode de Ma et al. Les blobs sans objet sont enlevés avant d'effectuer la détection des blobs représentatifs. Cela assure que les blobs représentatifs sont des blobs pertinents.

Concernant le langage de requêtes, de nombreuses requêtes peuvent être exprimées ce qui montre l'expressivité du langage proposé.

Concernant la mise en correspondance, notre méthode basée sur la distance EMD est comparée avec deux méthodes de l'état de l'art : la méthode de Ma et al. [Ma 2007] et celle de Calderara et al. [Calderara 2006]. Notre méthode est plus performante que celle de Ma et al. surtout pour les vidéos ayant une valeur élevée de la confusion d'étiquette. Elle est également meilleure que la méthode de Calderara et al. si la détection et le suivi d'un objet ne réussissent pas pour tous les frames. Si l'on fixe le descripteur utilisé dans notre approche les matrices de covariance, la différence entre notre méthode et celle de Ma et al. est la mise en correspondance deux ensembles de blobs tandis que celle entre notre méthode et la méthode de Calderara et al. sont les descripteurs utilisés et la mise en correspondance. L'analyse locale de performance aux premiers résultats montre que la recherche de la même personne observée par la même caméra à différents moments est relativement bon. La recherche du même objet observé par différentes caméras à différents moments pourtant est loin d'être satisfaisante en raison de nombreux facteurs : les personnes sont loins des caméras ; elles ne sont pas bien détectées et suivies ; les méthodes de détection des blobs représentatifs ne sont pas parfaites ; les descripteurs utilisés ne sont pas discriminants. L'évaluation des descripteurs d'apparence en indexation et recherche de vidéos de vidéosurveillance nous montre une possibilité de combinaison des descripteurs.

Concernant le retour de pertinence, les premiers résultats obtenus avec le retour de pertinence à court terme sont présentés. Cela nous motive pour travailler avec le retour de pertinence à long terme ce qui est présenté dans les perspectives.

Conclusions et perspectives

Dans ce chapitre, nous résumons tout d'abord nos contributions. Ensuite, nous analysons les limitations de l'approche proposée et discutons les résultats obtenus. Enfin, nos perspectives à court terme et à long terme sont présentées.

7.1 Résumé des contributions

Notre première contribution est un **modèle de données** pour l'indexation et la recherche de vidéos de vidéosurveillance. Le modèle de données est composé de deux concepts abstraits : objets et événements. Le modèle de données proposé possède deux bonnes propriétés. Premièrement, il est indépendant des algorithmes de vision des modules d'analyse vidéo. Quels que soient les algorithmes de vision utilisés, si leurs sorties sont des objets détectés (et des événements reconnus), le modèle de données proposé peut être appliqué. Dans le chapitre 6, nous avons présenté l'utilisation du modèle de données pour deux modules d'analyse vidéos. Deuxièmement, il peut travailler avec les modules d'analyse vidéos à différents niveaux d'analyse. Nous avons employé 16 vidéos analysées au niveau objets dont 10 provenant du projet CAVIAR et 6 provenant du projet CARETAKER, et 2 vidéos analysées aux deux niveaux : objets et événements.

La deuxième contribution est un **nouveau langage de requêtes**. En se basant sur le modèle de données, le langage de requêtes proposé permet de formuler les requêtes à trois niveaux : images, objets et événements. De nombreuses requêtes peuvent être exprimées dans le langage proposé ce qui prouve son expressivité.

La troisième contribution concerne les méthodes de **détection des blobs représentatifs** pour la représentation des objets. Les deux méthodes de détection des blobs représentatifs proposées permettent (1) de choisir les blobs pertinents pour chaque objet (2) de réduire des informations stockées. Nous avons analysé les propriétés de ces deux méthodes.

La quatrième contribution est une **nouvelle méthode de mise en correspondance des objets** basée sur la distance EMD (Earth Movers Distance). Cette méthode est supérieure à celles de Ma et al. [Ma 2007] et de Calderara et al. [Calderara 2006]. Elle est évidemment appropriée dans les cas où la valeur de la confusion d'étiquette est élevée.

La cinquième contribution concerne le **retour de pertinence**. Deux méthodes de retour de pertinence à court terme qui se basent sur les objets sont proposées. Nous avons montré que les deux méthodes améliorent remarquablement le résultat de recherche.

7.2 Limitations

Cette approche proposée possède cependant 3 limitations.

La première limitation concerne la qualité de la recherche. La recherche d'objets demeure un problème ouvert. Elle n'emploie que des descripteurs d'apparence afin de mettre en correspondance les objets. De plus, un seul type de descripteur d'apparence est utilisé à la fois. D'autres informations telles que les positions en 3D, la taille de l'objet ne sont pas encore analysées. L'approche est évaluée sur les vidéos provenant d'une seule caméra. Plusieurs aspects d'une personne peuvent être bien observés en utilisant un réseau de caméras. La mise en correspondance doit être modifiée afin de prendre en compte l'information provenant de plusieurs caméras.

La deuxième limitation concerne le temps de calcul. Une grande réduction des informations est faite par la détection des blobs représentatifs. Cette réduction nous permet de diminuer le temps de mise en correspondance entre objets. Cependant, aucune technique d'indexation n'est appliquée. Cette limitation doit être abordée lorsqu'on travaille avec de grandes bases de vidéos acquises et avec des applications où la recherche concerne des situations urgentes.

La troisième limitation est la convivialité de l'approche. Il n'est pas facile pour les utilisateurs novices d'exprimer les requêtes dans le langage proposé. En plus, l'information apprise à partir des interactions avec l'utilisateur n'est pas stockée et réutilisée.

7.3 Discussions

7.3.1 Relation entre l'indexation et la recherche de vidéos et les modules d'analyse vidéo

Nous analysons une relation bidirectionnelle entre l'indexation et la recherche de vidéos et les modules d'analyse vidéo pour la vidéosurveillance. Cette relation existe en permanence. Cependant, elle est cachée pour les travaux de l'état de l'art. Pour la première fois, nous séparons l'indexation et la recherche de vidéos et des modules d'analyse vidéo. Nous analysons leurs propres caractéristiques et mettons en évidence leur relation.

Le premier point est que la qualité d'indexation et de recherche dépend de celle des modules d'analyse vidéo surtout la détection et le suivi d'objets. Cela est inévitable. Les travaux présentés dans l'état de l'art sont également dépendants d'un module d'analyse de vidéo. L'objectif de cette thèse est que, quel que soit la qualité des modules d'analyse vidéo, les phases d'indexation et de recherche vont la compenser. L'approche proposée est prévue pour travailler avec des modules d'analyse vidéo ayant des qualités différentes. Les résultats de recherche analysés dans le chapitre 6 montrent l'efficacité de l'approche. Pour la détection et le suivi d'objets, notre approche permet de bien retrouver des objets quand ils ne sont pas bien détectés et suivis (les valeurs de la persistance et de la confusion d'étiquette sont élevées). Pour la reconnaissance d'événements, l'approche reconnaît des événements complexes à

partir des événements simples reconnus.

Le deuxième point est que les résultats de l'indexation et de la recherche peuvent être utilisés pour corriger et donc pour améliorer les résultats des modules d'analyse.

Cette thèse se focalise sur le premier point. Nos travaux futurs visent à aborder la deuxième direction de recherche.

7.3.2 Descripteurs visuels

Une mesure de similarité d'un descripteur est présentée lorsque ce descripteur est proposé. Un descripteur peut être évalué par deux mesures : sa taille et sa performance. L'évaluation d'un descripteur par sa taille est simple alors que l'analyse de sa performance est difficile. Habituellement, les descripteurs sont évalués par la qualité de la recherche ou la capacité de la reconnaissance.

En indexation et recherche d'images et de vidéos, un descripteur est idéal si la distance basée sur ce descripteur entre deux images (vidéos) qui sont visuellement similaires est plus petite que celle entre deux images (vidéos) quelconques. Cependant, la définition de "visuellement similaire" est différente entre l'humain et l'ordinateur. De plus, la plupart des descripteurs proposés assurent que la distance entre deux images visuellement similaires est petite. Mais ils ne garantissent pas que cette distance est plus petite que celle entre deux images quelconques. Les résultats de recherche sont ordonnés par leurs distances avec la requête c'est-à-dire les rangs des résultats sont importants. Il est à noter qu'une petite différence de distance entre deux résultats peut causer une grande différence de leurs rangs. Cela explique pourquoi la performance des descripteurs dépend de la caractéristiques de la requête et aussi de celle de la base de vidéos.

Dans le chapitre 6, nous avons évalué la performance de 4 types de descripteurs (les couleurs dominantes, les histogrammes des contours comprenant l'histogramme local, semi-local, global et composé, les matrices de covariance, les points d'intérêt comprenant MSER, HarrisAffine, DoG). L'évaluation montre que aucun descripteur n'est supérieur à tous les autres descripteurs pour toutes les requêtes. L'évaluation montre aussi que les descripteurs sont complémentaires.

Cette analyse nous conduit à deux pistes de recherche : le choix de descripteur approprié pour une requête et la fusion des descripteurs. Nous analysons ces pistes dans les perspectives.

7.3.3 Mise en correspondance entre des objets

La mise en correspondance entre deux objets dans la vidéosurveillance devient celle entre deux ensembles de blobs pondérés. Sa qualité dépend de trois facteurs. Le premier facteur est la similarité de chaque paire de blobs. Ce facteur est décidé par le choix de descripteur d'apparence utilisé. Le deuxième facteur est la manière de calculer la distance finale entre deux ensembles de blobs en se basant sur la distance entre chaque paire de blobs. La manière utilisée dans cette thèse se base sur une solution optimale. Le troisième facteur est la détermination des poids des blobs. Plus

le poids est élevé, plus le blob est pertinent. Cependant, si les résultats de la détection et du suivi sont très bruités, le blob ayant un poids élevé peut être non pertinent. Ce problème est partiellement résolu par la détection des blobs représentatifs basée sur le regroupement des blobs dans la phase d'indexation.

Notre méthode est supérieure à celle de Ma et al. [Ma 2007] pour la raison qu'elle a une manière appropriée et qu'elle tient compte du poids des blobs. La méthode de Calderara et al. [Calderara 2006] est différente de notre méthode pour tous les trois facteurs. Concernant le descripteur, elle a utilisé les modes de l'histogramme. Le vecteur de descripteurs d'un objet est mis à jour pour ensemble des blobs. La mise en correspondance entre deux objets devient une comparaison de deux vecteurs de descripteurs. Le poids des blobs est implicitement pris en compte par les poids des gaussiennes. Cependant, cette méthode prend tous les blobs des objets pour mettre à jour leurs gaussiennes. Cela n'est pas approprié si les erreurs de la détection et du suivi d'objets sont présentes pour un grand nombre de blobs. C'est pourquoi l'approche de Calderara et al. est moins performante que la nôtre dans ces cas.

À partir de cette analyse, la qualité de recherche peut être améliorée soit par le choix du descripteur approprié soit par l'amélioration de la méthode de détection des blobs représentatifs. Nous analysons ces possibilités dans nos perspectives.

7.4 Perspectives

L'objectif de cette section est de présenter les possibilités d'étendre l'approche proposée. Nous classifions les extensions en deux types : court terme et long terme.

7.4.1 Perspectives à court terme

Pour les perspectives à court terme, nous présentons 4 perspectives dont 3 pour la phase d'indexation et 1 pour celle de recherche.

7.4.1.1 Amélioration de la détection des blobs représentatifs

La méthode de détection des blobs représentatifs basée sur le regroupement des blobs effectue la classification des blobs en blobs avec objets et ceux sans objet. Pour cela, nous avons choisi les SVM à deux classes avec l'histogramme local des contours. Cependant, si les objets d'intérêt sont des personnes, d'autres descripteurs peuvent être extraits pour cette méthode. Le descripteur HoG (Histogram of Oriented Gradients) prouve sa performance en détection de personnes [Dalal 2005]. Les blobs sans personne peuvent être efficacement enlevés en utilisant ce descripteur surtout si l'on travaille avec les vidéos analysées par les modules d'analyse où la méthode de détection des personnes n'utilise pas ce descripteur.

7.4.1.2 Techniques d'indexation

Nous avons évalué la performance de l'approche proposée en fonction de la qualité de la recherche. Cependant le temps est également un facteur important en

indexation et recherche d'informations. Pour l'indexation et la recherche de vidéos de vidéosurveillance, cela devient indispensable car la personne impliquée dans un événement dangereux doit être retrouvée le plus vite possible. De plus, de nombreuses vidéos sont acquises et stockées jour après jour. Une de nos perspectives est d'étudier les techniques d'indexation telles que celles présentées dans [Berrani 2002] afin d'accélérer la vitesse de la recherche. Une technique d'indexation peut être appliquée aux concepts du modèle de données ou à la représentation d'objets par les descripteurs d'apparence ou temporels.

7.4.1.3 Analyse des trajectoires

Nous présentons et obtenons des premiers résultats pour l'analyse et la mise en correspondance entre des trajectoires. Cependant, cette analyse dépend des points de début de trajectoire. Nous ne l'évaluons que sur une base de trajectoires prédéfinies. En travaillant avec les trajectoires des objets qui sont préparées par les modules d'analyse vidéo, il faut prendre en compte les trajectoires incomplètes et erronées. Il est nécessaire d'étendre les analyses de trajectoires. Cette extension peut être une nouvelle représentation de trajectoires ou une nouvelle méthode de mise en correspondance.

7.4.1.4 Ontologie pour le langage de requêtes

Le langage de requêtes se base implicitement sur une ontologie. Cette ontologie doit être bien définie. La définition d'une ontologie a un double objectif. D'une part, elle décrit les concepts du modèle de données. Puisque nous envisageons de travailler avec plusieurs modalités, cette ontologie doit être étendue. D'autre part, il n'est pas facile pour les utilisateurs novices d'exprimer leurs requêtes dans le langage proposé. Une interface associée à cette ontologie peut être construite. Cette interface doit permettre à l'utilisateur de comprendre et de choisir facilement les concepts, les fonctions d'accès et les prédicats pour exprimer ses requêtes.

7.4.2 Perspectives à long terme

Dans les perspectives à long terme, nous souhaitons étudier 5 pistes de recherche.

7.4.2.1 Indexation en ligne

Dans certaines applications de vidéosurveillance, l'indexation en ligne est cruciale. L'indexation en ligne est une indexation qui s'effectue pendant l'analyse de vidéos. Par exemple, si le système de vidéosurveillance détecte un vol dans une station de métro et la personne soupçonnée de ce vol n'est plus observée par la caméra dans cette station. Le personnel de sécurité veut savoir si cette personne est observée par d'autres caméras dans d'autres endroits de la station. L'indexation en ligne a deux caractéristiques : elle ne prend pas de temps et elle ne doit pas extraire autant

de descripteurs que celle hors ligne. La rapidité de la réponse est un critère primordial. L'indexation en ligne doit de répondre le plus vite possible aux requêtes du personnel de sécurité. De plus, elle doit profiter des observations de chaque caméra du réseau de caméras. Notre approche peut être étendue pour faire l'indexation en ligne : le suivi d'objets, la détection des blobs représentatifs et la mise en correspondance entre objets utilisent le même type de descripteur. Avec cela, nous profitons des descripteurs extraits dans la détection et le suivi d'objets pour l'indexation et la recherche. Nous présentons ici deux possibilités. L'une est d'utiliser des matrices de covariance. Le suivi des objets en les utilisant a été présenté par [Porikli 2006a] [Porikli 2006b]. Dans la phase d'indexation, la détection des blobs représentatifs basée sur le changement d'apparence peut être effectuée en ligne. Pour la mise en correspondance entre objets (voir section 5.4.2 du chapitre 5), la distance des blobs est la distance de leurs matrices de covariance. L'autre possibilité est d'employer des points d'intérêts. La méthode de Trichet et al. [Trichet 2008] consiste à suivre les objets en utilisant des points d'intérêts. Les points d'intérêt calculés pour la détection des objets seront utilisés pour la détection des blobs représentatifs et la mise en correspondance entre objets.

7.4.2.2 Fusion/choix de descripteurs

Nous avons effectué plusieurs expérimentations sur les descripteurs visuels : les descripteurs de MPEG-7 (les couleurs dominantes, les histogrammes de contours), les points d'intérêts, et les matrices de covariance. Les résultats obtenus montrent que les matrices de covariance obtiennent les meilleurs résultats dans la plupart des cas. Cependant, chaque descripteur possède ses points forts et aussi ses points faibles. Concernant l'utilisation de descripteurs en indexation et recherche de vidéos de vidéosurveillance, nous analysons deux propriétés requises.

Premièrement, comme l'utilisateur (p. ex. les personnels de sécurité) en général ne comprend pas la description de descripteurs. Il ne sait pas donc choisir le descripteur approprié. L'approche d'indexation et de recherche doit déterminer par avance le descripteur utilisé.

Deuxièmement, à la différence de l'indexation et de la recherche d'images, la contrainte sur le rang des résultats retrouvés en indexation et recherche en vidéosurveillance est plus forte en raison de deux facteurs. Le premier facteur est que les résultats de recherche dans ce domaine concernent la sécurité. Dans certains cas, la recherche est urgente (p. ex. le vol, la bagarre dans les stations des métros). Le personnel de sécurité doit recevoir les résultats pertinents dans un petit ensemble des premiers résultats. Il ne peut pas prendre du temps pour naviguer dans une grande liste de résultats. Le deuxième facteur est que le personnel de sécurité est impatient pour naviguer des résultats. Il doit surveiller le système jour et nuit. Même s'il ne doit pas regarder en tout temps les écrans. Si les premiers résultats ne sont pas pertinents, il n'est pas volontaire pour voir d'autres résultats retrouvés.

En plus, avec les caractéristiques de la nature des vidéos et celle des modules d'analyse vidéo, il est difficile d'obtenir ces deux propriétés.

Dans le chapitre 6, nous avons montré que même si l'analyse vidéo est très bonne (nous avons utilisé les annotations manuelles du projet CAVIAR), la recherche d'objets en utilisant des descripteurs d'apparence est encore loin d'être parfaite.

Afin d'améliorer la performance, il y a deux possibilités. La première possibilité est d'utiliser un nouveau descripteur. Le nouveau descripteur doit être capable de capturer tous les aspects d'apparence. Les nouveaux descripteurs comprenant l'information en 3D et l'information d'apparence peuvent être analysés [Gandhi 2007], [Gheissari 2006]. La deuxième possibilité est de fusionner plusieurs descripteurs. À partir des résultats expérimentaux, nous trouvons que chacun des descripteurs arrive à retrouver des objets ayant des conditions différentes. Si l'apparence des objets n'est pas changée à différents points de vue, les couleurs dominantes et les matrices des covariance peuvent être employées. Si la posture de l'objet n'est pas largement changée, les histogrammes de contours sont appropriés. Quand l'occultation entre objets ou entre les objets mobiles et les objets de contexte est présente, les points d'intérêt sont pertinents. La fusion de ces descripteurs est donc prometteuse. Nos travaux futurs envisagent de fusionner différents descripteurs. Concernant la fusion de plusieurs descripteurs, deux types de fusion sont possibles : la fusion précoce et la fusion tardive. La fusion précoce vise à intégrer plusieurs vecteurs de descripteurs en un seul vecteur de descripteurs. La mise en correspondance entre objets basée sur la distance EMD sera calculée sur ce nouveau vecteur. La fusion tardive calcule la distance EMD pour chacun des descripteurs. Ces distances vont être combinées pour avoir une distance finale. Pour la fusion précoce, la distance EMD n'est calculée qu'une seule fois. Cependant, la fusion précoce n'est pas appropriée si les descripteurs ont des mesures de similarité différentes. La fusion tardive permet d'associer un poids à chaque descripteur. Elle calcule n mises en correspondance où n est le nombre de descripteurs.

7.4.2.3 Retour de pertinence à long terme

Les deux méthodes de retour de pertinence proposées dans cette thèse sont à court terme. Une de nos perspectives est de faire un retour de pertinence à long terme. À la différence du retour de pertinence à court terme, celui à long terme utilise la connaissance apprise lors d'une session de recherche pour d'autres sessions de recherche (du même ou de différents utilisateurs). Avant de présenter nos perspectives pour le retour de pertinence, il est à noter deux caractéristiques en indexation et recherche de vidéos pour la vidéosurveillance. Premièrement, le retour de l'utilisateur est quantitativement petit. Les résultats pertinents d'une personne recherchée sont la même personne observée à différents moments ou par différentes caméras. Le nombre de ces résultats est mineur. Pour l'indexation et la recherche d'images surtout d'images de tourisme, ces résultats peuvent être très nombreux. Deuxièmement, ce retour est fiable et cohérent. La définition de "visuellement similaire" est claire. Deux personnes indexées sont similaires si elles représentent la même personne réelle. Le retour d'un utilisateur (qui est en général un personnel de sécurité) est cohérent avec celui d'autres utilisateurs. Nous présentons 2 pistes pour

étendre notre approche avec un retour de pertinence à long terme.

La première piste consiste à étudier le degré d'importance de chaque descripteur en se basant sur les retours. Ce degré d'importance d'un descripteur doit montrer sa capacité à bien distinguer les résultats pertinents de ceux non pertinents. Il sera pris en compte dans la fusion de descripteurs. Pour cette piste, nous pouvons étendre la méthode de retour de pertinence basée sur plusieurs images d'exemple présentées dans cette thèse. La mise en correspondance entre objets dans cette méthode doit pouvoir fusionner plusieurs descripteurs.

La deuxième piste concerne la méthode de retour de pertinence basée sur les SVM. Les SVM entraînés avec l'objet recherché et les objets pertinents peuvent être stockés à la fin de la session comme un descripteur pour l'objet recherché et les objets pertinents. Si un de ces objets devient un objet recherché, ces SVM entraînés seront utilisés pour la première itération de recherche. Ces SVM doivent être progressivement mis à jour tout au long des sessions de recherche.

7.4.2.4 Amélioration de la qualité de l'information indexée

Comme nous le discutons dans la section précédente, cette thèse se focalise sur l'utilisation des résultats des modules d'analyse vidéo pour l'indexation et la recherche. Nous nous intéressons à l'autre direction : l'utilisation des résultats de la recherche pour l'analyse vidéo. Notre objectif est d'une part de corriger l'indexation et d'autre part de l'enrichir. Dans cette direction, le travail de Chau et al. [Chau 2009] vise à corriger les trajectoires erronées. Les zones dans une scène telles que la zone entrée/sortie, la zone de perte (où les objets sont habituellement perdus) et la zone de récupération (où les nouveaux objets sont habituellement détectés de nouveau) sont apprises. Si le système détecte un objet qui apparaît dans une zone de récupération, la trajectoire de cet objet va être associée à celle de l'objet précédemment perdu dans la zone de perte. Nous proposons deux possibilités dans cette direction.

La qualité du suivi d'objets est mesurée par la persistance et la confusion d'étiquette. Plus ces mesures sont élevées, plus faible est la qualité du suivi des objets. La persistance d'étiquette mesure le nombre d'objets réels dans la scène qui sont détectés et suivis comme un seul objet tandis que la confusion d'étiquette montre le nombre d'objets détectés et suivis correspondant à un seul objet réel. La mise en correspondance entre objets présentée dans le chapitre 5 avec les résultats de recherche du chapitre 6 montrent une possibilité d'améliorer l'indexation qui se base sur les résultats du suivi d'objets par la recherche d'objets. L'objectif est de lier des objets détectés correspondant à un seul objet réel. En prenant chaque fois un objet indexé comme un objet recherché, les objets retrouvés sont les candidats de l'aspect visuel pour la compensation. D'autres critères sur la position, le temps peuvent être utilisés afin de vérifier ces candidats. La participation de l'utilisateur est facultative. L'utilisateur peut y participer en validant les candidats retrouvés. Pour cela, l'approche doit fusionner plusieurs objets indexés en un seul objet. Cette fusion peut se baser sur l'ordre dans le temps de ces objets.

Le langage de requête proposé nous permet d'exprimer un événement complexe à partir des événements simples reconnus. Les résultats retrouvés peuvent être utilisés pour enrichir l'indexation. Afin de stocker ces résultats, ils seront convertis au format défini par le concept Events dans le modèle de données. Ce processus s'effectue progressivement à chaque session de recherche. L'information sera temporairement stockée pour une personne. Elle peut être définitivement stockée dans la base de données à l'aide d'une ontologie partagée. Pour cela, la définition d'un nouvel événement (exprimé dans la requête) doit être rajoutée dans la liste d'événements.

7.4.2.5 Multimodalité pour l'indexation et la recherche de vidéos de vidéosurveillance

L'une des hypothèses de l'approche proposée est qu'elle n'emploie que des informations visuelles. L'utilisation d'autres types de descripteurs est cruciale pour certaines applications telles que GERHOME (GERrontology at HOME)¹ [Zouba 2008]. Cependant, peu de travaux sont présentés pour l'analyse des vidéos en utilisant la multimodalité [Cristani 2007]. L'utilisation de multimodalités demande de résoudre le problème de la synchronisation des modalités. Comme l'analyse de chaque modalité n'est pas parfaite, l'analyse des multimodalités est encore un problème ouvert. Les résultats de cette analyse fournissent une indexation riche. Notre approche peut être étendue afin de prendre en compte la multimodalité. Concernant le modèle de données, notre modèle de données contient deux concepts : les objets et les événements. D'autres concepts qui nous permettent de caractériser d'autres modalités telles que l'acoustique doivent être rajoutés. Pour la représentation d'objets et l'extraction de descripteurs, il est nécessaire d'analyser et d'extraire des descripteurs auditifs appropriés. La mise en correspondance présentée dans cette thèse doit prendre en compte la similarité des concepts auditifs. Cette mise en correspondance peut être inspirée à partir des travaux sur la multimodalité en indexation et recherche de journaux télévisés [Snoek 2005]. De nouveaux prédicats et de nouvelles fonctions d'accès doivent être rajoutés dans le langage de requête.

¹<http://gerhome.cstb.fr>

Langage de requêtes

Nous présentons dans cette section, 18 fonctions d'accès dont 9 pour les objets mobiles, 1 pour les images d'exemple, 8 pour les événements, 7 opérateurs d'apparence entre deux objets ou entre un objet et une image, 3 opérateurs d'appartenance entre deux événements ou entre un événement et un objet. En plus de ces fonctions d'accès et ces opérateurs, le langage de requête comprend 13 opérateurs temporels qui sont présentés dans le chapitre 5, 6 opérateurs de comparaison des valeurs numériques et des chaînes de caractères ($=$, \neq , $>$, $<$, \geq , \leq), 4 opérateurs arithmétiques ($+$, $-$, $*$, $/$). Parmi les 7 opérateurs d'apparence, l'opérateur "Visual_matching" est créé pour que l'utilisateur puisse exprimer les requêtes sans indiquer le descripteur préféré. En effet, cet opérateur va utiliser la matrice de covariance afin de mettre en correspondance entre deux objets ou entre un objet et une image d'exemple. Puisque l'évaluation des descripteurs en recherche d'objets montre que la matrice de covariance est la plus performante.

TAB. A.1 – Fonctions d'accès et opérateurs d'apparence et ceux d'appartenance définis dans le langage de requêtes

Nom	Action
9 fonctions d'accès pour les objets mobiles	
's Id	accéder l'attribut "Id"
's Class	accéder l'attribut "Class"
's 2DPositions	accéder l'attribut "2D_positions"
's 3DPositions	accéder l'attribut "3D_positions"
's Blobs	accéder l'attribut "Blobs"
's Weights	accéder l'attribut "Weights"
's i_	accéder le moment auquel l'objet est détecté et suivi
's _i	accéder le moment à partir duquel l'objet n'est plus détecté et suivi
's Duration	calculer la durée pendant laquelle l'objet est détecté et suivi
1 fonction d'accès pour les images d'exemple	
's Image	accéder l'attribut "Image"
8 fonctions d'accès pour les événements	
's Id	accéder l'attribut "Id"
's Name	accéder l'attribut "Name"
's Confident_value	accéder l'attribut "Confidence_value"
's Listobjects	accéder l'attribut "Involved_Physical_objects"
's Subevents	accéder l'attribut "Sub_events"
's i_	accéder le moment auquel l'événement commence
's _i	accéder le moment auquel l'événement se termine
's Duration	calculer la durée pendant laquelle l'événement est reconnu
6 opérateurs d'apparence pour les objets et les images d'exemple	
DoCo_matching	mettre en correspondance par les couleurs dominantes
EH_matching	mettre en correspondance par histogrammes de contours
CM_matching	mettre en correspondance par la matrice de covariance
DoG_matching	mettre en correspondance par les points d'intérêt DoG
Haraff_matching	mettre en correspondance par les points d'intérêt Haraff
Mser_matching	mettre en correspondance par les points d'intérêt MSER
Visual_matching	mettre en correspondance par la matrice de covariance
3 opérateurs d'appartenance pour les événements et les objets	
involved_in	déterminer si un objet impliqué dans un événement
having_same_objects	déterminer si deux événements ont le même objet
is_subevent_of	déterminer si un événement est un sous événement d'un autre événement

Implémentation de l'approche proposée

Nous avons développé dans cette thèse un prototype en C++ pour l'indexation et la recherche de vidéos pour la vidéosurveillance. L'environnement de travail est montré dans le tableau B.1.

TAB. B.1 – Environnement de travail

Environnement	Linux Fedora Core 5 (kernel)
Compilateur	g++ v4.1.1
Hardware	Intel Xeon bi-processeurs double core à 2.33GHz avec 4Go de RAM

Le prototype développé comprend 3 bibliothèques en C++ nommées **libdescriptors**, **libutilities** et **libqlanguage** contenant 5 modules (extraction de descripteurs, détection des blobs représentatifs, distance EMD, langage de requêtes et retour de pertinence). La bibliothèque **libdescriptors** ayant 12530 lignes de code contient 3 modules : extraction de descripteurs, détection des blobs représentatifs et retour de pertinence. La bibliothèque **libutilities** avec 6806 lignes de code contient l'implémentation de distance EMD. En plus, cette bibliothèque contient les implémentations des mesures d'évaluation. La bibliothèque **libqlanguage** comprenant 1811 lignes de code est dédiée à l'implémentation du langage de requêtes.

Notre prototype s'appuie sur les logiciels utilisés : une bibliothèque en C++ du projet PULSAR, l'implémentation EMD en C de Rubner, la fonction de regroupement agglomératif en Matlab, les détecteurs des points d'intérêts en code binaire et la bibliothèque ltilib en C++. Nous détaillons ces logiciels dans la section suivante. La figure B.1 montre la relation entre les 5 modules dans notre prototype et les logiciels utilisés.

B.1 Logiciels utilisés

B.1.1 Bibliothèque ltilib

La bibliothèque ltilib (<http://ltilib.sourceforge.net/doc/homepage/index.shtml>), une bibliothèque en C++ de l'université Aachen en Allemagne, comprend des structures de

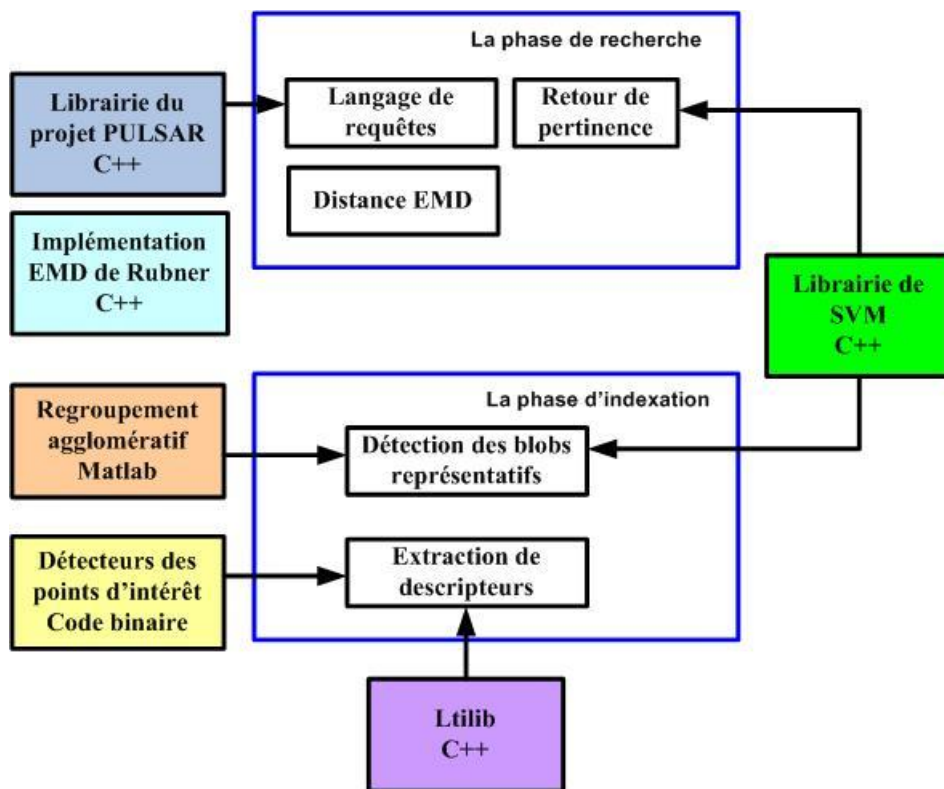


FIG. B.1 – Relation entre les 5 modules de notre prototype et les logiciels utilisés. Les 5 modules sont implémentés dans 3 bibliothèques (libdescriptors, libutilities, libqlanguage).

données et des algorithmes utilisés en traitement d'images et vision par ordinateur. Nous avons utilisé les fonctions afin de lire et de stocker des images.

B.1.2 Détecteurs des points d'intérêt

Afin de détecter les points d'intérêt, nous avons employé les implémentations fournies par leurs auteurs (<http://www.robots.ox.ac.uk/vgg/research/affine/detectors.html>).

B.1.3 Regroupement agglomératif

Nous avons utilisé la fonction de regroupement agglomératif de Matlab.

B.1.4 Librairie du projet PULSAR

Notre implémentation hérite une librairie développée par le projet PULSAR [Vu 2004].

B.1.5 Librairie de SVM

Nous avons utilisé la librairie de SVM fournie par Chih-Chung Chang et Chih-Jen Lin (<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>).

B.2 Phase d'indexation

B.2.1 Extraction de descripteurs

Nous avons implémenté en C++ plusieurs algorithmes de détection des couleurs dominantes, des histogrammes de contours et des matrices de covariance qui sont présentés dans le chapitre 4. Pour les matrices de covariance, nous avons implémenté l'algorithme rapide proposé par Porikli et al. [Porikli 2006a]. Nous avons également implémenté la distance correspondant à chaque type de descripteur.

B.2.2 Détection des blobs représentatifs

Nous avons implémenté deux méthodes de détection des blobs représentatifs proposées dans cette thèse : l'une basée sur le changement d'apparence et l'autre basée sur le regroupement des blobs.

B.3 Phase de recherche

B.3.1 Langage de requêtes

Nous avons développé une librairie en C++ pour le langage de requêtes proposé. Cette librairie consiste en des fonctions pour préparer les éléments des requêtes, pour analyser syntaxiquement les requêtes et pour effectuer la recherche définie dans les requêtes.

B.3.2 Calcul de distance EMD

L'algorithme de calcul EMD proposé par Hillier et al. [Hillier 1990] a été implémenté par Rubner (<http://www.cs.duke.edu/~tomasi/software/emd.htm>). Nous avons modifié l'implémentation de Rubner afin de l'appliquer à la mise en correspondance entre objets avec différentes distances et différents descripteurs.

B.3.3 Retour de pertinence

Nous avons implémenté deux méthodes de retour de pertinence (retour de pertinence basé sur plusieurs images d'exemple et celui basé sur les SVM à une classe). Nous avons également développé un processus permettant d'effectuer automatiquement le retour de pertinence pour un ensemble de requêtes.

B.4 Algorithme de Calderara et al. et celui de Ma et al.

Afin de comparer notre approche avec celle de [Calderara 2006] et de Ma et al. [Ma 2007], nous avons réimplémenté les algorithmes proposés par les auteurs. Pour l'algorithme de Calderara et al., les gaussiennes sont créées et mises à jour. La mise en correspondance entre une image et un objet est développée selon la description de l'auteur. Pour l'algorithme de Ma et al., la mise en correspondance entre objets en se basant sur la distance de Hausdorff et sur les matrices de covariance est implémentée.

Bibliographie

- [Allen 1983] James F. Allen. *Maintaining knowledge about temporal intervals*. Commun. ACM, vol. 26, no. 11, pages 832–843, 1983. 113
- [Annesley 2005] J. Annesley, J. Orwell et J.-P. Renno. *Evaluation of MPEG7 color descriptors for visual surveillance retrieval*. In Proceedings of the 14th International Conference on Computer Communications and Networks (ICCCN'05), pages 105–112, Washington, DC, USA, 2005. IEEE Computer Society. 77, 172
- [Assfalg 2003] J. Assfalg, M. Bertini, C. Colombo, A. del Bimbo et W. Nunziati. *Semantic annotation of soccer videos : automatic highlights identification*. Computer Vision and Image Understanding (CVIU), vol. 92, no. 2-3, pages 285–305, November 2003. xii, 37
- [Avanzi 2001] Alberto Avanzi, François Bremond et Monique Thonnat. *Tracking Multiple Individuals for Video Communication*. In International Conference on Image Processing (ICIP'01), Thessaloniki, Greece, October 2001. 126
- [Avanzi 2005] A. Avanzi, F. Brémond, C. Tornieri et M. Thonnat. *Design and Assessment of an Intelligent Activity Monitoring Platform*. EURASIP Journal on Applied Signal Processing, special issue in "Advances in Intelligent Vision Systems : Methods and Applications", pages 2359–2374, 2005. xii, 39, 126
- [Ayache 2007] Stéphane Ayache, Georges Quénot et Jérôme Gensel. *Classifier Fusion for SVM-Based Multimedia Semantic Indexing*. In European Conference on Information Retrieval (ECIR'07), pages 494–504, 2007. 28
- [Basharat 2007] Arslan Basharat, Yun Zhai et Mubarak Shah. *Content based video matching using spatiotemporal volumes*. Computer. Vis. Image Understand (CVIU), Special Issue on Similarity Matching in Computer Vision and Multimedia, 2007. 29
- [Bashir 2007] Faisal I. Bashir, Ashfaq A. Khokhar et Dan Schonfeld. *Real-Time Motion Trajectory-Based Indexing and Retrieval of Video Sequences*. IEEE Transactions on Multimedia, vol. 9, no. 1, pages 58–65, 2007. 88
- [Berrani 2002] S.-A. Berrani, L. Amsaleg et P. Gros. *Recherche par similarité dans les bases de données multidimensionnelles : panorama des techniques d'indexation*. RSTI - Ingénierie des systèmes d'information. Bases de données et multimédia, ed. Hermes - Lavoisier, vol. 7, no. 5-6, pages 9–44, 2002. 25, 199
- [Calderara 2006] Simone Calderara, Rita Cucchiara et Andrea Prati. *Multimedia surveillance : content-based retrieval with multicamera people tracking*. In Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks (VSSN'06), pages 95–100, New York, NY, USA, 2006. ACM. xii, 44, 45, 46, 58, 59, 60, 125, 138, 153, 163, 193, 195, 198, 210

- [Carson 2002] C. Carson, S. Belongie, H. Greenspan et J. Malik. *Blobworld : image segmentation using expectation-maximization and its application to image querying*. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 24, no. 8, pages 1026–1038, 2002. 12
- [Chau 2009] D.P. Chau, F. Bremond, E. Corvee et M. Thonnat. *Repairing People Trajectories Based on Point Clustering*. In In the International Conference on Computer Vision Theory and Applications (VISAPP'09), Lisboa, Portugal, 2009. 202
- [Chen 2004] Lei Chen et M. Tamer Özsu. *Symbolic Representation and Retrieval of Moving Object Trajectories*. In Multimedia Information Retrieval (MIR'04), New York, USA, 2004. xiv, 88, 90, 92, 110, 112
- [Chen 2005] Lei Chen et M. Tamer Özsu. *Robust and Fast Similarity Search for Moving Object Trajectories*. In SIGMOD, Baltimore, Maryland, USA., 2005. 88
- [Chen 2006] Xin Chen et Chengcui Zhang. *An Interactive Semantic Video Mining and Retrieval Platform—Application in Transportation Surveillance Video for Incident Detection*. In Sixth International Conference on Data Mining (ICDM'06), pages 129–138, Dec 2006. xiii, 54, 56, 58, 59, 60
- [Chen 2007] Xin Chen et Chengcui Zhang. *Interactive mining and semantic retrieval of videos*. In Proceedings of the 8th international workshop on Multimedia data mining (MDM'07), pages 1–9, New York, NY, USA, 2007. ACM. 54, 56, 58, 59, 60, 120
- [Chetverikov 2003] Dmitry Chetverikov. *A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves*. In Computer Analysis of Images and Patterns, pages 746–753, 2003. 88
- [Cohen 2008] Isaac Cohen, Yunqian Ma et Ben Miller. *Associating Moving Objects Across Non-overlapping Cameras : A Query-by-Example Approach*. In IEEE Conference on Technologies for Homeland Security (HST'08), pages 566–571, Waltham, MA, USA, 2008. 42, 58, 60, 145, 155
- [Comaniciu 2002] D. Comaniciu et P. Meer. *Mean Shift : A Robust Approach Toward Feature Space Analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 24, pages 603–619, 2002. 31
- [Conaire 2006] Ciarán O. Conaire, Noel E. O'Connor, Eddie Cooke et Alan F. Smeaton. *Multispectral Object Segmentation and Retrieval in Surveillance Video*. In International Conference on Image Processing (ICIP'06), pages 2381–2384, 2006. 39, 58, 59
- [Corridoni 1998] J. M. Corridoni et A. Del Bimbo. *Structured representation and automatic indexing of movie information content*. Pattern Recognition (PR), vol. 31, no. 12, pages 2027–2045, 1998. 26
- [Cotsaces 2006] C. Cotsaces, N. Nikolaidis et I. Pitas. *Shot detection and condensed representation—a review*. IEEE Signal Processing Magazine, vol. 23, no. 2, pages 28–37, June 2006. 27

- [Cristani 2007] Marco Cristani, Manuele Bicego et Vittorio Murino. *Audio-Visual Event Recognition in Surveillance Video Sequences*. IEEE Transactions on Multimedia, vol. 9, no. 2, pages 257–267, 2007. 203
- [Crucianu 2004] Michel Crucianu, Marin Ferecatu et Nozha Boujemaa. *Relevance feedback for image retrieval : a short review*. In In State of the Art in Audiovisual Content-Based Retrieval, Information Universal Access and Interaction including Datamodels and Languages (DELOS2 Report, 2004. vii, 21, 24
- [Cupillard 2002] Frédéric Cupillard, Francois Brémond et Monique Thonnat. *Tracking group of people for video surveillance*. In IEEE Proc. 2nd European Workshop on Advanced Video-Based Surveillance System (AVBS'02), 2002. 126
- [Cutler 2002] Ross Cutler, Yong Rui, Anoop Gupta, JJ Cadiz, Ivan Tashev, Li wei He, Alex Colburn, Zhengyou Zhang, Zicheng Liu et Steve Silverberg. *Distributed meetings : a meeting capture and broadcasting system*. In Proceedings of the tenth ACM international conference on Multimedia (MULTIMEDIA'02), pages 503–512, New York, NY, USA, 2002. ACM. 37
- [Dalal 2005] Navneet Dalal et Bill Triggs. *Histograms of Oriented Gradients for Human Detection*. In Cordelia Schmid, Stefano Soatto et Carlo Tomasi, éditeurs, International Conference on Computer Vision & Pattern Recognition (CVPR'05), volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005. 198
- [Datta 2008] Ritendra Datta, Dhiraj Joshi, Jia Li et James Z. Wang. *Image Retrieval : Ideas, Influences, and Trends of the New Age*. ACM Computing Surveys, vol. 40, no. 2, pages 1–60, 2008. 1, 11
- [Deng 1999] Yining Deng, C. Kenney, M. S. Moore et B. S. Manjunath. *Peer group filtering and perceptual color image quantization*. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'99), volume 4, pages 21–24, Orlando, FL, USA, 1999. ACM. 77
- [Deng 2001] Y. Deng, B. S. Manjunath, C. Kenney, M. S. Moore et H. Shin. *An efficient color representation for image retrieval*. IEEE Trans. Image Processing, vol. 10, pages 140–147, 2001. 77, 103
- [Ekin 2003a] A. Ekin, A.M. Tekalp et R. Mehrotra. *Automatic soccer video analysis and summarization*. Image Processing, vol. 12, no. 7, pages 796–807, July 2003. xi, xii, 35, 36
- [Ekin 2003b] A. Ekin et M. Tekalp. *Generic play-break event detection for summarization and hierarchical sports video analysis*. In Proceedings of the 2003 International Conference on Multimedia and Expo (ICME'03), pages 169–172, Washington, DC, USA, 2003. IEEE Computer Society. xi, 34, 35
- [Fauqueur 2006] Julien Fauqueur et Nozha Boujemaa. *Mental image search by boolean composition of region categories*. Multimedia Tools and Applications, vol. 31, no. 1, pages 95–117, 2006. 13

- [Foresti 2002] Gian Luca Foresti, Lucio Marcenaro et Carlo S. Regazzoni. *Automatic detection and indexing of video-event shots for surveillance applications*. IEEE Transactions on Multimedia, vol. 4, no. 4, pages 459–471, 2002. xiii, 49, 51, 52, 53, 58, 88
- [Forstner 1999] W. Forstner et B. Moonen. *A metric for covariance matrices*. In Tech. Rep., Dept. Geodesy Geoinform., Stuttgart Univ., Stuttgart, Germany, 1999. 104
- [Fusier 2007] F. Fusier, V Valentin, F. Brémond, M. Thonnat, M. Borg, D. Thirde et J. Ferryman. *Video Understanding for Complex Activity Recognition*. Machine Vision and Applications Journal (MVA), vol. 18, pages 167–188, 2007. 126
- [Gandhi 2007] Tarak Gandhi et Mohan Manubhai Trivedi. *Person tracking and reidentification : Introducing Panoramic Appearance Map (PAM) for feature representation*. Journal Machine Vision and Applications (MVA), Special Issue Paper, vol. 18, no. 3-4, pages 207–220, 2007. 201
- [Ghanem 2004] Nagia Ghanem, Daniel DeMenthon, David Doermann et Larry Davis. *Representation and Recognition of Events in Surveillance Video Using Petri Nets*. In 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04), volume 7, page 112, 2004. xii, xiv, xv, 46, 47, 58, 60, 67, 101, 114, 115, 117, 118, 119, 180
- [Gheissari 2006] Niloofar Gheissari, Thomas B. Sebastian et Richard Hartley. *Person Reidentification Using Spatiotemporal Appearance*. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), pages 1528–1535, Washington, DC, USA, 2006. IEEE Computer Society. 201
- [Goldmann 2006] Lutz Goldmann, Lars Thiele et Thomas Sikora. *Online Image Retrieval System Using Long Term Relevance Feedback*. In International Conference on Image and Video Retrieval (CIVR'06), volume LNCS 4071, pages 422–431, 2006. 24
- [Greenhill 2002] D. Greenhill, P. Remagnino et G.A. Jones. Video based surveillance systems - computer vision and distributed processing, chapitre VIGILANT : Content-Querying of Video Surveillance Streams, pages 193–204. ISBN/ISSN 0-7923-7632-3. Kluwer Academic Publishers, 2002. xiii, 51, 53, 58, 60
- [Haidar 2005] Siba Haidar, Philippe Joly et Bilal Chebaro. *Style Similarity Measure for Video Documents Comparison*. In 4th Int. Conf. on Image and Video Retrieval (CIVR'05), volume LNCS 3568/2005, pages 307–317, 2005. 38
- [Hampapur 2007] Arun Brown Hampapur, Lisa Feris, Rogerio Senior, Andrew Chiao-Fe Shu, Yingli Tian, Yun Zhai et Lu Max. *Searching surveillance video*. In IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'07), pages 75–80, 5-7 Sept 2007. 51, 58
- [Hauptmann 2008] A. G. Hauptmann, M. G. Christel et R. Yan. *Video Retrieval Based on Semantic Concepts*. Proceedings of the IEEE, Special Issue on

- Advances in Multimedia Information Retrieval, vol. 96, no. 4, pages 602–622, April 2008. 28
- [Hillier 1990] F. S. Hillier et G. J. Liebeman. Introduction to mathematical programming. McGraw-Hill, New York, NY, 1990. 107, 210
- [Hsieh 2006] Jun-Wei Hsieh, Shang-Li Yu et Yung-Sheng Chen. *Motion-Based Video Retrieval by Trajectory Matching*. IEEE Transaction on Circuits and Systems for Video Technology, vol. 16, no. 3, 2006. xiv, 88, 90
- [Hu 2007] Weiming Hu, Dan Xie, Zhouyu Fu, Wenrong Zeng et Steve Maybank. *Semantic-Based Surveillance Video Retrieval*. IEEE Transactions on Image Processing, vol. 16, no. 4, pages 1168–1181, April 2007. vii, xii, 46, 48, 49, 58
- [Jaffré 2004] Gaël Jaffré et Philippe Joly. *Costume : A New Feature for Automatic Video Content Indexing*. In Recherche d'Information Assistée par Ordinateur (RIAO'04), pages 314–325, Avignon, France, 2004. xi, 29, 31
- [Jaffré 2005] Gaël Jaffré et Philippe Joly. *Improvement of a Person Labelling Method Using Extracted Knowledge on Costume*. In In Proceedings of the 11th International Conference on Computer Analysis of Images and Patterns (CAIP'05), Rocquencourt, France, 2005. 29, 31
- [Jing 2003] Feng Jing, Mingjing Li, Lei Zhang, Hong-Jiang Zhang et Bo Zhang. *Learning in Region-Based Image Retrieval*. In International Conference on Image and Video Retrieval (CIVR'03), volume 2728/2003, pages 199–204, 2003. 25
- [Jing 2004] Feng Jing, Mingjing Li, HongJiang Zhang et Bo Zhang. *Relevance feedback in region-based image retrieval*. IEEE Trans. Circuits Syst. Video Techn., vol. 14, no. 5, pages 672–681, 2004. 25
- [Kent 1955] Allen Kent, Madeline M. Berry, Fred U. Luerhs, J. R. Perry et J. W. Perry. *Machine literature searching VIII : Operational criteria for designing information retrieval systems*. American Documentation, vol. 6, no. 2, pages 93–101, 1955. 136
- [Kijak 2003] Ewa Kijak. *Structuration multimodale des vidéos de sport par modèles stochastiques*. Thèse de doctorat, Université de Rennes 1, 2003. 34
- [Koprinska 2001] I. Koprinska et S. Carrato. *Temporal video segmentation : a survey*. Signal Processing : Image Communication, vol. 16, pages 477–500, June 2001. 27
- [Laaksonen 1999] J. Laaksonen, M. Koskela et E. Oja. *PicSOM : self-organizing maps for content-based image retrieval*. International Joint Conference on Neural Networks (IJCNN'99), vol. 4, pages 2470–2473, 1999. 69
- [Le 2006] D.D. Le, S. Satoh et M.E. Houle. *Face Retrieval in Broadcasting News Video by Fusing Temporal and Intensity Information*. In Int. Conf. on Image and Video Retrieval (CIVR'06), pages 391–400, 2006. 38

- [Lew 2006] Micheal S. Lew, Nicu Sebe et Ramesh Jain. *Content-Based Multimedia Information Retrieval : State of the Art and Challenges*. ACM Transactions on Multimedia Computing, Communications and Applications, vol. 2, no. 1, pages 1–19, February 2006. 1
- [Lien 2007] C.C. Lien, C.L. Chiang et C.H. Lee. *Scene-based event detection for baseball videos*. Journal of Visual Communication and Image Representation, vol. 18, no. 1, pages 1–14, February 2007. 34
- [Lim 2003a] Joo-Hwee Lim, P. Mulhem et Qi Tian. *Event-based home photo retrieval*. In International Conference on Multimedia and Expo (ICME'03), volume 2, 6-9 July 2003. 17, 18
- [Lim 2003b] Joo-Hwee Lim, Qi Tian et Philippe Mulhem. *Home Photo Content Modeling for Personalized Event-Based Retrieval*. IEEE MultiMedia, vol. 10, no. 4, pages 28–37, 2003. xi, 17, 18, 19, 20
- [Linde 1980] Y. Linde, A. Buzo et R. M. Gray. *An algorithm for vector quantizer design*. IEEE Trans. Commun., vol. COM-28, pages 84–95, 1980. 77
- [Lowe 2004] D. Lowe. *Distinctive image features from scale invariant keypoints*. In International Journal Computer Vision (IJCV), vol. 60, no. 2, pages 91–110, 2004. xiii, xiv, 86, 87, 105
- [Ma 2007] Yunqian Ma, Ben Miller et Isaac Cohen. *Video Sequence Querying Using Clustering of Objects' Appearance Models*. In International Symposium on Visual Computing (ISVC'07), pages 328–339, 2007. 42, 58, 59, 60, 66, 78, 95, 99, 105, 125, 138, 144, 145, 148, 153, 155, 192, 193, 195, 198, 210
- [Maillot 2005] Nicolas Maillot et Monique Thonnat. *A Weakly Supervised Approach for Semantic Image Indexing and Retrieval*. In International Conference on Image and Video Retrieval (CIVR'05), volume 3568/2005, pages 629–638, 19-22 Aug 2005. 17, 18
- [Manjunath 2001] B. S. Manjunath, J. R. Ohm, V. V. Vinod, et A. Yamada. *Color and Texture descriptors*. IEEE Trans. Circuits and Systems for Video Technology, Special Issue on MPEG-7, vol. 11, no. 6, pages 703–715, Jun 2001. 77
- [Matas 2002] J. Matas, O. Chum, M. Urban et T. Pajdla. *Robust wide baseline stereo from maximally stable extremal regions*. In British Machine Vision Conference (BMVC'02), pages 384–393, 2002. 84
- [Meessen 2006] Jérôme Meessen, Matthieu Coulanges, Xavier Desurmont et Jean-Francois Delaigle. *Content-Based Retrieval of Video Surveillance Scenes*. In Multimedia Content Representation, Classification and Security, pages 785–792. Springer Berlin / Heidelberg, 2006. 40, 58
- [Meessen 2007] J. Meessen, X. Desurmont, J.F. Delaigle, C. De Vleeschouwer et B. Macq. *Progressive Learning for Interactive Surveillance Scenes Retrieval*. In IEEE International Workshop on Visual Surveillance (VS'07), pages 1–8, 2007. 54, 58, 59, 60, 120

- [Mezaris 2004] Vasileios Mezaris, Ioannis Kompatsiaris et Michael G. Strintzis. *Region-Based Image Retrieval Using an Object Ontology and Relevance Feedback*. EURASIP Journal on Applied Signal Processing, vol. 6, pages 886–901, 2004. xi, 17, 18
- [Mikolajczyk 2001] K. Mikolajczyk et C. Schmid. *Indexing based on scale invariant interest points*. In Proceedings. Eighth IEEE International Conference on Computer Vision (ICCV'01), volume 1, pages 525–531, Vancouver, BC, Canada, 2001. 83
- [Mikolajczyk 2004] Tuytelaars T. Schmid C. Zisserman A. Matas J. Schaffalitzky F. Kadir T. Mikolajczyk K. et L.V. Gool. *A comparison of affine region detectors*. IJCV, vol. 65, no. 1/2, pages 43–72, 2004. 82, 174
- [Mikolajczyk 2005] K. Mikolajczyk et C. Schmid. *A performance evaluation of local descriptors*. PAMI, vol. 27, no. 10, pages 1615–1630, 2005. 86
- [Müller 2002] H. Müller, S. Marchand-Maillet et T. Pun. *The truth about corel - evaluation in image retrieval*. In Proc. Of. International Conference in Image and Video Retrieval (CIVR'02), pages 28–49, London, United Kingdom, July 2002. 136
- [Naftel 2006] Andrew Naftel et Shehzad Khalid. *Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space*. Multimedia Systems, vol. 16, no. 12, pages 227–238, 2006. 88
- [Naphade 2004] Milind R. Naphade et John R. Smith. *On the detection of semantic concepts at TRECVID*. In Proceedings of the 12th annual ACM international conference on Multimedia (MULTIMEDIA'04), pages 660–667, New York, NY, USA, 2004. ACM. 28
- [Nascimento 2006] J.C Nascimento et J. S. Marques. *Performance evaluation of object detection algorithms for video surveillance*. IEEE Transactions on Multimedia, vol. 8, no. 4, pages 761–774, 2006. 5
- [Nghiem 2007] Anh-Tuan Nghiem, Francois Bremond, Monique Thonnat et Valery Valentin. *ETISEO, performance evaluation for video surveillance systems*. In Proceedings of International Conference on Advanced Video and Signal Based Surveillance (AVSS'07), London, United Kingdom, September 2007. 5, 128
- [Park 2000] Dong Kwon Park, Yoon Seok Jeon et Chee Sun Won. *Efficient use of local edge histogram descriptor*. In Proceedings of the 2000 ACM workshops on Multimedia (MULTIMEDIA'00), pages 51–54, New York, NY, USA, 2000. ACM. xiii, 79, 80, 170
- [Patino 2008] J.L. Patino, H. Benhadda, E. Corvee, F. Brémont et M. Thonnat. *Extraction of activity patterns on large video recordings*. In Computer Vision, IET, vol. 2, pages 108–128, 2008. 129
- [Peng 2005] Yuxin Peng et Chong-Wah Ngo. *EMD-Based Video Clip Retrieval by Many-to-Many Matching*. In International Conference on Image and Video Retrieval (CIVR'05), pages 71–81, 2005. 106

- [Peng 2007] Yuxin Peng, Chong-Wah Ngo et Jianguo Xiao. *OM-based video shot retrieval by one-to-one matching*. *Multimedia Tools Appl.*, vol. 34, no. 2, pages 249–266, 2007. 29
- [Pentland 1994] A. Pentland, R. Picard et S. Sclaroff. *Photobook : Content-based manipulation of image databases*. *Proc. SPIE Storage and Retrieval for Image and Video Databases II*, pages 34–47, 1994. 11
- [Pickering 2003] M. J. Pickering et Ruger S. *Evaluation of key-frame based retrieval techniques for video*. *Comput. Vision Image Understand (CVIU)*, vol. 92, no. 2, pages 217–235, 2003. 27
- [Porikli 2006a] F. Porikli et O. Tuzel. *Fast construction of covariance matrices for arbitrary size image windows*. In *IEEE International Conference on Image Processing (ICIP'06)*, Atlanta, 2006. 148, 174, 200, 209
- [Porikli 2006b] Tuzel O. Meer P. Porikli F. *Covariance Tracking using Model Based on Means on Riemannian Manifolds*. In *In : Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006. 148, 200
- [Rowe 2005] Lawrence A. Rowe et Ramesh Jain. *ACM SIGMM retreat report on future directions in multimedia research*. *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 1, no. 1, pages 3–13, 2005. 1
- [Rubner 1998] Y. Rubner, C. Tomasi et L.J. Guibas. *A Metric for Distributions with Applications to Image Databases*. In *International Conference on Computer Vision (ICCV'98)*, pages 59–66, 1998. 12, 13, 68, 106
- [Rubner 2000] Yossi Rubner, Carlo Tomasi et Leonidas J. Guibas. *The Earth Mover's Distance as a Metric for Image Retrieval*. *Journal International Journal of Computer Vision (IJCV)*, vol. 40 (2), pages 99–121, 2000. 107, 163
- [Rui 1998] Yong Rui, Thomas S. Huang, Michael Ortega et Sharad Mehrotra. *Relevance Feedback : A Power Tool in Interactive Content-Based Image Retrieval*. *IEEE Trans. on Circuits and Systems for Video Technology*, Special Issue on Segmentation, Description, and Retrieval of Video Content, vol. 8, no. 5, pages 644–655, 1998. 24, 69
- [Rui 2001] Y. Rui et T. S. Huang. *Principles of visual information retrieval*, chapitre Relevance feedback techniques in image retrieval, pages 219–258. Springer-Verlag, 2001. 24, 54, 69
- [Schölkopf 1999] B. Schölkopf et J.C. Platt. *Estimating the support of a high-dimensional distribution*. In *Microsoft Research Corporation Technical Report MSR-TR-99-87*, 1999. 122
- [Sebe 2003] N. Sebe, M.S. Lew, X. Zhou, T.S. Huang et E. Bakker. *The State of the Art in Image and Video Retrieval*. In *International Conference on Image and Video Retrieval (CIVR'03)*, pages 1–8, Urbana, USA, Jul 2003. 1
- [Sivic 2006] J. Sivic et A. Zisserman. *Video Google : Efficient Visual Search of Videos*. *Toward Category-Level Object Recognition*, vol. LNCS 4170, pages 127–144, 2006. 31

- [Sivic 2008] J. Sivic et A. Zisserman. *Efficient Visual Search for Objects in Videos*. Proceedings of the IEEE, Special Issue on Advances in Multimedia Information Retrieval, vol. 96, no. 4, pages 548–566, April 2008. 14, 31
- [Smeulders 2000] A. Smeulders, M. Worring, S. Santini, A. Gupta et R. Jain. *Content based image retrieval at the end of the early years*. IEEE Trans. Patt. Analy. Mach. Intell., vol. 22, no. 12, pages 1349–1380, 2000. 1, 5
- [Snoek 2005] Cees G. M. Snoek, Marcel Worring et Arnold W. M. Smeulders. *Early versus late fusion in semantic video analysis*. In Proceedings of the 13th annual ACM international conference on Multimedia (MULTIMEDIA'05), pages 399–402, New York, NY, USA, 2005. ACM. 203
- [Snoek 2007a] Cees G. M. Snoek, Bouke Huurnink, Laura Hollink, Maarten de Rijke, Guus Schreiber et Marcel Worring. *Adding Semantics to Detectors for Video Retrieval*. IEEE Transactions on Multimedia, vol. 9, no. 5, pages 975–986, August 2007. 28
- [Snoek 2007b] Cees G. M. Snoek, Marcel Worring, Dennis C. Koelma et Arnold W. M. Smeulders. *A Learned Lexicon-Driven Paradigm for Interactive Video Retrieval*. IEEE Transactions on Multimedia, vol. 9, no. 2, pages 280–292, February 2007. 28
- [Souvannavong 2005] Fabrice Souvannavong, Bernard Mérialdo et Benoit Huet. *Multi-modal classifier fusion for video shot content retrieval*. In 6th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'05), Montreux, Switzerland, April 13-15 2005. 28
- [Stauffer 1999] C. Stauffer et W.E.L. Grimson. *Adaptive background mixture models for real-time tracking*. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99), volume 2, pages 250–252, 1999. 163
- [Stringa 1998] E. Stringa et C.S. Regazzoni. *Content-based retrieval and real time detection from videosequences acquired by surveillance systems*. In International Conference on Image Processing (ICIP'98), volume 3, pages 138–142, 4-7 Oct 1998. xii, 49, 50, 58
- [Stringa 2000] E. Stringa et C.S. Regazzoni. *Real-time video-shot detection for scene surveillance applications*. IEEE Transactions on Image Processing, vol. 9, no. 1, pages 69–79, Jan 2000. xii, 49, 50, 58
- [Tian 2008] Ying-Li Tian, Lisa Brown, Arun Hampapur, Max Lu, Andrew Senior et Chiao fe Shu. *IBM smart surveillance system (S3) : event based video surveillance system with an open and extensible framework*. Journal Machine Vision and Applications (MVA), 2008. xiv, 51, 58, 114, 115, 180
- [Tirilly 2008] Pierre Tirilly, Vincent Claveau et Patrick Gros. *Language modeling for bag-of-visual words image categorization*. In Proceedings of the international conference on Content-based image and video retrieval (CIVR'08), pages 249–258, New York, NY, USA, 2008. ACM. xi, 15, 16

- [Trichet 2008] Rémi Trichet et Bernard Mérialdo. *Keypoints labeling for background subtraction in tracking applications*. In International Conference on Multimedia & Expo (ICME'08), Hannover, Germany, Jun 23-26 2008. 76, 200
- [Veltkamp 2000] Remco C. Veltkamp et Mirela Tanase. *Content-Based Image Retrieval Systems : A Survey*. Technical Report UU-CS-2000-34, 2000. 11
- [Viola 2004] Paul Viola et Michael J. Jones. *Robust Real-Time Face Detection*. Int. J. Comput. Vision (IJCV), vol. 57, no. 2, pages 137–154, 2004. 148
- [Vu 2003] V.T. Vu, F. Brémond et M. Thonnat. *Automatic video interpretation : a novel algorithm for temporal scenario recognition*. In Proc. 18th Int. Joint Conf. Artificial Intelligence. (IJCAI'03), pages 1295–1302, 2003. 127
- [Vu 2004] Van-Thinh Vu. *Temporal Scenario for Automatic Video Interpretation*. Thèse de doctorat, Université de Nice Sophia Antipolis, October 2004. 71, 209
- [Wang 2001] J. Z. Wang, J. Li et G. Wiederhold. *SIMPLicity : Semantics-sensitive integrated matching for picture libraries*. IEEE Trans. Pattern Anal. Machine Intell., vol. 23, pages 947–963, 2001. 12, 13
- [Wayne 1993] Niblack Wayne, Barber Ron, Equitz Will, Myron D. Flickner, Eduardo H. Glasman, Dragutin Petkovic, Peter Yanker, C. Faloutsos et Gabriel Taubin. *The QBIC project : Querying images by content using color, texture, and shape*. Proc. SPIE, vol. 1908, no. 1, pages 173–187, 1993. 11
- [Won 2002] Chee Sun Won, Dong Kwon Park et S.-J. Park. *Efficient use of MPEG-7 edge histogram descriptor*. ETRI Journal, vol. 24, pages 23–30, 2002. xiii, 79, 80
- [Won 2004] Chee Sun Won. *Feature Extraction and Evaluation Using Edge Histogram Descriptor in MPEG-7*. In Pacific-Rim Conference on Multimedia (PCM'04), pages 583–590, 2004. 170
- [Xiao 1993] WS Xiao. *Graph theory and its algorithms*. Aviation Industrial Press, Beijing, 1993. 30
- [Xie 2004a] D. Xie, Weiming Hu, Tieniu Tan et Junyi Peng. *Semantic-based traffic video retrieval using activity pattern analysis*. In International Conference on Image Processing (ICIP'04), volume 1, pages 693–696, 2004. 48
- [Xie 2004b] Lexing Xie, Peng Xu, Shih-Fu Chang, Ajay Divakaran et Huifang Sun. *Structure analysis of soccer video with domain knowledge and hidden Markov models*. Pattern Recogn. Lett., vol. 25, no. 7, pages 767–775, 2004. 58
- [Xiong 2006] Zhou X.S. Tian Q. Rui R. Xiong Z. et T.S. Huang. *Semantic Retrieval of Video [Review of research on video retrieval in meetings, movies and broadcast news, and sports]*. IEEE Processing Magazine, March 2006. xi, 25, 27, 33
- [Yin 2005] Peng-Yeng Yin, Kuang-Cheng Chang et Anlei Dong. *Integrating Relevance Feedback Techniques for Image Retrieval Using Reinforcement Learning*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 10, pages 1536–1551, 2005. xi, 23, 24

- [Yuk 2007] J.S.C. Yuk, K.Y.K. Wong, R.H.Y. Chung, K.P. Chow, F.Y.L. Chin et K.S.H. Tsang. *Object-Based Surveillance Video Retrieval System with Real-Time Indexing Methodology*. In International Conference on Image Analysis and Recognition (ICIAR'07), pages 626–637, 2007. 40, 58, 59
- [Zhai 2006] Yun Zhai, Jingen Liu et Mubarak Shah. *Automatic Query Expansion in News Video Retrieval*. In IEEE International Conference on Multimedia and Expo (ICME'06), Toronto, Canada, 2006. xi, 29, 30
- [Zouba 2008] Nadia Zouba, François Bremond et Monique Thonnat. *Monitoring Activities of Daily Living (ADLs) of Elderly Based on 3D Key Human Postures*. In In the 4th International Cognitive Vision Workshop (ICVW'08), Santorini, Greece, May 12-15 2008. 203
- [Zúniga 2006] Marcos Zúniga, François Bremond et Monique Thonnat. *Fast and reliable object classification in video based on a 3D generic model*. In Proc. Of. 3rd International Conference on Visual Information Engineering (VIE'06), Bangalore, India, September 26-28 2006. 126

Référence de l'auteur

Revue scientifique

- Le Thi Lan, Monique Thonnat, Alain Boucher, Francois Bremond. *Surveillance video indexing and retrieval using objet features and semantic events*, International Journal of Pattern Recognition and Artificial Intelligence, special issue on Visual Analysis and Understanding for Surveillance Applications (à paraître).

Congrès internationaux

- Le Thi Lan, Alain Boucher, Monique Thonnat, Francois Bremond. *A framework for surveillance video indexing and retrieval*, The 6th International Workshop on Content Based Multimedia Indexing(CBMI'08), pages 338-345, London, UK, June 18-20, 2008.
- Le Thi Lan, Monique Thonnat, Alain Boucher, Francois Bremond. *A Query Language Combining Object Features and Semantic Events for Surveillance Video Retrieval*, The International 14th MultiMedia Modeling Conference (MMM'08), pages 307-317, Kyoto, Japan, January 9-12 2008.
- Le Thi Lan, Alain Boucher, Monique Thonnat. *Subtrajectory-Based Video Indexing and Retrieval*, The International MultiMedia Modeling Conference (MMM'07), pages 418-427, Singapore, January 9-12.
- Le Thi Lan, Alain Boucher, Monique Thonnat. *Trajectory-Based Video Indexing and Retrieval Enabling Relevance Feedback*, First International Conference on Communications and Electronics (ICCE'06), Hanoi, Vietnam, October 10-11.

Congrès nationaux

- Le Thi Lan, Alain Boucher, Monique Thonnat. *Une interface de visualisation avec retour de pertinence pour la recherche d'images*, Rencontres des Jeunes Chercheurs en Recherche d'Information (RJCRI'06), 15-17 mars 2006, Lyon (France).
- Le Thi Lan, Alain Boucher, Monique Thonnat. *An interface for image retrieval and its extension to video retrieval*. Third national symposium on Research, Development and Application of Information and Communication Technology (ICT.rda), October 2006, Hanoi (Vietnam).

