N° d'ordre : 2366
EDSPIC : 616

Université Blaise Pascal - Clermont II

École Doctorale
Sciences Pour L'Ingénieur De Clermont-Ferrand

Thèse

présentée par :

Siméon SCHWAB

pour obtenir le grade de

Docteur d'Université

Spécialité : Informatique

Suivi visuel multi-cibles par partitionnement de détections : application à la construction d'albums de visages.

Soutenue publiquement le 8 Juillet 2013 devant le jury :

Président du jury :	Frédéric LERASLE	*Professeur*	LAAS
Rapporteurs :	Vincent CHARVILLAT	*Professeur*	IRIT
	Séverine DUBUISSON	*Maître de Conférences HDR*	LIP6
Directeur :	Laurent TRASSOUDAINE	*Professeur*	Institut Pascal
Encadrants :	Thierry CHATEAU	*Professeur*	Institut Pascal
	Christophe BLANC	*Maître de Conférences*	Institut Pascal

Remerciements

Difficile en quelques mots de remercier tous ceux qui, de près ou de loin, m'ont aidé, soutenu et encadré. Ces remerciements sont loin d'être exhaustifs.

Je remercie tous les membres du jury d'avoir examiné ma thèse, pour leurs remarques et questions qui m'ont permis de clarifier certains aspects de mes travaux. Je remercie Christophe et Laurent pour la mise en place du projet, leur soutien et la direction de la thèse. Un grand merci à Thierry pour son encadrement, ses conseils, nos échanges enrichissants ainsi que pour sa bonne animation de l'équipe ComSee.

Je remercie aussi tous les collègues et amis de ComSee, qui donnent à l'équipe cette chaleureuse atmosphère. Merci pour leurs conseils théoriques, techniques et personnels. Je garde notamment un bon souvenir des discussions souvent hautement concrètes, débutant par des aspects historico-politico-psycho-sociologiques, pour s'évader sur des débats houleux concernant nos différentes interprétations des paradigmes du développement informatique en passant par des causerie concernant la température idéale (discussions bercées par le franc parlé de la poésie russe et les emphases empreintes d'exagérations provençales) avant de se perdre dans les hautes sphères obscures et brumeuses de la physique quantique et des mathématiques théoriques. En particulier je remercie Jean-Marc, Baptiste, Datta, Shuda, Pierre, Pierre, Clément, Vadim, Alexis, Sébastien, François, Gaspard, François, Laétitia, tous ceux qui ont initié ou rejoint la V-team et tous ceux qui participent de quelconque manière au bon fonctionnement du laboratoire et de ses locaux. Mes remerciements vont aussi aux employés de Vesalis qui m'ont aidé malgré les circonstances.

Je remercie tous les amis de l'EPE de Clermont-Ferrand, ainsi que ma famille et belle-famille pour leurs attentions et leur présence à la soutenance. Et pour finir, un immense merci à ma femme pour son aide, son encouragement dans les moments difficiles et ses nombreuses relectures, ainsi qu'à ma fille pour sa venue qui m'a redonné une bouffée d'oxygène par ses merveilleux sourires.

Résumé

Ce mémoire décrit mes travaux de thèse menés au sein de l'équipe ComSee (*Computers that See*) rattachée à l'axe ISPR (Image, Systèmes de Perception et Robotique) de l'Institut Pascal. Celle-ci a été financée par la société Vesalis par le biais d'une convention CIFRE avec l'Institut Pascal, subventionnée par l'ANRT (Association Nationale de la Recherche et de la Technologie). Les travaux de thèse s'inscrivent dans le cadre de l'automatisation de la fouille d'archives vidéo intervenant lors d'enquêtes policières.

L'application rattachée à cette thèse concerne la création automatique d'un album photo des individus apparaissant sur une séquence de vidéosurveillance. En s'appuyant sur un détecteur de visages, l'objectif est de regrouper par identité les visages détectés sur l'ensemble d'une séquence vidéo. Comme la reconnaissance faciale en environnement non-contrôlé reste difficilement exploitable, les travaux se sont orientés vers le suivi visuel multi-cibles global basé détections. Ce type de suivi est relativement récent. Il fait intervenir un détecteur d'objets et traite la vidéo dans son ensemble (en opposition au traitement séquentiel couramment utilisé). Cette problématique a été représentée par un modèle probabiliste de type Maximum *A Posteriori*. La recherche de ce maximum fait intervenir un algorithme de circulation de flot sur un graphe, issu de travaux antérieurs. Ceci permet l'obtention d'une solution optimale au problème (défini par l'*a posteriori*) du regroupement des détections pour le suivi.

L'accent a particulièrement été mis sur la représentation de la similarité entre les détections qui s'intègre dans le terme de vraisemblance du modèle. Plusieurs mesures de similarités s'appuyant sur différents indices (temps, position dans l'image, apparence et mouvement local) ont été testées. Une méthode originale d'estimation de ces similarités entre les visages détectés a été développée pour fusionner les différentes informations et s'adapter à la situation rencontrée.

Plusieurs expérimentations ont été menées sur des situations complexes, mais réalistes, de scènes de vidéosurveillance. Même si les qualités des albums construits ne satisfont pas encore à une utilisation pratique, le système de regroupement de détections mis en œuvre au cours de cette thèse donne déjà une première solution. Grâce au point de vue *partitionnement de données* adopté au cours de cette thèse, le suivi multi-cibles développé permet une extension simple à du suivi autre que celui des visages.

Mots-clés : suivi multi-cibles visuel, partitionnement de données, détecteur de visages, construction d'album photo, vidéosurveillance.

Abstract

This report describes my thesis work conducted within the ComSee (*Computers That See*) team related to the ISPR axis (ImageS, Perception Systems and Robotics) of *Institut Pascal*. It was financed by the Vesalis company via a CIFRE (Research Training in Industry Convention) agreement with *Institut Pascal* and publicly funded by ANRT (National Association of Research and Technology). The thesis was motivated by issues related to automation of video analysis encountered during police investigations.

The theoretical research carried out in this thesis is applied to the automatic creation of a photo album summarizing people appearing in a CCTV sequence. Using a face detector, the aim is to group by identity all the faces detected throughout the whole video sequence. As the use of facial recognition techniques in unconstrained environments remains unreliable, we have focused instead on global multi-target tracking based on detections. This type of tracking is relatively recent. It involves an object detector and global processing of the video (as opposed to sequential processing commonly used). This issue has been represented by a Maximum *A Posteriori* probabilistic model. To find an optimal solution of Maximum *A Posteriori* formulation, we use a graph-based network flow approach, built upon third-party research.

The study concentrates on the definition of inter-detections similarities related to the likelihood term of the model. Multiple similarity metrics based on different clues (time, position in the image, appearance and local movement) were tested. An original method to estimate these similarities was developed to merge these various clues and adjust to the encountered situation.

Several experiments were done on challenging but real-world situations which may be gathered from CCTVs. Although the quality of generated albums do not yet satisfy practical use, the detections clustering system developed in this thesis provides a good initial solution. Thanks to the *data clustering* point of view adopted in this thesis, the proposed detection-based multi-target tracking allows easy transfer to other tracking domains.

Keywords : visual multi-target tracking, data clustering, face detector, photo album generation, CCTV.

Table des matières

Table des figures

Notations :

$\doteq$	Symbole permettant d'introduire une définition ($a \doteq b$ signifie : a est défini par b).
$\propto$	Symbole signifiant *proportionnel à*.
$\|x\|$	Valeur absolue d'un réel x.
$\|X\|$	Nombre d'éléments d'un ensemble X.
$\boldsymbol{x}$	Les vecteurs sont représentés en gras.
$\boldsymbol{x}^\top$	Transposé du vecteur colonne $\boldsymbol{x}$.
$\|A\|$	Déterminant de la matrice A.
$A^\top$	Transposée de la matrice A.
$\binom{n}{k}$	Coefficient binomial (nombre de parties à k éléments d'un ensemble à n éléments).
$\left\{ {n \atop k} \right\}$	Nombre de Stirling de seconde espèce (nombre de partitions à k parties d'un ensemble à n éléments).
B_n	Nombre de Bell (nombre de partitionnements d'un ensemble à n éléments).
$G = (\mathcal{V}, \mathcal{A})$	Graphe orienté où $\mathcal{V}$ est l'ensemble des nœuds et $\mathcal{A}$ l'ensemble des arcs.
$G = (\mathcal{V}, \mathcal{E})$	Graphe non-orienté où $\mathcal{V}$ est l'ensemble des nœuds et $\mathcal{E}$ l'ensemble des arrêtes.
$P_{nom}(x)$	Densité de probabilité de la variable aléatoire x où nom permet de spécifier quelle loi est utilisée quand plusieurs sont envisagées pour représenter x.
$\mathcal{N}(\mu, \Sigma)$	Loi normale multivariée de moyenne μ et de matrice de covariance Σ.
$\mathcal{N}_x(\mu, \Sigma)$	Valeur en $\boldsymbol{x}$ de la densité de probabilité correspondant à la loi normale multivariée de moyenne μ et de covariance Σ.
$D_H(\ ,\ )$	Distance de Hellinger entre deux lois de probabilités.
$\mathcal{M}$	Variété riemannienne des matrices de covariance.
$D_{\mathcal{M}}(x, y)$	Distance utilisée sur la variété riemannienne des matrices de covariance : norme euclidienne du projeté (par l'application exponentielle) de y sur le plan tangent à $\mathcal{M}$ en x.

Définitions :

$\mathcal{D} \doteq \{1, ..., D\}$	Suite ordonnée des détections obtenues sur la séquence vidéo considérée.
$D \doteq \lvert\mathcal{D}\rvert$	Nombre de détections.
$T \doteq \{T_1, ..., T_K, T_{FP}\}$	Partitionnement des D détections en trajectoires.
T_k	k-ième trajectoire de T.
T_k^i	i-ième élément de la trajectoire T_k.
T_{FP}	Élément de T contenant les détections considérées comme fausses positives.
$\mathcal{T}$	Ensemble de tous les ensembles de trajectoires possibles (pour un ensemble de détections et de contraintes données).
$Z \doteq \{z_i\}_{i \in \mathcal{D}}$	Ensemble des observations liées aux détections.
z_i	Observation correspondant à la détection i, elle est constituée d'un vecteur de caractéristiques (ex : $z_i \doteq (\boldsymbol{x}_i, \boldsymbol{s}_i, \boldsymbol{a}_i, \boldsymbol{of}_i)$).
P_{start}	Probabilité paramétrique *a priori* sur le nombre de trajectoires.
P_{struct}	Probabilité sans paramètres de l'*a priori* sur le nombre de trajectoires.
P^s_{struct}	Probabilité *a priori* sur le nombre de trajectoires, approchée par le nombre de Stirling de seconde espèce.
P_{fp}	Probabilité *a priori* sur le nombre de faux positifs.
P_f	Probabilité qu'une détection ne soit pas un faux positif.
P_{traj}	Vraisemblance d'une trajectoire.
P_{link}	Vraisemblance du lien (transition) entre deux détections.
p_e	Probabilité de regroupement (intervient dans l'*a priori* P_{start}), régit la quantité de trajectoires.
β	Probabilité de faux positifs (intervient dans l'*a priori* P_{fp}), régit la quantité de faux positifs.
t_i	Temps vidéo d'apparition de la détection i.
$\boldsymbol{x}_i$	Position dans l'image (en pixels) du centre de la détection i.
w_i et h_i	Largeur et hauteur (en pixels) de la détection i.
$\boldsymbol{a}_i$	Descripteur d'apparence de la détection i.
$s_x(\ ,\ )$	Fonction de similarité x de $\mathcal{D} \times \mathcal{D} \rightarrow [0, 1]$. Par convention : $s(i, j) = 0$ si $t_i \geq t_j$.
$d_x(\ ,\ )$	Fonction de dissimilarité x de $\mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}^+$.
k-CC	Algorithme donnant l'ensemble de trajectoires optimal étant donné une fonction de dissimilarité et un nombre de trajectoires fixé.
EP	Pureté d'un partitionnement estimé par rapport à la vérité-terrain.
GTP	Pureté de la vérité-terrain au vu de l'estimation.
F-pureté	F-mesure entre les puretés.

Chapitre 1.

Introduction

L'information visuelle est très riche et fait partie des informations les plus utilisées par l'humain pour percevoir son environnement. La caméra tend aussi à être un capteur de plus en plus employé, elle permet d'obtenir de nombreuses informations. Toutefois, et cela se voit très nettement dans le domaine de la vision par ordinateur, il est très difficile d'extraire du sens des images issues des caméras. La perception visuelle humaine fait intervenir tout un mécanisme d'interprétation, basé entre autres sur l'apprentissage, qui lui permet de comprendre la scène observée. Ainsi, des tâches paraissant évidentes pour l'homme (détecter ou reconnaître des visages, suivre des objets dans des vidéos) s'avèrent être des tâches très difficiles à automatiser par un ordinateur, notamment parce que les algorithmes développés parviennent difficilement à interpréter conjointement tous les aspects de la scène.

Pour qu'un algorithme puisse être aussi efficace que la vision humaine (notamment dans le cadre de la vidéosurveillance) il lui faudrait un apprentissage qui lui permette d'analyser tous les indices de la scène (estimation 3D de l'environnement, éclairage et ombrages) et de se spécialiser dans la reconnaissance des activités et des identités des individus (en analysant les mouvements, les interactions, le visage, la silhouette et les vêtements).

Ainsi, même si l'humain n'a pas de difficulté à détecter des visages sur une image, ou à suivre des personnes dans une vidéo, l'automatisation de ces tâches par un ordinateur reste encore un réel défi.

Les travaux de recherche présentés dans ce rapport traitent du problème de suivi des visages dans le but de pouvoir automatiquement résumer des séquences issues de la vidéosurveillance. Le contexte applicatif, la problématique détaillée et les contributions apportées seront explicités dans les sections suivantes.

1.1. Contexte de la thèse

Les travaux de cette thèse ont été effectués dans le cadre d'une convention CIFRE (Conventions Industrielles de Formation par la REcherche) faisant intervenir la société Vesalis et l'Institut Pascal.

La thèse a été menée au sein de l'équipe de vision par ordinateur ComSee de l'axe thématique ISPR (Image, Systèmes de Perception, Robotique) de l'Institut Pascal. L'équipe travaille principalement sur des problématiques de localisation et reconstruction 3D, de calibration de capteurs et de suivi visuel.

Vesalis est une société qui travaille dans différents domaines liés à la vision par ordinateur et plus particulièrement sur les images de visages. Après avoir développé des technologies de traitement automatique du visage pour une application de simulation de maquillage, elle s'oriente aussi sur des problématiques de vidéosurveillance. Elle a initié le projet BioRafale

financé par OSEO. Ce projet vise à sécuriser les stades par la détection d'individus dont l'accès n'est pas autorisé. Le sujet de cette thèse n'est pas rattaché directement à l'identification des interdits de stade, mais à un besoin particulier de la police judiciaire relevé au cours de la mise en place du projet. Lors d'enquêtes, quand un suspect est recherché, la fouille de vidéos est longue et fastidieuse. La possibilité d'obtenir automatiquement un album photo des individus apparaissant sur la vidéo présenterait un gain de temps remarquable. Les travaux de thèse se sont inscrits dans cette problématique de construction d'album photo.

L'entreprise a aussi exprimé son souhait d'étendre l'étude à des vidéos du grand public (films, séries TV) ou des vidéos personnelles. Comme expliqué par la suite (2.1.1), la construction d'albums issus de vidéosurveillance et l'étiquetage d'acteurs sur des vidéos grand public ou amateurs, font intervenir des problématiques très différentes et les qualités des résultats ne sont pas transposables. L'étude s'est donc plus concentrée sur l'application à la vidéosurveillance et l'extension aux cas des vidéos grand public n'a finalement pas encore été expérimentée.

1.2. Besoins et contraintes de la vidéosurveillance

Face à la multiplication du nombre de caméras installées, la vidéosurveillance présente actuellement de nombreux besoins dans l'automatisation des tâches. La recherche dans le domaine de la vidéosurveillance intelligente suit deux grandes directions : la détection d'évènements à surveiller (évènements rares : mouvements brusques de foules, comportements violents, abandon d'objets ...) et l'identification d'individus.

La première direction a pour objectif d'apporter des outils aux centres de contrôles. Plusieurs centaines de caméras peuvent intervenir dans les systèmes de vidéosurveillance et, pour un instant donné, seule une petite fraction d'entre elles est visionnée par un opérateur. L'objectif de la détection d'évènements rares est de donner aux opérateurs des alertes, afin qu'ils puissent observer en direct les vidéos des caméras pointant vers ces évènements.

Dans la deuxième direction, l'objectif est de pouvoir reconnaitre automatiquement les personnes. Deux cas peuvent se distinguer suivant le système de caméras mis en œuvre :

1. Environnement contrôlé. Il correspond à un système où la scène et la position du visage sont contrôlées grâce à une coopération des individus. Cela permet de se placer dans le cadre où la reconnaissance faciale automatique peut être directement employée. Cette situation est rencontrée pour le contrôle d'identité, notamment lors de vérification des passeports aux frontières.
2. Environnement non-contrôlé. C'est le cas des caméras qui n'ont pas été initialement placées pour la reconnaissance faciale automatique et les individus ne sont pas en situation de coopération. Ce cas se rencontre, par exemple, avec les caméras situées sur la voie publique utilisées pour détecter des délits ou des accidents et engager des interventions en direct.

Actuellement, les solutions intelligentes de vidéosurveillance en environnement non contrôlé ne peuvent être entièrement automatisées. Toutefois, des outils effectuant des tâches semi-automatisées (nécessitant une vérification par un opérateur) commencent à être de plus en plus sérieusement envisagés.

Le sujet de cette thèse a été motivé par un besoin soulevé par la police dans le cadre d'enquêtes. Suite à un délit, les archives vidéos de la scène peuvent être utilisées pour guider

l'enquête. Le visionnement de plusieurs heures de vidéo est long et fastidieux. Il serait plus aisé et plus rapide de ne consulter qu'un album photo des personnes apparaissant dans la vidéo en question. L'opérateur pourrait ensuite vérifier directement au bon endroit dans la vidéo. L'objectif de cette thèse a été de contribuer à la recherche et au développement d'un système permettant la construction automatique d'un album de visages représentant les individus visibles sur une vidéo.

Dans l'idéal, l'objectif est de construire un album des visages de toutes les personnes dont le visage est visible sur la vidéo et ne comportant pas de doublons. Pour que le système soit utilisable par la police, il faudrait que le logiciel puisse traiter des scènes très denses (jusqu'à 16 individus par image) sur des vidéos avec une résolution allant jusqu'à 704×576 (en pixel), avec une fréquence maximale de 25 images par seconde. Une autre contrainte très forte est de pouvoir traiter de faibles résolutions de visages (allant de 50×50 à 20×20 pixels). Par rapport à la qualité de l'album construit, les tolérances se situent entre 3% et 10% pour les doublons, 1% et 10% d'omission d'individu et 3% et 5% pour les fausses détections. Enfin, les temps de calculs ne doivent pas être rédhibitoires. En pratique, il ne faudrait pas prendre plus de temps à analyser la vidéo que la durée de la vidéo elle-même.

En plus de ces contraintes fortes, on peut souligner la difficulté de l'évaluation d'un tel système. Un des obstacles majeurs réside dans l'obtention d'une base de test suffisamment large pour représenter au mieux la qualité de la méthode élaborée. La vidéosurveillance fait intervenir des contextes de prises de vues et d'illumination très différents. Les vidéos de test ne pouvant toutes les représenter, il faut faire attention à la sur-extrapolation des résultats obtenus. Les vidéos publiques, représentant fidèlement la vidéosurveillance non-contrôlée, n'existent pas en accès libre. Et pour des raisons de droit à l'image, les vidéos issues de cas réels de vidéosurveillance ne peuvent pas être utilisées.

Finalement, il reste très difficile de répondre aux besoins soulevés par la vidéosurveillance. Premièrement, les vidéos provenant des caméras installées ne sont pas favorables à la fouille automatique ou à l'identification et deuxièmement très peu de vidéos sont disponibles ce qui freine l'évaluation.

1.3. Définition de la problématique

Le regroupement des visages d'une vidéo par individu, pour en extraire un album photo, fait intervenir trois étapes :

— détection des visages
— regroupement des visages détectés
— sélection des images les plus représentatives des visages.

Actuellement, la détection des visages de face est un domaine qui a déjà vu apparaitre de nombreuses solutions relativement performantes, même si le cas de la vidéosurveillance non-contrôlée peut encore poser problème. C'est pourquoi la détection des visages de face n'a pas été traité au cours de ces travaux de thèse, une solution déjà existante ayant été employée.

La sélection des images pour l'album a été brièvement abordée au cours de cette thèse, principalement parce que cette étape n'est pas la plus critique en termes de performance et présente des aspects subjectifs qu'il est difficile d'appréhender.

Les travaux de thèse se sont concentrés sur le deuxième point : le regroupement des détections de visages. Les techniques de reconnaissance faciale étant difficilement applicables aux

séquences de vidéosurveillance en environnement non contrôlé, le regroupement des détections de visages se rapproche de la problématique du suivi de visages. Toutefois, un intérêt particulier est porté sur l'identité des trajectoires plus que sur la précision des trajectoires. Cela a conduit la thèse vers le suivi visuel multi-cibles tout en gardant le point de vue du "regroupement de détections" en s'appuyant sur les théories du partitionnement de données. Plus théoriquement, deux questions ont guidé les travaux de recherche :

1. Comment définir le regroupement considéré comme optimal et comment le trouver ?
2. Comment décrire les détections et mesurer les similarités entre elles ?

De nombreuses directions de recherche ont été envisagées pour traiter au mieux la difficulté des situations rencontrées en pratique. Deux aspects pratiques ont particulièrement dirigé le choix des différentes techniques et théories employées :

1. le temps de calcul : traiter une vidéo dans sa globalité fait vite intervenir une complexité combinatoire importante. L'étude s'est concentrée sur un algorithme de regroupement compétitif en temps de calcul. Des stratégies de subdivision du problème ont aussi été employées pour un traitement séquentiel.
2. la simplicité du paramétrage et l'évolutivité du système : de nombreux efforts ont été faits pour éviter le sur-ajustement des paramètres et pour conduire à une approche ne nécessitant pas une trop grande expertise pour appliquer la méthode à des cas concrets. Cela permet aussi de faire évoluer le système en y ajoutant facilement des informations supplémentaires.

1.4. Contributions

Les contributions s'inscrivent principalement dans le cadre du problème du suivi multi-cibles global basé détections. L'aspect *basé détections* et encore plus l'aspect *global* sont très récents dans la littérature et n'ont pas encore été largement traités. Le côté global est à mettre en opposition aux approches séquentielles plus couramment utilisées ; il signifie que la vidéo est traitée dans sa globalité. L'aspect *détection* provient du fait qu'un détecteur, issu généralement d'un apprentissage supervisé, est utilisé. En s'appuyant sur un algorithme de regroupement existant, le problème du suivi multi-cibles a été modélisé en prenant le point de vue du partitionnement de données. Cette approche a conduit à des utilisations originales d'algorithmes de partitionnement.

D'un point de vue plus pratique, peu d'études présentent des résultats sur les cas difficiles de la vidéosurveillance en environnement non contrôlé. Même si les résultats finaux n'atteignent pas l'exigence de la vidéosurveillance, cette thèse donne les premiers résultats sur des situations réelles que l'on peut rencontrer en vidéosurveillance. Cette thèse a aussi permis de présenter les premiers résultats sur la construction d'album photo en vidéosurveillance non-contrôlée.

Plus en détails, les principales contributions apportées peuvent se décliner en cinq points :

1. adaptation et amélioration d'un algorithme permettant de trouver un regroupement optimal
2. mise en œuvre d'une stratégie séquentielle offrant la possibilité d'aborder de plus longues vidéo

3. élaboration d'une méthode d'estimation permettant de construire une mesure de similarités inter-détection adaptée
4. évaluation et comparaison avec des approches génériques de partitionnement de données
5. expérimentation traitant des situations de vidéosurveillance non-contrôlée.

1.5. Organisation du mémoire

Dans un premier temps, un état de l'art des différents aspects de la problématique sera présenté. Il traitera de la littérature sur l'identification d'acteurs pour les vidéos grand public et d'individus pour la vidéosurveillance non-contrôlée. Ensuite, les différentes approches de suivi multi-cibles et des approches génériques de partitionnement seront abordées. Les critères d'évaluation des résultats et la base de test mise en place seront aussi évoqués dans ce chapitre, cela permet notamment d'introduire tout au long du rapport des expérimentations illustrant les propos tenus.

Le chapitre 3 traite premièrement de la modélisation par maximum *a posteriori* mise en œuvre pour représenter le regroupement en trajectoires des visages détectés, puis de l'algorithme de résolution permettant de trouver une solution optimal.

Les différentes possibilités quant à la définition des similarités entre les détections (intervenant dans le terme de vraisemblance) sont présentées et comparées au chapitre 4. Une méthode originale d'estimation des similarités y est aussi détaillée.

Le chapitre 5 décrira les expérimentations menées pour donner un aperçu quantitatif des différents aspects de la méthode proposée. Des résultats qualitatifs sur des applications autres que celle du suivi de visages sont aussi présentés.

Le dernier chapitre fait un bilan des différents travaux et propose des améliorations envisageables.

Chapitre 2.

État de l'art

Ce chapitre présente un panorama des différentes publications relatives au regroupement de détections issues d'une vidéo ainsi que des méthodes d'évaluation et bases de test.

En premier lieu les recherches spécifiques au regroupement des visages d'une vidéo sont abordées. Il est question des études menées pour traiter les vidéos grand public (films, séries TV ...) et celles pour traiter le cas de la vidéosurveillance. L'accent est mis sur deux problématiques intervenant dans ce domaine : les suivis mono-cibles et l'association des suiveurs. Différents travaux de suivi multi-cibles seront ensuite présentés. L'état de l'art se concentrera sur les approches globales basées sur des détections.

Deuxièmement, quelques méthodes traitant les problèmes génériques de regroupement seront présentées. Une focalisation est faite sur les méthodes de partitionnement de données ayant été employées dans le cadre de cette thèse.

Troisièmement, différentes méthodologies d'évaluation des résultats seront abordées. Cette section détaillera aussi la mesure de qualité sélectionnée pour quantifier les résultats obtenus.

En dernier lieu, la base de test mise en place pour les expérimentations menées est présentée. Elle est décrite à ce niveau du rapport parce qu'elle permet d'appréhender les différentes évaluations qui sont présentées au cours du rapport.

2.1. Identification de visages issus de vidéos

La fouille des visages contenus dans une vidéo a toujours été l'objet de nombreux intérêts. Toutefois, pour ce qui est de la reconnaissance faciale, les études menées se concentrent principalement sur les cas des visages de face en environnement contrôlé. Un nombre plus restreint de recherches ont porté sur le cas des vidéos qui n'ont pas été spécialement conçues pour l'identification des visages.

Deux domaines d'application de la fouille de visage peuvent se distinguer : l'étiquetage des acteurs des vidéos du grand public (émissions TV, films, séries TV ...) et l'identification faciale en vidéosurveillance. Le premier domaine a donné lieu à des techniques très variées et complexes en terme de suivi et de regroupement pour pouvoir traiter la grande variété des situations rencontrées. Dans le cadre de la vidéosurveillance, l'accent est plutôt mis sur la reconnaissance faciale, mais présente encore actuellement beaucoup d'obstacles.

La section suivante décrit les différentes recherches menées dans l'objectif d'étiquetage automatique d'acteurs des vidéos du grand public. Ensuite, les études appliquées à la vidéosurveillance seront abordées.

2.1.1. Étiquetage d'acteurs de vidéos du grand public

L'analyse de contenus vidéo a fait l'objet de nombreuses recherches. Comme la plupart des vidéos de la vie courante, elles font intervenir des figurants humains, et l'étiquetage automatique des acteurs a tout particulièrement intéressé l'industrie du multimédia de l'Internet.

Les travaux publiés dans ce domaine sont peu nombreux mais très variés et font intervenir un cumul de techniques et de théories très diverses. En ajoutant à cela la jeunesse des publications, un classement ou une taxinomie de ces travaux n'est pas chose facile.

Parmi l'ensemble des solutions publiées dans le cadre de l'identification d'acteurs, deux étapes interviennent : l'extraction des éléments (détections de visages ou petites trajectoires de visages) et l'identification (ou le partitionnement) de ces éléments afin de les associer à des acteurs.

Pour la première étape, les éléments construits sont généralement des petites trajectoires issues du suivi visuel. On peut noter que certains travaux (J. Choi et al. 2010 ; O. Arandjelovic et A. Zisserman 2005) ne font pas intervenir de suivi et traitent directement le partitionnement des détections de visages en n'utilisant que leur apparence. Cela nécessite une bonne qualité des images ainsi que de nombreux traitements pour permettre la reconnaissance des visages.

Comme le montre la publication R. C. Verma et al. 2003, le couplage de la détection de visages et de son suivi permet de prendre en compte la continuité temporelle qu'il y a dans une vidéo. La plupart des travaux d'étiquetage d'acteurs, tout en ne l'utilisant pas de la même façon que R. C. Verma et al. 2003, s'appuient sur l'information du détecteur et de la continuité provenant de la vidéo pour suivre les visages. La section 2.2 présente plus en détails différentes approches du suivi basé détections.

Pour aller plus loin, certains travaux (M. C. Nechyba et al. 2008 ; N. Pande et al. 2012) font intervenir plusieurs détecteurs de visages suivant la position de la tête (face, profil droit/gauche et 45° droit/gauche).

Les travaux de M. C. Nechyba et al. 2008 mettent l'accent sur le suivi et un paramétrage adapté à la situation. Même si ces travaux ne présentent pas de résultats dans le cadre de l'étiquetage d'acteurs, ils ont permis la mise en œuvre de belles démonstrations (par la société *PittPatt* rachetée par *Google*) sur des séries TV. Mais comme le système de regroupement des trajectoires par identité n'a pas été publié, il est difficile de dire si la qualité des résultats provient d'une bonne reconnaissance des visages de face ou s'ils font aussi intervenir toutes les images de visages des trajectoires.

Pour ce qui est des travaux de N. Pande et al. 2012, ils font intervenir un suivi mono-cible par *Mean-Shift* basé sur des histogrammes des couleurs. Pour passer au suivi multi-cibles, les auteurs mettent en place des stratégies de fusion/séparation et d'entrée/sortie de nouvelles cibles. Elles restent empiriques et font intervenir de nombreux paramétrages et seuillages.

J. Sivic et al. 2005 présentent une approche basée sur des descripteurs de points d'intérêt. Une fois les visages de face extraits par un détecteur, ils sont décrits par des descripteurs SIFT correspondant à des points d'intérêt des composantes faciales (yeux, nez et bouche). Ces travaux présentent l'originalité de suivre toutes les régions de l'image au cours de la vidéo pour ensuite se baser sur ces suivis afin de relier les détections. Les détections posées sur des régions se correspondant en termes de suivi, seront considérées comme proches et fusionnées. L'intérêt de cette approche est de suivre tous les objets mobiles de la scène, ce qui permet notamment de mieux interpréter les occultations.

Après avoir décrit comment les différentes approches extrayaient les trajectoires de visages,

on peut s'intéresser aux méthodes mises en œuvre pour les regrouper par acteur.

Mise à part la méthode de O. Arandjelovic et A. Zisserman 2005 où il est question de requête d'acteur par image, l'étiquetage d'acteurs fait intervenir une étape de partitionnement des trajectoires par individu. Pour ce faire, les efforts se sont concentrés sur la construction d'inter-distances plutôt que sur l'algorithme de partitionnement en lui-même. En effet, les solutions proposées par J. Sivic et al. 2005 ; J. Choi et al. 2010 ; A. Fitzgibbon et A. Zisserman 2002 ; A. Fitzgibbon et A. Zisserman 2003 ; D. Ramanan et al. 2007 ; M. Tapaswi et al. 2012 mettent en œuvre un simple regroupement agglomératif plus ou moins proche du partitionnement ascendant hiérarchique (cf. section 2.3 pour la description de l'algorithme).

Il existe tout de même quelques travaux qui mettent davantage l'accent sur l'aspect partitionnement de données. En s'appuyant sur le partitionnement spectral (décrit plus précisément à la section 2.3.2.1), les travaux de S. Foucher et L. Gagnon 2007 ; N. Vretos et al. 2011 présentent de bons résultats en étiquetage. Comme le montrera la suite du rapport, notre étude a aussi mis l'accent sur l'aspect partitionnement, même si ce n'est pas l'approche classique du suivi multi-cibles.

Pour ce qui est du calcul des similarités entre les trajectoires d'individus, plusieurs approches peuvent être citées.

J. Sivic et al. 2005 utilisent 5 descripteurs SIFT par visage pour construire un vecteur de caractéristiques par visage. À l'instar des méthodes de type "sac-de-mots", l'espace de ces vecteurs de caractéristiques est arbitrairement partitionné en classes, et un histogramme d'occurrences de ces classes est construit à l'aide des visages de chaque trajectoire. Cet histogramme est alors utilisé pour mesurer la similarité entre deux ensembles (trajectoires) de visages. L'intérêt de cette approche est de représenter (de façon statistique et non-paramétrique) les différentes poses de visages apparaissant au cours d'une trajectoire.

N. Pande et al. 2012 présentent une approche paramétrique, basée sur un modèle de mélange de gaussiennes. Ce modèle n'emploie qu'une imagette normalisée de l'apparence de chaque visage, la majorité des regroupements étant déjà acquis par le suivi.

Certaines études (A. Fitzgibbon et A. Zisserman 2002 ; A. Fitzgibbon et A. Zisserman 2003) font intervenir des variétés (au sens géométrique) qui permettent de représenter la forme des ensembles de détections. Ces variétés sont construites pour être invariantes aux transformations affines et adaptées à la comparaison de visages. Comme le montre A. Fitzgibbon et A. Zisserman 2003, cela permet de mesurer des distances *image* à *image*, *ensemble d'images* à *image* et *ensemble d'images* à *ensemble d'images*. Grâce à cette métrique, un partitionnement agglomératif permet de regrouper les séquences de visage par individu.

M. Tapaswi et al. 2012 présentent un modèle probabiliste (basé sur les champs de Markov) qui permet de fusionner différents indices : reconnaissance des visages, des habits, détection de ceux qui parlent et découpage en sous-séquences. Même s'il reste difficile de comparer les travaux par manque de bases de test communes, cette approche semble donner de bons résultats.

Il est important de noter que de nombreux travaux n'utilisent pas que l'information visuelle de la vidéo, mais aussi les sous-titres attachés aux films ou séries TV. Cela se fait généralement en utilisant un détecteur de mouvements des lèvres pour identifier les acteurs en train de parler, et une correspondance est faite avec les sous-titres. Parfois le script est aussi employé pour directement relier les dialogues avec les acteurs. Cette information supplémentaire permet notamment d'utiliser de l'apprentissage semi-supervisé pour bien classer

les visages des acteurs (M. Bäuml et al. 2013 ; J. Sivic et al. 2009). D'après les résultats de M. Everingham et al. 2009, la moité des performances est acquise par les sous-titres, le reste provenant de l'information visuelle et notamment du suivi des visages.

Finalement, les études du domaine applicatif des vidéos grand public ont permis la mise en œuvre de stratégies assez complexes pour le traitement de vidéos très diverses. Les meilleurs résultats présentés dans ce domaine n'utilisent pas que l'information visuelle, mais aussi les textes de sous-titres. Pour ce qui est des méthodes n'employant pas d'information textuelle, le suivi de visages est bien exploité (parce que la qualité des images de visages le permet) et débouche sur de bons résultats.

2.1.2. Reconnaissance faciale en vidéosurveillance non-contrôlée

La reconnaissance faciale dans le cadre de la vidéosurveillance non-contrôlée a engendré un grand nombre de recherches, mais surtout de nombreux obstacles.

L'étude récente de J. Barr et al. 2012 présente un panorama des travaux dans ce domaine. Il permet de voir qu'un panel très divers de théories et techniques est employé pour tenter de donner des solutions à la reconnaissance faciale en vidéosurveillance non-contrôlée. Selon leur taxinomie, les différentes approches peuvent se regrouper en deux classes.

La première traite les visages issus des vidéos comme une collection non ordonnée d'images et tire avantage de la multiplicité des observations. Dans cette classe de solutions, une grande importance est accordée à l'analyse fine des images (super-résolution ou reconstruction 3D) et à la construction d'espaces de représentation des groupes de visages.

Le deuxième type d'approche utilise explicitement les données temporelles et construit des séquences d'images. Ces approches permettent le traitement de vidéos plus dégradées et tirent parti du suivi de visages pour la reconnaissance. Les travaux menés au cours de cette thèse s'inscrivent plutôt dans cette catégorie.

Parmi les approches basées sur les séquences, on peut noter une approche probabiliste intéressante (K. Lee et al. 2005) cherchant à représenter les séquences par des variétés en représentant conjointement les aspects suivi et reconnaissance.

En termes applicatifs, la reconnaissance faciale en vidéosurveillance non-contrôlée fait intervenir des obstacles provenant de la qualité de la vidéo. La publication de J. Barr et al. 2012 donne un bon aperçu des difficultés rencontrées. En environnement non-contrôlé, de nombreux facteurs altèrent l'apparence des visages :

- Variation de pose : les visages ne sont pas nécessairement de face avec une pose canonique, comme le sont les visages en environnement contraint.
- Variation d'illumination : en passant sous des éclairages différents l'apparence des visages varie, notamment à cause d'ombrages.
- Variation de l'expression : le visage change fortement en fonction des expressions (rire, clignement de paupière, discussion ...).
- Variation d'échelle : un visage proche peut être très différent d'un visage éloigné, le changement de résolution peut influer les différents détails discriminent du visage.
- Flou : lorsque les individus se déplacent il apparait très souvent un flou dû au mouvement.
- Occlusions : différentes parties du visage peuvent être occultées par un autre objet ou individu de la scène.

Face à toutes ces altérations possibles, il arrive souvent qu'une image du visage d'un individu à un instant donné soit plus éloigné d'une image de ce même individu à un autre

instant que de celui d'un autre individu. Le fait que la variation de l'apparence du visage d'un même individu puisse être supérieure à celles entre différents individus rend la reconnaissance faciale très difficile.

À ces nombreux facteurs d'altération, il nous faut ajouter les aspects liés à la caméra elle-même. Les caméras installées dans le cadre de la vidéosurveillance sont souvent de coût plutôt réduit et donnent une mauvaise qualité d'image. De plus, pour avoir un accès temps-réel aux images, la compression est inévitable ce qui dégrade aussi la qualité. Comme le soulignent les travaux de C. Fiche 2012, en prenant uniquement en compte la compression spatiale, 6 effets perturbant la reconnaissance faciale ont été recensés. Dans le cadre des vidéos, on pourrait encore ajouter à cela la compression temporelle.

Figure 2.1. Aperçu des bases de test en reconnaissance faciale. Les deux bases de gauche (*FERET* et*LFW*) sont classiquement employées pour évaluer la reconnaissance faciale. *scFace* est une base présentant la réelle qualité des images provenant de la vidéosurveillance. Les images de *vidéo 6* et *vidéo 7* proviennent de la base de test construite pour tester les méthodes mises en œuvre au cours de cette thèse. Le schéma à droite permet de se rendre compte de la différence de résolution existant entre les bases de test pour la reconnaissance faciale en environnement contrôlé et en vidéosurveillance non-contrôlée.

La principale différence entre la reconnaissance faciale basée sur des images fixes et celle basée vidéo est la résolution des images. La figure 2.1 donne un aperçu des visages issus des bases de test les plus courantes. Il peut y avoir jusqu'à un facteur 5 entre les tailles des images provenant des vidéosurveillances et celles d'images de test en reconnaissance faciale. Étant donné le grand écart entre les bases d'évaluation des approches classiques de reconnaissance et la vidéosurveillance en environnement non-contrôlé, les techniques de reconnaissance faciale n'ont pas été abordées au cours de cette thèse.

Pour la grande majorité des recherches en reconnaissance faciale basée vidéo, les conditions sont telles que la détection et le suivi des visages ne représentent pas la tâche la plus difficile. Et la plupart des bases de test (R. Gross et J. Shi 2001 ; K.-C. Lee et al. 2003 ; K. Lee et al. 2005 ; M. Kim et al. 2008 ; P. J. Phillips et al. 2009 ; M. Grgic et al. 2011) font intervenir des séquences vidéo où n'apparaissent pas deux visages différents au sein de chaque séquence. Il est uniquement question de reconnaissance faciale, l'extraction des séquences d'images par individu est supposée faite. C'est pourquoi nous avons construit une base de test (cf. 2.5) constituée de vidéos faisant intervenir les résolutions de visages rencontrées en vidéosurveillance non-contrôlée et faisant intervenir plusieurs visages par image. Cela a permis d'avoir un ensemble de vidéos représentant à la fois les qualités des visages issues de la vidéosurveillance et les problématiques de suivi multi-cibles.

2.1.3. Conclusion

Dans le cadre de cette thèse, la qualité des vidéos à traiter ne permet pas une bonne exploitation du suivi des visages et l'identification des personnes est difficilement abordable par les méthodes développées dans le cadre de l'annotation d'acteurs. L'autre aspect rendant le problème d'étiquetage d'acteurs différent de celui de la création d'album photo en vidéosurveillance, est le nombre d'identités recherchées. Les travaux dans le domaine de l'étiquetage d'acteurs font intervenir des vidéos avec peu de personnes et l'objectif est de ne bien identifier que les acteurs principaux (en général une dizaine). Alors que, dans le cadre de la vidéosurveillance, des séquences de quelques minutes peuvent faire intervenir une centaine d'individus. C'est pourquoi, transposer la qualité des résultats présentés avec l'étiquetage d'acteurs à ceux de la vidéosurveillance ne peut se faire.

La tâche de l'extraction de séquences de visages par individu est plus en lien avec la problématique d'album photo que la reconnaissance faciale. De plus, la qualité des séquences vidéo à traiter ne permet pas l'utilisation de la reconnaissance faciale. Finalement, dans le cadre des travaux de cette thèse, l'objectif étant le traitement de vidéos en environnement non-contrôlé avec des résolutions de visages et des qualités très faibles, les techniques de reconnaissance faciale n'ont pas été abordées.

Pour conclure, les études citées ici ont été utilisées au cours de cette thèse pour donner un panorama de techniques, mais les systèmes présentés n'ont pu être directement employés.

2.2. Suivi multi-cibles basé détections

L'ensemble des recherches relatives au domaine du suivi multi-cibles basé vision peut se diviser en deux sous-domaines. Cette division est faite par la manière dont le temps est appréhendé. Ces deux sortes d'approches seront qualifiées de *séquentielles* ou *globales*. Les approches *globales* ont principalement vu le jour grâce à l'utilisation de détecteurs provenant d'apprentissages statistiques.

Traiter le suivi multi-cibles en s'appuyant sur les détections, et cela de manière globale, correspond exactement à notre cadre applicatif.

Cette section décrit les différents aspects liés au suivi multi-cibles basé détections, à savoir : les détections et le suivi comportant des approches séquentielles et globales.

En premier lieu, un bref panorama des détecteurs basés apprentissage est dressé, en mettant l'accent sur les détecteurs de visages qui nous concernent plus particulièrement.

Ensuite, quelques méthodes séquentielles de suivi multi-cibles basé détections sont présentées.

Pour finir, les approches globales de suivi multi-cibles sont abordées. Comme l'on cherche à traiter des vidéos d'archive et non pas un flux vidéo en direct, les approches globales correspondent mieux à notre cadre applicatif d'analyse d'archives vidéo.

2.2.1. Détecteur de visages de face

Les outils de reconnaissance faciale sont difficilement applicables à la construction d'albums photo en vidéosurveillance non-contrôlée. Les efforts menés depuis quelques années en détection faciale ont tout de même permis de détecter les visages de face dans ces conditions très contraintes.

Les approches de type *apprentissage supervisé* se sont imposées dans le domaine. Elles font intervenir deux phases, l'une *hors-ligne* et l'autre *en ligne*. La phase *hors-ligne* est constituée de trois étapes :

1. Construction d'une base d'apprentissage : des exemples positifs (visages) et des exemples négatifs (non-visages) sont recueillis.
2. Description des exemples. Chaque exemple est décrit par un vecteur résumant l'information contenue dans l'image. Les descripteurs les plus courants sont : les ondelettes de Haar, les *Local Binary Pattern*, les histogrammes d'orientation des gradients et les matrices de covariance.
3. Apprentissage de la fonction de décision. Ici intervient un algorithme qui, au vu des descripteurs de la base d'apprentissage, va construire une fonction de tri de l'espace de description. Cette fonction permet de définir si un descripteur est un visage ou non. L'usage des algorithmes de type *Adaboost* s'est imposé dans le domaine.

Ensuite, face à une image donnée dont on recherche les visages, la phase *en ligne* intervient. Elle suit le procédé suivant :

— Parcours des sous-fenêtres de l'image. Il s'effectue en parcourant un ensemble de sous-images en faisant glisser une sous-fenêtre sur différentes positions et échelles.
— Description des sous-images. Chaque sous-image est décrite en prenant le descripteur employé pour la phase *hors-ligne*.
— Décision. Les descripteurs sont déterminés comme positifs ou négatifs suivant leur valeur obtenue par la fonction de décision apprise.

La figure 2.2 illustre les différentes étapes intervenant pour un détecteur. Ce type d'apprentissage a déjà été mis en œuvre avec succès pour des détecteurs de visages, piétons ou voitures.

Les travaux les plus marquants dans le domaine de la détection de visages sont ceux de P. Viola et M. Jones 2001. Ils font intervenir un algorithme de type *Adaboost* pour l'apprentissage de la fonction de décision, les images sont décrites par des ondelettes de Haar et la structure en cascade est employée. La popularité de cette méthode vient probablement du fait qu'elle permet une détection temps-réel, principalement grâce à un descripteur rapide ainsi qu'à la cascade.

La structure en cascade correspond à une succession de fonctions de décision (classifieurs). Les premiers niveaux de cette cascade sont constitués de classifieurs traitant les cas les plus simples où les décisions sont vite prises, alors que les derniers niveaux traitent les cas difficiles. Cette stratégie permet de rapidement mettre de côté les négatifs (qui sont en pratique très nombreux) et de prendre le temps de décider les cas difficiles (qui sont moins fréquents).

Les expérimentations menées au cours de cette thèse ont employé un détecteur en cascade du même type que celui de P. Viola et M. Jones 2001, à l'exception du descripteur : *Local Binary Pattern* est pris à la place des ondelettes de Haar. Ce détecteur à été choisi pour sa disponibilité, sa qualité et sa rapidité.

2.2.2. Suivi multi-cibles séquentiel

Ces dernières années, grâce aux performances des détecteurs (piétons ou visages), plusieurs méthodes de suivi visuel utilisant ces détecteurs sont apparues, et l'expression *tracking-by-detection* avec elles. Au sens de M. Breitenstein et al. 2010, *tracking-by-detection* désigne

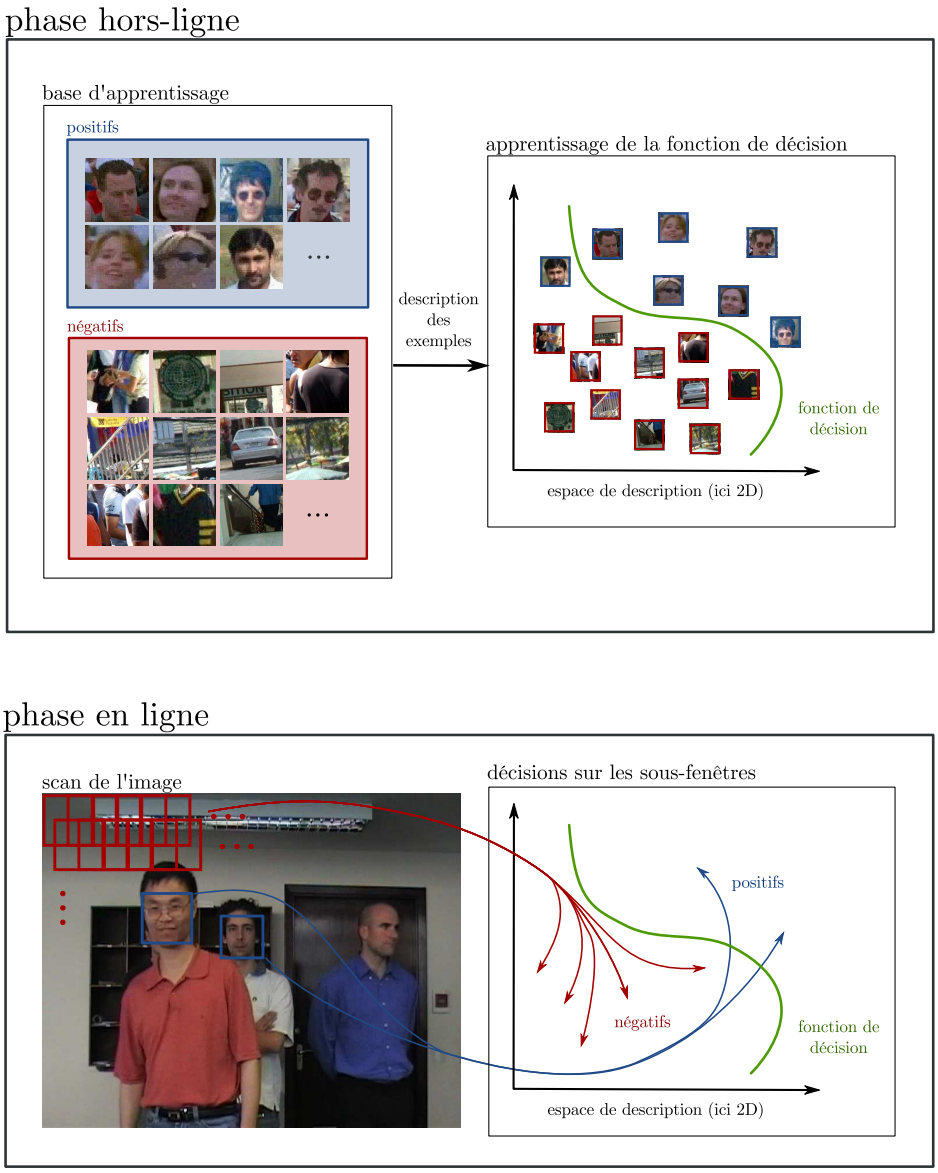

FIGURE 2.2. Illustration des différentes étapes d'un détecteur.

des méthodes de suivi multiple se basant sur des détections d'objets à chaque image et une association de ces détections au cours d'une vidéo.

Parmi les méthodes de *tracking-by-detection*, deux catégories se sont distinguées : les approches séquentielles (ou causales) et les approches globales (non causales) s'appuyant sur l'ensemble de la vidéo ou sur une fenêtre temporelle glissante.

Les méthodes les plus rependues en suivi multi-cibles basé vision sont celles faisant intervenir une approche séquentielle. Séquentielle veut dire que les états des objets suivis à un instant de la vidéo ne dépendent que de leurs antécédents. Ainsi, image par image, les états des cibles sont ré-estimés en fonction des estimations faites précédemment. Ces approches sont parfois qualifiées de *causales*.

De nombreuses méthodes très diverses ont vu le jour dans le cadre du suivi visuel. Et même si le suivi d'un seul objet peut paraître moins complexe que le suivi multiple, il existe encore de nombreux cas où le suivi visuel reste encore un réel défi. Les différentes solutions

apportées au problème du suivi ne sont pas abordées dans le cadre de ce rapport.

Le point important à évoquer pour passer du simple suivi au suivi multi-cibles est de pouvoir bien distinguer et apparier les observations faites à l'état courant du modèle estimé dans le passé. Historiquement, pour traiter ce point, deux stratégies probabilistes se sont distinguées : le MHT (Multiple Hypothesis Tracker) et le JPDAF (Joint Probabilistic Data Association Filter). Le MHT garde en mémoire les différentes associations potentielles au cours du temps. Sa complexité limite souvent le nombre d'hypothèses conservées. D'un autre côté, le JPDAF recherche à effectuer les meilleures associations entre les pistes estimées et les nouvelles détections apparaissant. Ces deux approches ont nourri bon nombre de techniques de suivi.

Toutefois, dans le cadre du suivi basé vision, quand une attention particulière est accordée à la description des différentes cibles, des heuristiques simples sont souvent suffisantes. La solution présentée par M. Breitenstein et al. 2010 donne un bon exemple de méthode s'appuyant sur cette idée. Elle fait intervenir un mécanisme d'apprentissage qui permet de bien décrire les différentes cibles. L'autre intérêt de cette étude vient du fait qu'elle utilise, dans un même cadre, l'information du détecteur et tout le système de suivi multi-cibles. Bien que cette approche ne soit pas globale, elle donne des résultats intéressants dans des cas difficiles de suivi de piétons.

Les approches séquentielles n'ont pas fait l'objet d'une grande attention au cours de la thèse, parce que l'application visée invitait plus à une analyse globale de la séquence vidéo. Elles restent tout de même le fondement sur lequel les approches globales se sont érigées.

2.2.3. Suivi multi-cibles global

En supposant avoir accès directement à la totalité de la vidéo et non à un flux, notre problématique s'inscrit très bien dans le cadre du suivi global (en opposition au séquentiel). Traiter le suivi comme une optimisation globale sur l'ensemble de la vidéo est une approche plutôt récente et peu de travaux ont été réalisés dans ce sens. La plupart des recherches ont été évaluées pour le cas du suivi de piétons, et elles ne traitent que rarement de longues vidéos présentant des situations très diverses.

De par la jeunesse des recherches traitant de ce sujet, les méthodes recensées ici sont peu nombreuses et très différentes en théorie comme en pratique.

Plusieurs méthodes de suivi multi-cibles global s'appuient sur une stratégie en deux étapes : associer premièrement les détections les plus proches pour construire de courts suivis, et ensuite fusionner ces suivis pour aboutir aux trajectoires complètes.

Dans la plupart des cas, la première étape est faite en associant les détections les plus proches (C. Huang et al. 2008 ; J. Henriques et al. 2011). Cette première étape est cruciale, car si de mauvaises associations apparaissent, le système n'est plus capable de les remettre en cause. La publication de J. Prokaj et al. 2011 donne un bon aperçu des différentes approches et présente un système d'inférence plus complet pour pallier les défauts liés à cette première étape. Une fois ces petites trajectoires construites, la deuxième étape fait en général intervenir divers systèmes d'association.

D'autres méthodes se distinguent par le fait qu'elles traitent le problème du suivi avec plusieurs angles de vue sur la même scène. En général, ces méthodes font intervenir des cartes d'occupation correspondant à une discrétisation du sol où évoluent les piétons. C'est le cas des travaux de A. Andriyenko et K. Schindler 2010 et de J. Berclaz et al. 2009.

Ensuite, on peut aussi citer la méthode de B. Leibe et al. 2007 qui présente un couplage de la détection et du suivi par une optimisation globale, et la méthode de Y. Li et al. 2009 qui s'appuie sur de l'apprentissage statistique pour traiter les associations. Cette dernière a le défaut de nécessiter préalablement un apprentissage supervisé.

Les deux principaux obstacles rencontrés par les approches globales sont la complexité calculatoire et la complexité de la modélisation et du paramétrage. Nous présentons maintenant les apports des différentes méthodes face à ces deux obstacles.

2.2.3.1. Différentes méthodes de résolution

Le temps de calcul des approches globales est souvent rédhibitoire. Différentes modélisations sont utilisées pour représenter le problème, mais des méthodes très diverses de résolution ont aussi vu le jour. Ces méthodes peuvent se classer en deux catégories selon qu'elles soient probabilistes ou déterministes.

Les résolutions probabilistes les plus employées sont probablement celles fondées sur un échantillonnage par MCMC (*Monte Carlo Markov Chain*). Cette popularité vient du fait que la plupart des modélisations sont probabilistes et donnent donc déjà un cadre au MCMC, mais aussi parce que ces méthodes sont relativement simples à mettre en œuvre tout en étant potentiellement rapides. Parmi celles-ci nous en avons principalement recensé trois : W. Ge et R. Collins 2008 ; Q. Yu et G. Medioni 2009 ; Z. Tao et al. 2008.

Dans le cadre du suivi séquentiel, les approches probabilistes ont un atout qu'il est important de souligner : elles ne se contentent pas d'estimer un maximum *a posteriori* mais permettent de propager dynamiquement l'ensemble de la loi *a posteriori*. Pour ce qui est de l'approche globale, cet atout est à relativiser : on ne s'intéresse généralement qu'à un seul maximum de l'*a posteriori* et non plus à l'ensemble de la loi. C'est probablement pour cette raison que les optimisations déterministes sont plus rependues dans le cadre du suivi multi-cibles global que dans celui du suivi multi-cibles séquentiel.

Parmi les méthodes déterministes de résolution, on peut citer :

— Algorithme hongrois : cet algorithme traite le problème des associations de ressources. Il est souvent utilisé en suivi séquentiel pour associer trajectoires du modèle et nouvelles détections, mais aussi en suivi global (C. Huang et al. 2008) modulo quelques modifications.

— Flot de coût minimal : se base sur la théorie des graphes pour traduire le problème d'optimisation comme un problème de circulation de flot. Cette méthode a été employée pour cette thèse. Elle se base sur les travaux de L. Zhang et al. 2008.

— Programme linéaire : les travaux de J. Berclaz et al. 2009 mettent en place une résolution par un programme linéaire. Cette résolution suppose une discrétisation de l'espace des positions des cibles et construit une carte d'occupation. Elle s'applique surtout dans une situation multi-vues avec calibration des caméras pour estimer les positions au sol.

L'algorithme hongrois et le flot de coût minimal se basent sur des modélisations du problème sensiblement équivalentes. L'idée principale est de trouver un ensemble de trajectoires qui minimisera la somme des coûts des transitions issues des trajectoires et d'un coût lié au nombre de trajectoires. L'intérêt de ces deux approches est de donner une solution optimale avec une complexité calculatoire peu élevée, grâce à des méthodes d'optimisation déjà existantes et ayant déjà des implémentations efficaces.

Une autre piste envisagée par plusieurs travaux consiste à découper le problème en une

succession de sous-problèmes. Une fenêtre temporelle glissante (B. Benfold et I. Reid 2011 ; Q. Yu et G. Medioni 2009) est souvent utilisée pour réduire la complexité calculatoire. Cela revient à traiter une succession de suivis globaux le long de la vidéo. Et pour des raisons de temps de calcul, la taille des fenêtres temporelles est souvent très réduite.

Au cours de la thèse, la méthode déterministe basée sur le flot de coût minimal a été choisie. Le choix d'une telle méthode déterministe a été motivé par le fait qu'elle ne nécessite pas d'expertise particulière, contrairement aux méthodes probabilistes qui demandent la construction de lois de propositions. Ce choix a aussi été motivé par le fait que la représentation par un graphe donne un cadre théorique intéressant à la représentation d'une solution (cf. section 3.1.1).

Des stratégies de découpage du problème ont aussi été envisagées pour réduire la complexité calculatoire (cf. section 3.3).

2.2.3.2. Complexité de la modélisation

Quand on cherche à mettre en œuvre un algorithme de suivi multi-cibles global, le premier frein est souvent la grande quantité de paramètres. Cela vient du fait que le paramétrage du suivi multi-cibles séquentiel est déjà important, et qu'il faut rajouter les paramètres gérant l'organisation des trajectoires dans leur globalité (ils sont mis en général dans la partie *a priori*).

À titre d'exemple on peut citer les travaux de Q. Yu et G. Medioni 2009 qui mettent en place un modèle probabiliste dont l'optimum est trouvé par une méthode d'échantillonnage. Dans ce cas, le nombre de paramètres est très élevé, parce qu'il y a les paramètres liés au modèle et aussi ceux liés au processus d'échantillonnage qui permettent de faire converger l'algorithme vers une bonne solution. Les paramètres sont fixés en utilisant une vérité terrain experte et une optimisation permettant de les fixer de telle sorte que la convergence du MCMC soit garantie.

Dans la plupart des travaux faisant intervenir un paramétrage complexe (L. Zhang et al. 2008 ; Q. Yu et G. Medioni 2009 ; W. Ge et R. Collins 2008 ; Y. Li et al. 2009), il est question d' "apprentissage" des paramètres (ou au moins d'une partie des paramètres) à partir d'une séquence où la vérité-terrain est connue. Cette étape n'est pas toujours accessible dans la pratique. Particulièrement en vidéosurveillance, face à la variété des scènes observées, il n'est pas possible d'avoir des vérités-terrain sur chaque situation rencontrée. Ainsi des réglages de paramètres effectués à partir de quelques situations peuvent s'avérer inadaptés à d'autres situations rencontrées.

Les modélisations élaborées au cours de cette thèse ont mis l'accent sur la réduction de la complexité de la représentation. Cela a été fait dans le but d'être le plus général possible pour traiter la grande variété des cas survenant en vidéosurveillance et pour permettre une utilisation qui ne demande pas beaucoup d'expertise. La définition de l'*a priori* (section 3.1.2) et l'estimation des probabilités de transition (section 4.2.4) illustrent particulièrement cela.

2.3. Point de vue du partitionnement de données pour le regroupement de détections

Supposant avoir accès aux détections de visages et cherchant à les regrouper par individu, la problématique de la thèse, même si elle se rapproche de celle du suivi visuel multi-objets, peut se définir de manière plus générale comme un partitionnement des détections d'une vidéo. Le partitionnement de données (parfois appelé classification, apprentissage non-supervisé, ou encore *clustering*) est un domaine à part entière qui a déjà vu naître de nombreuses théories avec des degrés de complexité plus ou moins variés. Les différents algorithmes issus de ce domaine ont déjà été largement exploités dans la classification d'images ou même dans la classification de visages. Toutefois, ce type d'approche n'a pas été appliqué au regroupement de détections issues d'une vidéo ou encore au suivi multi-cibles basé détections. Le suivi multi-cibles global (décrit à la section 2.2) a vu apparaitre ses propres algorithmes provenant principalement de ce qui avait été fait dans le cas du suivi séquentiel. Dans notre étude, nous avons voulu mettre en avant que les recherches menées dans le cadre du partitionnement de données peuvent servir au traitement du problème du suivi basé détections. Cela pourrait permettre de généraliser le suivi à une large variété des situations et éviter le sur-ajustement des méthodes.

En parlant de *partitionnement*, son sens mathématique est considéré et il peut se définir de la manière suivante :

un partitionnement d'un ensemble $\mathcal{D}$ est un ensemble T de sous-ensembles de $\mathcal{D}$ non vides et deux à deux disjoints.

Certaines définitions ne font pas intervenir l'aspect non-vide des ensembles constituant un partitionnement, mais nous avons décidé de tenir compte de cet aspect pour que $|T|$ désigne bien le nombre de groupes des détections (nombre de trajectoires).

Dans le domaine du partitionnement de données, le nombre de parties reste une question épineuse. Dans la littérature, de nombreuses investigations ont été faites dans ce sens, mais nous n'en présenterons pas ici. Pour traiter ce problème, nous avons préféré construire un terme d'*a priori* spécifique à notre cadre applicatif plutôt qu'une solution générique provenant du partitionnement de données.

Ce rapport ne prétend pas à un état de l'art exhaustif des différentes solutions apportées au problème du partitionnement de données. L'accent est mis sur les approches qui emploient uniquement une mesure des inter-similarités entre éléments. Premièrement, parmi les méthodes de partitionnement les plus couramment employées, celles qui paraissent les plus adaptées à notre problématique sont décrites. Ensuite, nous présentons quelques techniques employées pour améliorer la fonction de similarité entre objets.

2.3.1. Algorithmes standards de partitionnement

Les deux algorithmes présentés ici peuvent paraître simplistes mais sont les incontournables du partitionnement. Ils sont amplement employés dans un grand nombre de situations, notamment dans le regroupement de détections issues d'une vidéo.

2.3.1.1. Partitionnement hiérarchique

Le partitionnement hiérarchique fait partie des outils les plus largement utilisés dans l'analyse de données. Son fonctionnement est assez simple et l'un de ses atouts est de ne

faire intervenir que la notion de similarité entre objets. Le partitionnement hiérarchique que nous décrivons est de type ascendant (ou agglomératif) que l'on nommera HAC (*Hierarchical Agglomerative Clustering*), la variante descendante étant peu utilisée, principalement pour des raisons de complexité calculatoire. Les approches hiérarchiques ne donnent pas directement un partitionnement, mais une hiérarchie qui peut ensuite donner lieu à différents partitionnements. En cherchant à partitionner un ensemble $\mathcal{D} = \{x_1, ..., x_D\}$ et ayant une similarité $s_{ij} \in \mathbb{R}$ pour chaque paire (x_i, x_j) d'éléments, l'algorithme général peut se décrire de la façon suivante :

1. initialiser le partitionnement courant $T = \{\{x_1\}, ..., \{x_D\}\}$ en isolant tous les éléments
2. fusionner les deux ensembles de T les plus proches
3. si $|T| > 1$ aller en 2, sinon s'arrêter.

Comme ne sont fusionnés que deux éléments de T à chaque itération, les différents T obtenus au cours de l'algorithme permettent de construire une hiérarchie. Le point important est celui de l'étape 2, et plusieurs stratégies peuvent être employées pour définir les groupes de T les plus proches au vu de s. Typiquement trois stratégies peuvent être employées pour définir une proximité entre deux sous-ensembles disjoints A et B de D :

— *single-linkage* :

$$\max_{x_i \in A \; x_j \in B} s_{ij} \tag{2.1}$$

— *complete-linkage* :

$$\min_{x_i \in A \; x_j \in B} s_{ij} \tag{2.2}$$

— moyenne :

$$\frac{1}{|A||B|} \sum_{x_i \in A \; x_j \in B} s_{ij} \tag{2.3}$$

Dans notre cas d'utilisation le *single-linkage* semble le plus pertinent et c'est aussi ce qui a été observé en pratique. La raison en est expliquée dans le chapitre traitant des expérimentations.

Le HAC peut paraître assez simpliste, mais les expérimentations menées ont permis de nous rendre compte qu'elle reste compétitive et a même surpassé dans quelques cas une méthode qui paraissait beaucoup plus spécialisée au problème du suivi multi-cibles global. Le principal problème rencontré par cette approche est le fait de ne pas remettre en cause un regroupement fait au début de l'algorithme. Si une similarité très élevée est rencontrée pour deux détections provenant de différentes personnes, la fusion va être faite et ne plus être remise en cause, alors qu'une approche minimisant un critère global peut refuser de regrouper ces deux détections au vu de la globalité du problème. Un autre point important est que le partitionnement n'est pas directement acquis par le HAC. Il reste une étape de "découpage" de la hiérarchie pour aboutir à un partitionnement des détections. Plusieurs méthodes peuvent être employées pour ce découpage : seuiller la similarité, fixer le nombre de groupes à obtenir ou encore une stratégie fusionnant les deux méthodes. Au cours des expérimentations, le choix du nombre de groupes a été préféré.

2.3.1.2. Algorithme *k-means*

L'algorithme *k-means* fait partie des méthodes les plus courantes pour partitionner des données. Comme l'explique bien A. K. Jain 2010, malgré le nombre conséquent de méthodes

de partitionnement de données, le *k-means*, avec ses multiples variantes, reste encore très largement utilisé.

Voici les grandes lignes de l'algorithme du *k-means* :

1. choisir aléatoirement k éléments de $\mathcal{D}$ qui seront les centres des k groupes initiaux
2. assigner chaque élément au cluster dont le centre est le plus proche
3. tant qu'au moins un élément change de cluster : ré-estimer chaque centre en moyennant les éléments du cluster correspondant et aller en 2

Outre le fait qu'il faut spécifier au préalable un nombre de classes (k), la méthode nécessite un espace de représentation des données qui rende possible le calcul d'une moyenne sur un ensemble de données. Se basant sur des détections de visages qui font intervenir des notions de positions, temps et apparences, il est difficile de se placer dans un espace qui soit capable de bien représenter une moyenne. De ce fait, la variante *k-medoids* parait plus adaptée. Cette variante ne fait pas intervenir une moyenne pour représenter une classe, mais utilise directement une donnée "médiane" qui sera utilisée comme représentante d'une classe. Ainsi, une simple similarité entre les paires de points à classer est demandée, ce qui permet une application directe au regroupement de détections.

De manière générale, l'algorithme du *k-means* recherche à minimiser la somme des carrés des distances entre les données et la moyenne du groupe auquel elles appartiennent. Cette méthode reste sensible à l'initialisation des centres et converge vers des optima locaux.

Un des points faibles des méthodes issues du *k-means* est la convexité des groupes construits. Or, si l'on s'intéresse au regroupement des détections provenant d'une vidéo, les groupes ont plutôt des formes de chaines, car, au sein d'un regroupement réaliste, une détection est en général proche de celles issues des frames proches et peut être éloignée de celles étant de nombreuses frames en avant ou en arrière.

Une des méthodes pour pallier ce problème est de re-conditionner les distances entre les éléments pour qu'elles prennent mieux en compte la répartition des données. Cela est fait par exemple avec le partitionnement spectral qui est abordé à la section 2.3.2.1. Ce type d'approche permet aussi d'utiliser le *k-means* sur des données dont uniquement les similarités entre elles sont utilisées (ne nécessite pas de distance ni de calcul de moyenne).

2.3.2. Approches pour améliorer les similarités entre les éléments

Dans le cadre du partitionnement de données, le choix des similarités entre les différents éléments à classer est crucial. Ce choix peut avoir un impact important sur la forme des groupes. Un autre enjeu dans la construction des similarités est de pouvoir utiliser conjointement des informations qui sont hétérogènes et appartiennent à des espaces de représentation différents. Face à ces deux enjeux (forme particulière des groupes et hétérogénéité des informations qu'utilise la fonction de similarité), nous nous sommes penchés sur deux approches relativement récentes dans le domaine du partitionnement de données : le partitionnement spectral et l'*ensemble clustering*.

2.3.2.1. Approche spectrale

Ces dernières années, d'après U. Von Luxburg 2007, un ensemble d'algorithmes, qualifié de partitionnement spectral (*spectral clustering*), s'est imposé parmi l'ensemble des méthodes de partitionnement. La notoriété du partitionnement spectral vient principalement de ses

performances face aux approches plus classiques décrites précédemment, mais aussi du fait qu'il est relativement facile à mettre en œuvre et peut se résoudre par des méthodes standard d'algèbre linéaire.

Le partitionnement spectral se base sur la matrice des inter-dissimilarités entre les objets à partitionner. C'est la seule information dont l'algorithme a besoin. La procédure consiste à transformer cette matrice pour que les groupes soient plus discernables. Ensuite, est employé un algorithme classique de partitionnement, le plus couramment choisi étant l'algorithme *k-means*. Comme nous l'avons déjà décrit précédemment, l'exécution directe d'un *k-means* donne des groupes convexes, alors qu'avec les inter-dissimilarités issues d'une approche de type spectrale, le *k-means* permet de représenter des formes plus complexes de regroupement. La figure 2.3 provenant de A. Ng et al. 2002 illustre cela.

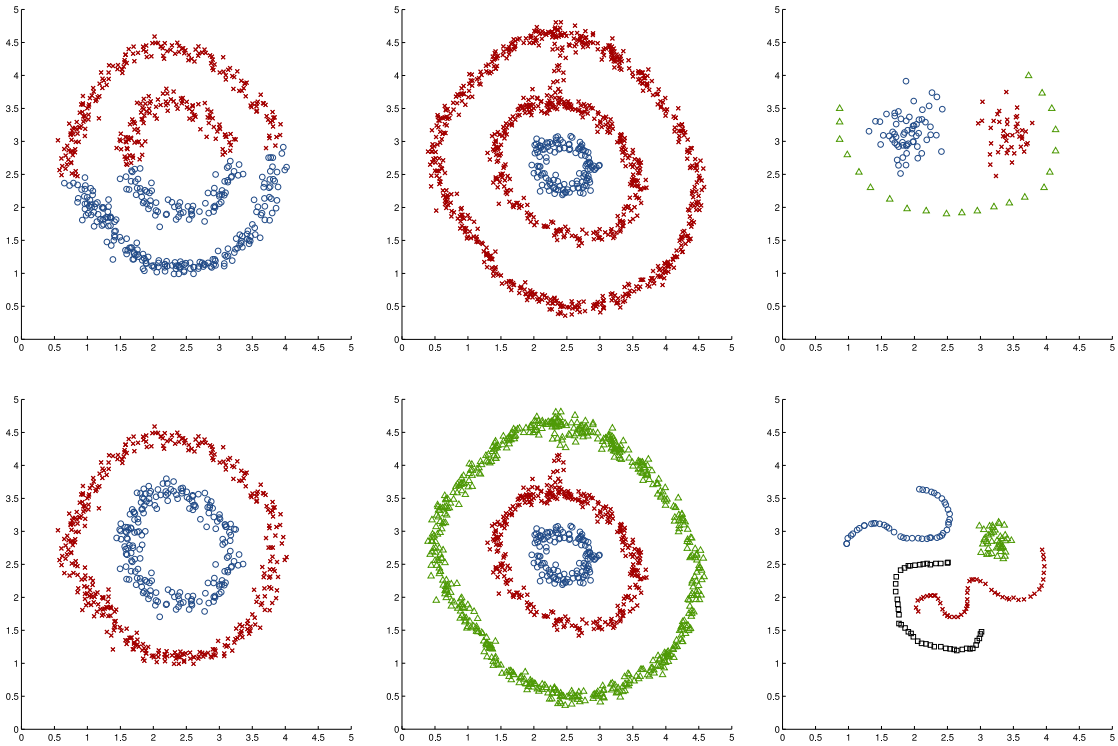

Figure 2.3. Exemples de résultats de partitionnement spectral sur des données synthétiques 2D avec distance euclidienne. En haut à gauche : *k-means* avec deux classes, en bas à gauche : partitionnement spectral avec *k-means* et deux classes, les autres figures représentent des résultats obtenus par partitionnement spectral et *k-means* où seul le nombre de classes du *k-means* varie d'un cas à l'autre. Ces résultats proviennent de A. Ng et al. 2002.

Un grand avantage du partitionnement spectral est qu'il n'a pas d'*a priori* sur la forme des groupes ; il traite le problème en terme de partitionnement de graphe. Cela autorise aussi à traiter des problèmes avec un grand nombre d'éléments dont peu de liens existent entre eux. Comme il se base sur une structure de graphe entre les éléments, il est possible de ne pas envisager toutes les connexions. Cet aspect est particulièrement adapté à la problématique du regroupement de détections car les groupes peuvent avoir des formes différentes (selon que l'on utilise par exemple une mesure d'apparence entre les détections ou une mesure spatio-temporelle) et la modélisation que nous avons employée se base déjà sur une structure de graphe entre les détections.

Pour plus de détails, nous renvoyons au tutoriel de U. Von Luxburg 2007. Sur le cadre théorique, la publication de A. Ng et al. 2002 apporte des précisions, notamment sur les conditions d'utilisation et d'optimalité.

Le partitionnement spectral a aussi été employé, notamment par H.-C. Huang et al. 2012, avec un ensemble de matrices des similarités. La publication présente des résultats de regroupement de visages de la base CMU-PIE, où les auteurs utilisent le partitionnement spectral sur trois types de similarités (*Eigenface*, *Gabor texture* et *Local Binary Pattern*). Cette approche a attiré notre attention du fait des résultats présentés sur le regroupement de visages et l'aspect "fusion" des similarités. Ayant déjà concentré nos efforts pour la fusion des similarités par l'*Ensemble clustering*, l'approche de H.-C. Huang et al. 2012 a finalement été mise de côté.

Le partitionnement spectral a aussi été récemment employé (N. Vretos et al. 2011 ; S. Foucher et L. Gagnon 2007) dans un cadre applicatif proche de celui de la thèse. Cela souligne l'importance que prend ce type d'approche dans le domaine.

Tout comme les méthodes de partitionnement décrites précédemment, le partitionnement spectral ne règle pas le problème du nombre de classes, il nécessite sa connaissance *a priori*. La publication de P. Perona et L. Zelnik-Manor 2004 le souligne. Les auteurs proposent une approche basée sur l'analyse spectrale de la matrice des similarités pour sélectionner le nombre de parties, mais le problème reste encore ouvert.

2.3.2.2. *Ensemble clustering*

Face à l'hétérogénéité des informations manipulées pour construire les similarités entre les détections, on peut se demander s'il ne serait pas intéressant de construire un ensemble de partitionnements issus de ces différentes informations et de les fusionner pour obtenir un meilleur partitionnement. Ce principe a déjà fait l'objet de quelques travaux, et les méthodes qui en ont émergé ont été regroupées sous le terme de *ensemble clustering* (par analogie au *ensemble learning* qui a notamment donné naissance au *boosting*) parfois aussi appelé : *consensus clustering*. L'idée fondatrice de l'*ensemble clustering* est d'utiliser un ensemble de partitionnements de qualité moindre pour construire un partitionnement de meilleure qualité. L'article A. Strehl et J. Ghosh 2003 a servi de base à plusieurs autres travaux dans ce domaine.

Les travaux de A. Strehl et J. Ghosh 2003 présentent trois heuristiques pour approcher le partitionnement partageant, en moyenne, le plus d'information (au sens de l'information mutuelle) avec l'ensemble de partitionnement de départ :

- *Cluster-based Similarity Partitioning Algorithm* (CSPA) : cet algorithme s'appuie sur la construction d'une mesure de similarité basée sur le nombre de fois que les partitionnements ont regroupé des paires d'éléments. Ces similarités sont ensuite employées par un algorithme de partitionnement quelconque.
- *HyperGraph Partitioning Algorithm* (HPGA) : cette méthode représente l'ensemble de départ comme un hypergraphe et utilise un algorithme de partitionnement d'hypergraphe minimisant le nombre d'hyper-arcs coupés pour construire le partitionnement.
- *Meta-Clustering Algorithm* (MCLA) : cet algorithme se base aussi sur l'hypergraphe de l'ensemble de départ mais va chercher à fusionner successivement les hyper-arcs les plus similaires.

Face à la largeur du champ applicatif lié au partitionnement de données, il est difficile de les comparer. On peut tout de même citer une publication (A. Goder et V. Filkov 2008)

qui en évalue quelques-unes et présente encore une autre variété d'heuristiques.

2.3.3. Conclusion

Dans le cadre de cette thèse, les algorithmes standards de partitionnement ont été employés pour le regroupement de détections de visages. Ils ont servi de comparatif pour les expérimentations menées. Deux approches de partitionnement (*ensemble clustering* et partitionnement spectral) ont aussi conduit nos recherches dans le cadre des similarités entre détections.

2.4. Méthodes d'évaluation

Cette section présente plusieurs approches permettant d'évaluer une solution obtenue. Un système cherchant à regrouper des détections de visages par personne peut s'évaluer à plusieurs niveaux.

Premièrement, si elle est vue comme un algorithme de suivi multi-cibles basé détections, on peut s'intéresser à la précision des trajectoires obtenues.

Deuxièmement, si l'objectif est de classer les détections et que l'on se place dans un cadre où il y a plusieurs objets à suivre, les mesures des erreurs d'identification et des redondances utilisées pour le suivi sont importantes.

Troisièmement, on peut s'intéresser non plus à la qualité d'une solution en terme de suivi, mais en terme de partitionnement des détections.

Et finalement, comme l'on recherche à obtenir un album photo représentatif de la vidéo, il est important de pouvoir mesurer la qualité d'une solution en ayant extrait un représentant par groupe construit.

Les sections suivantes reprennent dans l'ordre ces quatre niveaux d'évaluation.

2.4.1. Mesure de la précision des suivis

Il existe de nombreuses méthodes pour évaluer la qualité d'un suivi multi-cibles basé vidéo. C. Needham et R. Boyle 2003 citent plusieurs méthodes permettant de mesurer la qualité d'un suivi par rapport à une vérité-terrain, cependant elles mesurent surtout la précision en terme de position, ce qui dans notre cas de figure ne nous intéresse pas véritablement.

Le protocole d'évaluation du projet *CLEAR 2006-2007* propose des critères pour évaluer la performance des suivis. Dans le cadre du suivi multi-cibles, trois critères (MOTA, MOTP et STDA) semblent les plus utilisés. Plus précisément :

Multi-Object Tracking Accuracy : mesure de l'exactitude des suivis

$$MOTA = 1 - \frac{\sum_{t=1}^{N_{frames}} m_t + fp_t + mme_t}{\sum_{t=1}^{N_{frames}} g_t}$$

m_t : nombre d'erreurs de suivi pour la frame t (sans compter les faux positifs)
fp_t : nombre de faux positifs pour la frame t

mme_t : nombre d'erreurs d'appariement pour la frame t
g_t : nombre d'objets (vérité-terrain) présents à la frame t
Multi-Object Tracking Precision : mesure de la précision des positions des suivis

$$MOTP = \frac{\sum_{i=1}^{N_{mapped}} \sum_{t=1}^{N_{frames}} overlap(G_i^t, D_i^t)}{\sum_{t=1}^{N_{frames}} N_{mapped}^t}$$

G_i^t : élément du suivi i de la vérité-terrain pour la frame t
D_i^t : élément du suivi i pour la frame t
N_{mapped} : nombre de suivis associés à la vérité-terrain
N_{mapped}^t : nombre d'objets de la frame t associés à des objets de la vérité-terrain
$overlap(x, y)$: recouvrement entre les zones recouvertes par l'élément, il est calculé par le rapport de l'intersection sur l'union des zones suivies
Sequence Tracking Detection Accuracy : mesure de la précision (variante de MOTP)

$$STDA = \sum_{i=1}^{N_{mapped}} \frac{\sum_{t=1}^{N_{frames}} overlap(G_i^t, D_i^t)}{|FG_i \cup FD_i|}$$

FG_i : frames support du suivi i de la vérité-terrain
FD_i : frames support du suivi i trouvé

Certaines publications (R. Stiefelhagen et al. 2006 ; K. Bernardin et R. Stiefelhagen 2008) font intervenir une distance de localisation (en mm par exemple) au lieu d'une mesure de recouvrement (*overlap* en %). Dans la pratique, la mesure de recouvrement est la plus employée pour présenter des résultats. Cela vient du fait que les distances réelles ne sont pas toujours accessibles et ne sont pas utiles dans le cadre du suivi 2D car il n'y pas de calibration.

Il est important de noter que ces mesures demandent une association un à un entre les objets suivis à un instant t et ceux de la vérité-terrain au même instant. Il peut arriver que la trajectoire estimée soit la meilleure représentation pour deux suivis de la vérité-terrain et vice-versa (cf. figure 2.4), et l'appariement peut alors conduire à des ambiguïtés. Dans le cadre de ces critères, une procédure basée sur une optimisation des appariements peut être employée (R. Stiefelhagen et al. 2006).

Parmi ces critères, le MOTA est celui qui peut nous intéresser le plus, parce qu'il met l'accent sur les changements d'identifiants. Toutefois, la redondance (plusieurs suiveurs sur le même objet) n'est pas directement mesurée.

2.4.2. Mesure des erreurs d'identification

Par rapport à l'objectif final, les critères suivants semblent plus raisonnables :
— erreurs : nombre d'éléments associés à un mauvais suivi vérité-terrain (FIO)
— faux positifs : nombre d'éléments associés à aucun suivi vérité-terrain
— changements : nombre de changements d'identifiant pour un même suivi (FIT)

— redondance : nombre de suivis associés à la même vérité-terrain

Le problème de la redondance est bien présent dans notre cas, car les objets suivis peuvent prendre des apparences très différentes ou être occultés ce qui entraîne de nombreux arrêts de suivi.

Pour ces critères, il nous faut pouvoir dire à quel suivi de la vérité-terrain va correspondre un suivi trouvé. Une solution simple consiste à associer chaque élément du suivi trouvé à l'élément de la vérité-terrain le plus proche. Le suivi trouvé sera ensuite associé au suivi de la vérité-terrain le plus représenté.

K. Smith et al. 2005 mettent en place une version plus élaborée de cette méthode par une double identification : la vérité-terrain identifie l'estimation et l'estimation est identifiée par la vérité-terrain.

Plus précisément, les trajectoires estimées sont parcourues pour leur associer les suivis vérité-terrain, ce qui permet de compter les erreurs d'identification (FIO : False Identified Object). La vérité-terrain est ensuite parcourue pour associer les suivis estimés et mesurer la redondance (FIT : False Identified Track). Ces erreurs sont calculées pour chaque frame. Afin d'avoir une erreur globale, ces erreurs sont normalisées par le nombre d'objets de la vérité-terrain, puis moyennées sur l'ensemble des images de la vidéo :

$$FIT = \frac{1}{|F|} \sum_{f \in F} \frac{FIT_f}{\max(1, N_{GT}^f)}$$

$$FIO = \frac{1}{|F|} \sum_{f \in F} \frac{FIO_f}{\max(1, N_{GT}^f)}$$

où F est l'ensemble des frames, N_{GT}^f est le nombre d'objets de la vérité-terrain présents à la frame f. La figure 2.4 illustre ces erreurs d'identification.

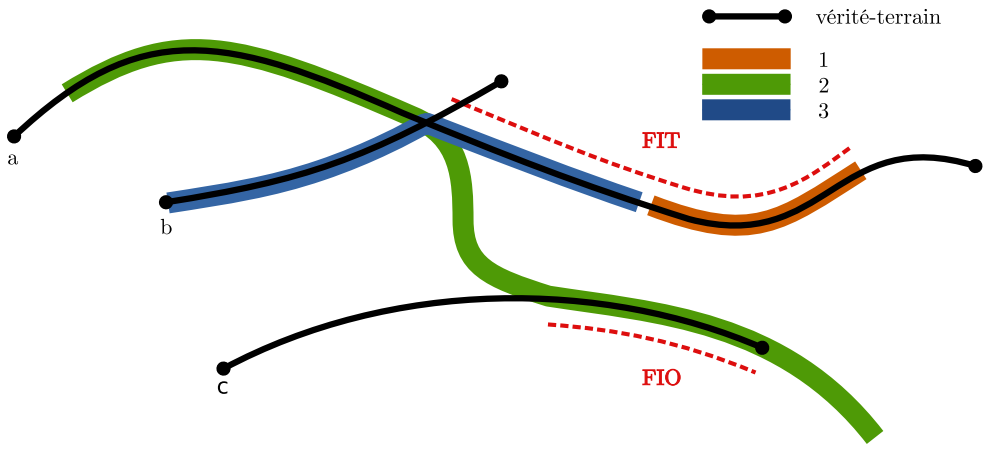

Figure 2.4. Illustration des deux types d'erreurs (FIT et FIO) liées à d'identification. Les traits noirs (a,b et c) représentent les trajectoires de la vérité-terrains et les traits colorés (1,2 et 3) sont les trajectoires estimées par l'algorithme.

K. Smith et al. 2005 définissent des notions de *pureté* qui sont particulièrement avantageuses dans notre cas d'étude, car elles permettent de mesurer : la capacité de la solution à représenter les différentes trajectoires (*Tracker Purity* : TP) de la vérité-terrain, et la capacité à vérifier que la vérité-terrain soit bien représentée par les suiveurs (*Object Purtity* :

OP). Ces puretés sont illustrées à la figure 2.5 qui présente aussi les mesures de faux positifs (FP) et de faux négatifs (FN) employées.

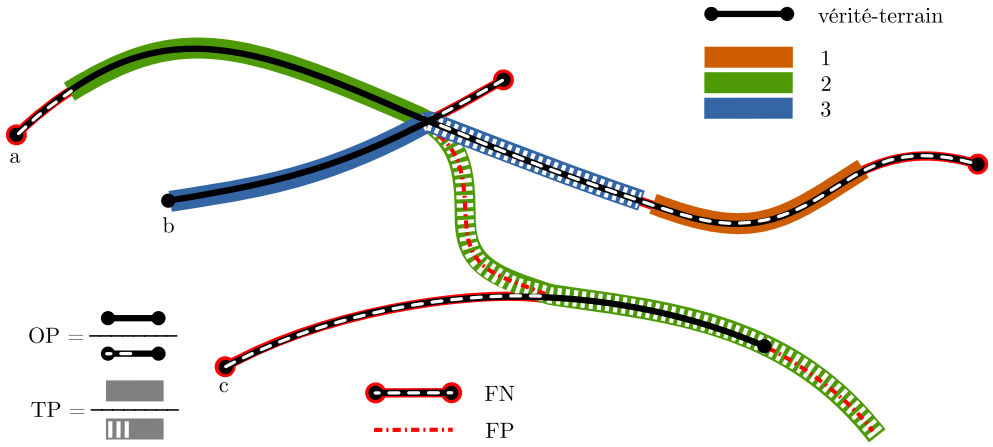

FIGURE 2.5. Illustration de la méthode proposée par K. SMITH et al. 2005 pour mesurer les puretés (OP et TP). Cela suppose une identification des suiveurs ($1 \rightarrow a$, $2 \rightarrow a$ et $3 \rightarrow b$) et des objets par les suiveurs ($a \rightarrow 2$, $b \rightarrow 3$ et $c \rightarrow 2$). FN (resp. FP) représente le taux de faux négatifs (resp. faux positifs).

Les puretés définies par K. SMITH et al. 2005 nécessitent l'utilisation d'un algorithme permettant de construire des cartes d'identification de l'estimation par rapport à la vérité-terrain et de la vérité-terrain par rapport à l'estimation. L'idée de pureté du suiveur et des objets nous a amené à nous pencher sur ce principe, non plus dans le domaine du suivi, mais dans le domaine du partitionnement de données. Et c'est cette dernière approche qui a finalement été employée.

2.4.3. Mesure de la qualité d'un partitionnement

Il existe plusieurs façons de mesurer la qualité d'un algorithme de partitionnement. Les méthodes dites *intrinsèques* n'utilisent pas de vérité-terrain construite par un expert, elles mesurent les proximités des éléments au sein d'un regroupement et les distances entre les groupes. En ce qui nous concerne, nous nous intéressons plutôt à des mesures quantitatives et nos regroupement représentent des objets ayant une réalité physique. Les méthodes dites *extrinsèques* sont plus adaptées à notre situation, car il existe un partitionnement jugé optimal : celui qui classe les détections de visages par personnes et contient un groupe des faux positifs.

Comme le montre une étude des méthodes extrinsèques d'évaluation (E. AMIGÓ et al. 2009), il existe de nombreuses approches pour évaluer la qualité d'un partitionnement : le comptage de paires d'éléments suivant qu'elles soient classées de la même façon par les deux partitionnements (vérité-terrain et estimé), la pureté des groupes ou encore l'information mutuelle calculée entre le partitionnement estimé et celui de la vérité-terrain.

Principalement pour des raisons d'interprétation des résultats et pour faire suite à l'intérêt que nous avons porté aux OP et TP de K. SMITH et al. 2005, nous avons mis en place un critère basé sur la pureté et la pureté inverse. L'avantage à utiliser ces deux critères est de pouvoir déterminer si la mauvaise qualité d'un partitionnement provient d'erreurs de regroupements (groupes contenant des détections de différentes personnes) ou d'erreurs de représentations (une personne est représentée par plusieurs groupes).

Par la suite, *partitionnement estimé* (noté E) désignera un partitionnement obtenu par des algorithmes proposés et *partitionnement vérité-terrain* (noté GT) celui qui est utilisé comme référence, construit manuellement. De façon similaire, ce qui est couramment appelé pureté sera nommée *pureté de l'estimation* (noté EP) et la pureté inverse sera nommée *pureté vérité-terrain* (noté GTP). La *pureté de l'estimation* représente la pureté de l'estimation en prenant la vérité-terrain comme référence, et la *pureté vérité-terrain* celle de la vérité-terrain en se basant sur le partitionnement estimé.

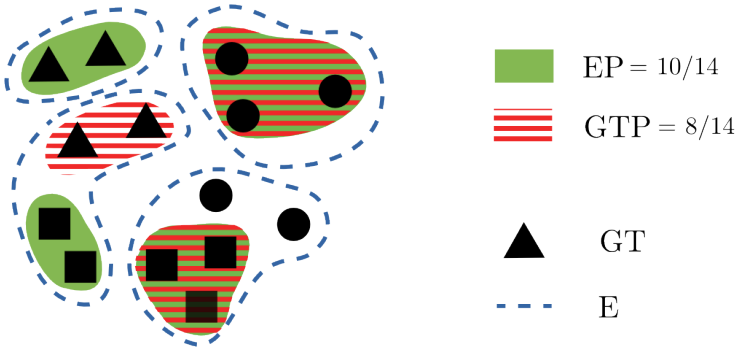

FIGURE 2.6. Illustration de la *pureté de l'estimation* (EP) et de la *pureté vérité-terrain* (GTP).

Ces deux critères (EP et GTP) sont définis de la manière suivante (cf. figure 2.6) :

— **pureté de l'estimation** :

$$EP \doteq \frac{1}{D} \sum_k \max_j |E_k \cap GT_j| \tag{2.4}$$

où D désigne le nombre de détections, $GT \doteq \{GT_j\}$ le partitionnement vérité construit par un expert et $E \doteq \{E_k\}$ le partitionnement estimé par un algorithme. Au plus EP est élevé, au moins il y a d'erreurs de recouvrement. Les *erreurs de recouvrement* apparaissent quand un groupe inclut des détections provenant de différentes personnes.

— **pureté vérité-terrain** :

$$GTP \doteq \frac{1}{D} \sum_k \max_j |GT_k \cap E_j| \tag{2.5}$$

mesure la proportion de personnes bien représentées. Au plus GTP est élevé, au moins il y a de détections d'une personne apparaissant dans plusieurs groupes estimés.

Cependant, ces mesures sont biaisées, notamment par le nombre de détections ou de personnes de la vérité-terrain. Si l'on considère les deux cas extrêmes :

— un cluster regroupe toutes les détections :

$$EP = \frac{\max_j |GT_j|}{D} \text{ (noté } EP_{min}) \text{ et } GTP = 1$$

— tous les groupes ne contiennent qu'une détection :

$$EP = 1 \text{ et } GTP = \frac{|GT|}{D} \text{ (noté } GTP_{min})$$

On voit que le minimum de pureté atteignable n'est pas nul mais dépend du nombre de détections D, de la taille du plus grand cluster de GT et du nombre de personnes $|GTP|$. Afin de pouvoir comparer les qualités des algorithmes sur différents jeux de données, les deux critères ont été modifiés pour que leur minima soient nuls :

$$EP_n = \frac{EP - EP_{min}}{1 - EP_{min}} \tag{2.6}$$

$$GTP_n = \frac{GTP - GTP_{min}}{1 - GTP_{min}} \tag{2.7}$$

Au plus EP et GTP seront toutes les deux proches de 1, au plus le partitionnement estimé sera proche de la référence. Pour utiliser en un même critère les deux puretés, nous utiliserons la F-mesure, qui s'utilise couramment pour la recherche d'information. Ainsi la F-mesure permettra de représenter la qualité d'un partitionnement par une valeur dans $[0, 1]$. Elle est définie par :

$$\text{F-pureté} = 2\frac{EP_n \times GTP_n}{EP_n + GTP_n} \tag{2.8}$$

elle sera nommée F-pureté (pour signifier F-mesure entre puretés) et sera utilisée pour les expérimentations.

2.4.4. Mesure de la qualité d'un album

L'objectif final est de bien rassembler les détections d'une même personne, d'éviter les faux positifs et surtout de n'oublier personne. En supposant que l'on ait sélectionné une photo représentative de chaque cluster, on peut utiliser les critères suivants :

— doublons :

$$C_d = \frac{N - N_p^r}{N} = 1 - \frac{N_p^r}{N} = 1 - \text{précision} \tag{2.9}$$

— retrouvés :

$$C_r = \frac{N_p^r}{N_p} = \text{rappel} \tag{2.10}$$

— fausses détections :

$$C_f = \frac{FP}{N} \tag{2.11}$$

avec les notations suivantes :

— N : nombre de photos de l'album
— N_p^r : nombre de personnes retrouvées par l'album
— N_p : nombre de personnes détectées dans la vidéo (vérité-terrain)
— FP : nombre de photos de l'album provenant de faux positifs du détecteur

C_d représente la proportion de doublons et C_r la proportion de personnes retrouvées. On peut noter que $1 - C_r$ représentera la quantité de personnes oubliées dans l'album. Ces critères peuvent avoir une interprétation en termes de précision-rappel couramment employés dans la recherche d'information. La quantité $1 - C_d$ représente la proportion de personnes retrouvées par l'album et peut s'apparenter à une précision et C_r au rappel.

2.5. Base de test

Tout au long de ce mémoire de thèse, des expérimentations sont présentées pour illustrer les propos tenus. C'est ce qui a motivé la présentation de la base de test mise en place à ce niveau du rapport.

Dans le cadre de la vidéosurveillance en environnement non contrôlé, peu d'études traitent le regroupement des détections de visages et il ne semble pas exister de base de test réaliste (basse qualité d'images avec effets de compression, faible résolution des visages et des scènes denses en terme de nombre de personnes).

Une série d'acquisitions vidéo ont été faites dans le cadre du laboratoire et du projet BioRafale. Deux autres séquences ont aussi été intégrées à l'ensemble des vidéos de test. La première, tournée initialement pour la détection de colis abandonnés, a été sélectionnée parce qu'elle représente une scène typique de vidéosurveillance. La seconde, bien qu'elle ne soit pas un cas de vidéosurveillance, permet d'évaluer le suivi de visages lors d'occultations et de rotations de la tête. Un total de neuf vidéos allant de 30 secondes à 3 minutes a été utilisé comme base de test.

Des statistiques de ces vidéos figurent à la table 2.1 et des aperçus des scènes filmées sont donnés aux figures 2.7 et 2.8.

vidéo	durée	passages	frames	nb detect.	FP	taille visage
1	1m17s	24	1934	1725	2.78%	35.0 (10.1)
2	38s	7	951	1135	1.41%	42.2 (11.6)
3	12s	6	307	200	11.5%	35.2 (8.3)
4	15s	7	384	920	2.61%	35.9 (11.9)
5	41s	6	485	463	3.45%	57.9 (11.8)
6	1m43s	29	1966	1794	1.56%	62.8 (10.0)
7	55s	11	1394	6309	16.3%	39.8 (14.1)
8	3m22s	16	5060	1299	22.4%	31.6 (9.1)
9	40s	4	1006	2686	3.76%	79.0 (11.3)

TABLE 2.1. Chiffres clés des vidéos d'évaluation. *passages* : nombre de personnes traversant le champ de vue et ayant au moins une détection de visage, *frames* : nombre total d'images dans la vidéo, *nb detect.* : nombre de détections de visages, *FP* : taux de faux positifs observés, *taille visage* : taille moyenne (et écart-type) des détections de visages (en pixels).

Les séquences vidéos 1 à 6 et 8 ont été tournées dans le cadre du laboratoire ou du projet BioRafale.

La vidéo 1 a été tournée en intérieur avec des variations d'illumination dues à l'éclairage. Elle fait intervenir des passages d'individus avec de nombreux croisements et changements de direction. Les détections de visages sont relativement éparses : plusieurs centaines de frames peuvent défiler avant de re-détecter le visage d'une même personne. La vidéo 2 présente la même scène, avec cependant des individus qui passent de façon alignée et dont les visages sont presque détectés tout au long de leur passage. Cette vidéo présente une situation plus facile à traiter.

Les vidéos 3 et 4 sont de courtes séquences où figure un groupe d'individus évoluant de manière très rapprochée et effectuant des mouvements très brusques. La taille et la qualité de ces vidéos sont à la limite de l'acceptable pour le détecteur de visages. Les individus ont

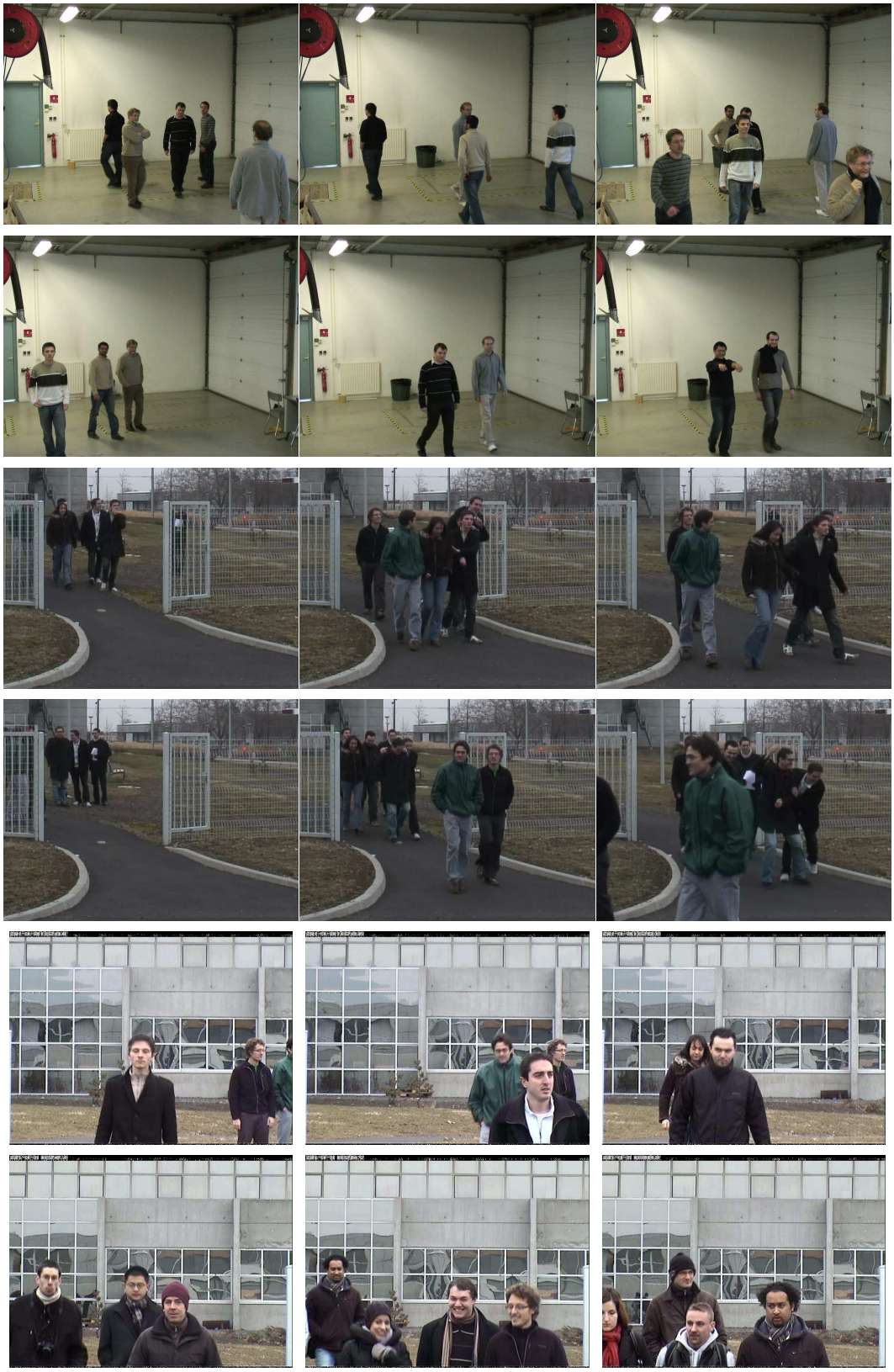

FIGURE 2.7. Les lignes présentent trois captures des vidéos 1 à 6 de haut en bas. Ces vidéos on été acquises dans le cadre du laboratoire et du projet BioRafale.

FIGURE 2.8. Les lignes présentent trois captures des vidéos 7 à 9 de haut en bas.

des couleurs de vêtements très similaires et il y a de nombreuses occultations des visages. Ces vidéos sont très difficiles à traiter.

Les vidéos 5 et 6 ont été tournées dans le cadre du projet BioRafale. Elles présentent des passages de plusieurs personnes en plan serré. La taille des visages est assez grande, mais l'exploitation de la reconnaissance faciale reste difficile voir impossible. Cela vient du fait que les visages bien résolus soient flous de par la vitesse de déplacement et que la caméra génère un bruit non négligeable.

La vidéo 7 a été tournée au laboratoire. Elle présente une vidéo de basse qualité proche de ce que l'on peut obtenir d'une caméra de vidéosurveillance. Les individus évoluent le long d'un couloir avec de fortes variations d'illumination et un flou de mouvement assez important, particulièrement quand les visages sont proches de la caméra.

La vidéo 8 provient d'une base de test construite pour évaluer la détection de bagages abandonnés dans le cadre de AVSS 2007 (*Advanced Video and Signal based Surveillance Conference*). Elle correspond à la vidéo *AVSS AB Hard* accessible à l'adresse : `www.eecs.qmul.ac.uk/~andrea/avss2007_d.html`. Elle présente une scène de métro qui fait intervenir des passages de piétons à la sortie des rames et aussi des personnes sur un banc dont les visages sont détectés de manière éparse tout au long de la vidéo.

La vidéo 9 provient aussi de AVSS 2007 et sert à évaluer le suivi de visages. Nous avons sélectionné la vidéo 2 du *Multiple faces dataset* faisant intervenir quatre personnes se déplaçant et tournant leur tête au cours de la séquence. Cette vidéo provient des travaux de E. MAGGIO et al. 2007 et est accessible à la même adresse internet que la vidéo 7.

La table 2.2 résume les différentes difficultés rencontrées sur les vidéos la base de test

construite.

vidéo	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
variation de l'illumination	*	*					**	*	*
variation de la taille des détections	*	*			*	*	**	*	*
uniformité des vêtements	*	*	**	**	*	*	*	*	
changements de direction	**		*	*					**
durée de perte de détection	**		*	*	*		**	**	*

TABLE 2.2. Résumé des caractéristiques des différentes vidéos. Le nombre d'étoiles représente l'importance des caractéristiques décrites à la première colonne.

Afin de construire une base de test, les visages de face ont été extraits par un classifieur en cascade se basant sur les descripteurs LBP (*Local Binary Pattern*). L'apprentissage hors ligne de ce classifieur a été fourni par la bibliothèque C++ OpenCV. Il a été choisi pour sa libre disponibilité, pour ses bonnes performances et sa rapidité. D'autres apprentissages, distribués dans OpenCV, du classifieur en cascade issu des travaux de P. VIOLA et M. JONES 2001 ont aussi été testés. Le détecteur étant légèrement plus lents que celui basé sur les LBP, ils n'ont pas été retenus principalement pour des raisons de temps de calcul. Le paramétrage du détecteur de visages (nombre d'échelles, facteur du changement d'échelle et amplitude des tailles des détections) a été fixé empiriquement sur chacune des vidéos afin de limiter les faux positifs et de pouvoir s'adapter à la variation des tailles des détections de chaque situation. Cela souligne un des problèmes rencontrés avec le détecteur : comment adapter automatiquement les échelles de détections en fonction des situations rencontrées ? Comme le sujet de la thèse ne porte pas directement sur le détecteur, mais qu'il est utilisé comme point d'entrée, nous nous sommes satisfaits de réglages manuels du détecteur pour construire la base de tests.

Pour chaque vidéo, les détections de visages étant extraites des images, le tri des visages a ensuite été effectué manuellement. Ainsi une vérité-terrain du partitionnement des détections de visages par identité a été construite, et les faux positifs ont fait l'objet d'une classe particulière.

2.6. Conclusion

L'état de l'art sur les techniques d'identification faciale basées vidéo permet de se rendre compte qu'il est difficile d'employer la reconnaissance faciale dans le cadre de la vidéosurveillance non-contrôlée. C'est pourquoi les travaux se sont orientés vers les méthodes de suivi visuel.

Cherchant à regrouper des images de visages, l'utilisation d'un détecteur de visages est incontournable. Cela nous a amené à nous pencher sur les méthodes de suivi basées détections. Le fait que l'on suppose avoir à disposition l'ensemble de la vidéo, les approches globales semblent les plus adaptées. Face à la complexité calculatoire intervenant dans le suivi global, une méthode de résolution déterministe basée sur la circulation de flot sur un graphe a été retenue.

Le partitionnement de données propose aussi des solutions intéressantes pour notre cadre applicatif. Les algorithmes standards de partitionnement ont servi de base pour les expérimentations menées. Deux autres approches issues du partitionnement de données permettent

aussi d'améliorer la qualité des regroupements. Elles ont été employées pour améliorer la mesure de similarité entre détections.

Il existe plusieurs manières de mesurer la qualité des résultats obtenus suivant que l'on accorde plus d'importance à la précision du suivi ou à la conservation des identités. De par l'application à la construction d'album photo, les évaluations présentées par la suite de ce rapport ont principalement utilisé la mesure indiquant la qualité des regroupements. Ainsi la F-pureté est employée pour évaluer la qualité du suivi des visages. La proportion de doublons et de personnes retrouvées ont servi à évaluer la qualité des albums photo construits à partir des regroupements de détections.

Comme il n'existe pas de base de test traitant les situations de vidéosurveillance non-contrôlée, avec des tailles de visages petites et des scènes denses, une base de neuf séquences présentant différents scénarios a été mise en place. Pour ces neuf vidéos, les visages ont été détectés automatiquement et triés à la main pour avoir une vérité terrain des regroupements. Quatre de ces vidéos proviennent du projet BioRafale, trois autres ont été tournées à l'Institut Pascal et les deux restantes viennent de travaux extérieurs. Cette base de test est citée tout au long de ce rapport car des expérimentations sont présentées pour justifier ce qui est exposé.

Chapitre 3.

Modélisation et algorithme de résolution pour le suivi multi-cibles global basé détections

De manière générale, la problématique consiste à regrouper les détections provenant d'une vidéo quelconque pour aboutir à un suivi des différents objets détectés. L'étude s'est concentrée sur une approche globale : toutes les détections de la vidéo sont supposées accessibles lors du regroupement. La méthodologie présentée dans ce chapitre peut s'appliquer à tous types d'objets, tant qu'il est possible d'avoir un détecteur (plus ou moins fiable) permettant de localiser un objet sur une image de la vidéo. Cependant, certains choix ont été justifiés par des expérimentations dans le cadre du suivi de visages. Celles-ci sont présentées dans ce chapitre pour montrer, par l'expérience, l'impact de certains éléments de la modélisation. Elles font intervenir les vidéos et les mesures présentées dans le chapitre précédent.

Ce chapitre décrit la définition du problème en termes probabilistes et un algorithme qui permet d'en trouver une solution optimale.

La section 3.1 expose la modélisation qui a servi de base aux travaux présentés, elle s'appuie sur les principes de l'estimation du Maximum *A Posteriori* (MAP).

Après avoir défini le problème, la recherche d'une solution optimale cache encore une complexité combinatoire importante. Face à cette difficulté, nous présentons (section 3.2) une méthode de résolution et expliquons comment elle permet de trouver une solution optimale (au sens du Maximum *A Posteriori* défini). Cette résolution s'appuie sur des travaux antérieurs, auxquels des contributions ont été apportées. La première permet l'obtention d'une solution optimale sans avoir à démontrer une convexité de la fonction de coût. La deuxième contribution étend la résolution à d'autres modélisations possibles de l'*a priori*.

En dernier lieu, une version séquentielle de la méthode est présentée. Elle permet d'envisager le traitement de vidéos plus longues et une utilisation sur un flux vidéo qui ne suppose pas un accès direct à la vidéo dans sa globalité.

3.1. Modélisation probabiliste

Cette section présente la modélisation mise en place pour traiter le problème du regroupement de visages détectés dans une vidéo. Le principe du Maximum *A Posteriori* appliqué à notre problématique est premièrement présenté, puis les termes d'*a priori* et de vraisemblance sont introduits plus en détails. Un autre principe (*Minimal Description Length*) a aussi été envisagé, mais n'a pas été détaillé ici, l'annexe A.1 donne un aperçu de cette représentation.

3.1.1. Maximum *a posteriori*

Pour modéliser le problème de la recherche d'un bon partitionnement des détections, nous avons décidé de nous placer dans le cadre probabiliste. Étant donnée la complexité de la solution recherchée et la difficulté à extraire des lois de probabilité réalistes sur les différentes grandeurs observées, il est vrai que le choix du cadre probabiliste peut paraître discutable. Il serait possible de rester uniquement dans le cadre de l'optimisation combinatoire, avec une fonction de coût définie empiriquement, sans interprétation probabiliste. Dans notre cas, l'intérêt de l'approche probabiliste est principalement sa facilité de compréhension et d'interprétation. Une autre raison de ce choix provient du fait que les approches probabilistes sont très largement utilisées dans le domaine du suivi multi-cibles et aussi dans certaines problématiques de partitionnement.

L'estimateur du Maximum *A Posteriori* (MAP) semble être une modélisation intéressante, car il permet de faire intervenir un *a priori* sur le modèle, ce que ne ferait pas, par exemple, un estimateur du maximum de vraisemblance.

En notant Z l'ensemble des observations et T le modèle recherché, à l'aide de la règle de Bayes, le MAP peut s'écrire :

$$\arg\max_T P(T|Z) = \arg\max_T P(Z|T)P(T) \tag{3.1}$$

En se plaçant dans le cadre du MAP, deux probabilités sont à définir : l'*a priori* d'un partitionnement $P(T)$ et la vraisemblance des détections par rapport à un partitionnement $P(Z|T)$. L'*a priori* caractérise le fait qu'un partitionnement des détections soit un bon partitionnement indépendamment de ce qui pourrait être observé aux détections. La vraisemblance définit la capacité du modèle T à expliquer les observations, elle mesure la proximité du modèle aux observations. Ainsi, le regroupement des détections considéré comme optimal sera celui qui maximisera l'*a posteriori* (*ie* maximisera conjointement l'*a priori* et la vraisemblance).

Pour appliquer ce principe à notre problème de regroupement de détections, la première étape consiste à définir ce qu'est le modèle T recherché et ce que sont les observations Z.

Étant donné que l'on suppose avoir accès à la vidéo dans sa globalité, on peut considérer que l'ensemble des détections constitue les observations et que l'on recherche un regroupement de détections. Comme le modèle ne peut contenir les observations, les détections vont être indexées et le modèle sera une partition de ces index.

Ainsi, $T \doteq \{T_1, ..., T_K, T_{FP}\}$ désignera une partition des D détections, où T_i est la i-ième partie, $K \doteq |T|$ désigne le nombre de parties et T_{FP} est la partie regroupant les détections considérées comme fausses positives. Cette dernière partie sera omise quand il n'y aura pas de gestion particulière des faux positifs du détecteur.

En pratique, une trajectoire n'est pas qu'un simple groupe de détections. Elle fait intervenir le temps, ou au moins un ordre temporel. En supposant connu l'ordre d'apparition des détections au cours de la vidéo, une trajectoire k peut être assimilée à un groupe T_k de détections ordonnées temporellement.

Une trajectoire T_k de T sera notée comme une suite $(T_k^i)_{i\in\{1,...,|T_k|\}}$ de détections, où T_k^i est élément de $\{1, .., D\}$ et permet d'identifier une détection. Plus formellement, une trajectoire T_k vérifie : $\Rightarrow d_{T_k^i} < d_{T_k^{i+1}}\ \forall i \in \{1, ..., D-1\}$ où d_n représente la date de la détection n. Cela impose qu'une trajectoire ne peut faire intervenir deux détections d'un même temps (*ie*

d'une même image de la vidéo), ce qui est cohérent avec le fait que les détecteurs donnent très rarement plusieurs détections de la même personne sur une image de la vidéo. Classiquement, les détecteurs font intervenir un système de fusion, qui permet de ne considérer qu'une seule détection quand le classifieur donne plusieurs réponses positives proches.

Finalement, si chaque groupe de T (à l'exception du groupe des faux positifs) représente une suite de détections strictement ordonnées temporellement, T_k sera appelée *trajectoire* et T *ensemble de trajectoires*. Ainsi T ne représente pas qu'un partitionnement des D détections d'une vidéo, il intègre en plus une relation d'ordre stricte définie par les temps d'apparition des détections.

Une autre interprétation possible consiste à utiliser un graphe orienté représentant les liens possibles entre les détections. On peut noter que, plus formellement, ce graphe peut être vu comme la fermeture transitive d'un graphe représentant la relation d'ordre temporelle définie sur les détections. Une solution T peut alors être vue comme un ensemble de chemins disjoints (n'ayant pas de nœuds en commun) dont les détections isolées (correspondant à un chemin d'un seul nœud) sont placées dans l'ensemble des faux positifs. La figure 3.1 illustre cela.

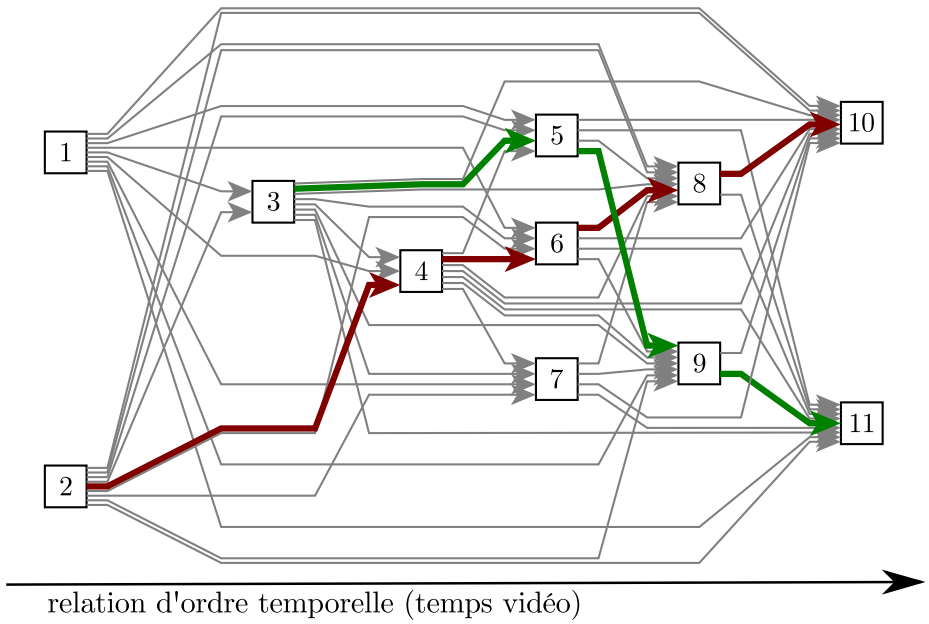

FIGURE 3.1. Illustration du graphe représentant une ensemble T de trajectoires. Les nœuds (carrés) représentent les détections, les arcs gris représentent tous les liens potentiels entre les détections. Cet exemple montre deux trajectoires (rouge et verte) comme des chemins disjoints sur le graphe. Les détections 1 et 7 sont considérées comme fausses positives car elles sont isolées. Ainsi $T = \{\{3, 5, 9, 11\}, \{2, 4, 6, 8, 10\}, T_{FP}\}$ où $T_{FP} = \{1, 7\}$.

Cette représentation par un graphe présente l'avantage d'être générale et de pouvoir facilement ajouter des contraintes sur le problème, en enlevant par exemple des arcs correspondant à un regroupement a priori impossible. Ces regroupements considérés comme impossibles peuvent, par exemple, être d'ordre temporel (*eg* la différence de temps entre les détections est trop élevée pour qu'un lien entre les détections puisse être envisagé) ou de vitesse (*eg* une détection en haut à gauche d'une image ne peut se retrouver en bas à droite à l'image

suivante). Ce procédé peut être considéré comme une étape de pré-traitement.

L'autre avantage est qu'elle permet de s'appuyer sur des méthodes issues de la Théorie des Graphes pour trouver une solution optimale en temps de calcul raisonnable, cette approche est explicitée à la section 3.2.

3.1.2. Définition de l'*a priori*

De manière générale, l'*a priori* du modèle caractérise le fait qu'un ensemble de trajectoires soit intrinsèquement "bon", et cela sans avoir observé les caractéristiques des détections. En pratique, la principale difficulté de la modélisation de l'*a priori* réside dans le fait de trouver un équilibre entre une description très restrictive d'un "bon" partitionnement, et une description trop lâche qui autoriserait trop de solutions. Dans le premier cas, si l'*a priori* est trop exigeant il est difficile de mettre en œuvre une méthode qui soit générale et traite un nombre varié de cas. À l'autre extrême, si aucun *a priori* n'est utilisé, la solution trouvée sera proche des observations mais représentera un ensemble de trajectoires qui peut être loin de ce qui est attendu en pratique. La définition de l'*a priori* sans perte de généralité est une tâche délicate.

En reliant la définition de l'*a priori* au problème du nombre de parties dans le cadre du partitionnement, on voit que cette même difficulté est rencontrée dans le partitionnement de données. Que ce soit dans les approches de type k-moyennes (*k-means* et *k-medoids*) ou celles de type mixture de gaussiennes, le nombre k de groupes doit être connu. La recherche du k idéal se fait, là aussi, par un compromis entre la qualité des groupes construits (vraisemblance) et le nombre de groupes (*a priori*). À titre d'exemple on peut citer les travaux de D. Pelleg, A. Moore et al. 2000 dans le cadre des algorithmes de type k-moyenne et le rapport de G. Celeux et G. Soromenho 1996 pour le cas des modèles de mélange.

Plusieurs définitions de l'*a priori* ont été envisagées. La première est une définition plutôt paramétrique, elle utilise les spécificités du problème du regroupement de détections de visages. La deuxième, appelée *a priori structurel*, ne fait pas intervenir de paramètres mais se focalise plutôt sur la pénalisation du nombre de trajectoires en s'appuyant sur les dépendances entre les détections.

3.1.2.1. *A Priori* avec paramétres

Étant donné qu'une trajectoire est considérée comme une suite de détections, il est intéressant de se pencher sur le processus de détection. Dans tous les cas, deux acteurs interviennent dans la production des détections : le détecteur et la vidéo.

Théoriquement, une absence de détection entre deux détections espacées dans le temps peut s'expliquer par deux raisons : un défaut du détecteur ou bien une occultation de l'objet à détecter.

Pour intégrer les absences de détection, plusieurs travaux de *tracking-by-detection* basés sur des détecteurs de piétons (L. Zhang et al. 2008 ; C. Huang et al. 2008) font intervenir un paramètre de précision du détecteur dans l'*a priori*. Cet *a priori* permet de favoriser les solutions qui respectent le paramètre en question. Ce paramètre correspond au nombre de fois qu'un piéton est bien détecté par rapport au nombre de fois où le piéton est visible. C'est un paramètre délicat à estimer.

Ensuite, pour pallier l'absence de détection, il faudrait intégrer les occultations entre les objets détectés. Dans le cadre du détecteur de visages, il est plus difficile d'utiliser un tel

paramètre car les apparitions de visages sont moins constantes. Dans le cas des piétons, la non-détection provient d'une occultation par un objet statique ou mobile, ou bien d'un défaut du détecteur ; pour le détecteur de visage il faut rajouter les non-détections dues à la rotation de la tête. Ainsi, agrémenter l'*a priori* d'un taux d'erreur d'un détecteur de visage ou une stratégie de gestion des occultations n'est plus très significatif.

Certains travaux (F. Septier et al. 2009 ; Q. Yu et G. Medioni 2009) font intervenir les longueurs de trajectoires pour pénaliser les trajectoires trop courtes. En pratique, les trajectoires peuvent être de longueur très différentes, tout particulièrement quand la séquence fait intervenir des personnes restant plus ou moins longtemps dans le champ de vision de la caméra. Comme les groupes peuvent être très variés en taille au sein d'une même vidéo, injecter un *a priori* sur la taille des trajectoires est risqué.

Pour ce qui est des détections fausses positives, il peut également être intéressant de pénaliser leur nombre dans l'*a priori* (comme le font Q. Yu et G. Medioni 2009) ou encore intégrer une estimation du taux de faux positifs du détecteur (L. Zhang et al. 2008).

Finalement, nous avons choisi de ne faire intervenir que deux aspects relativement simples pour modéliser l'*a priori* d'un ensemble T de trajectoires : son nombre de trajectoires K et son nombre de faux positifs $|T_{FP}|$. L'*a priori* est alors décomposé de la manière suivante :

$$P(T) = P_{start}(K)P_{fp}(|T_{FP}|) \tag{3.2}$$

où $P_{start}(K)$ représente la probabilité d'avoir *a priori* K trajectoires et $P_{fp}(k)$ la probabilité d'avoir k détections fausses positives, pour un ensemble de trajectoires T donné.

Utilisant un paramètre (noté p_e) représentant le ratio du nombre de trajectoires sur le nombre de détections, la façon la plus simple et la plus intuitive de modéliser la probabilité P_{start}, consiste à utiliser une loi binomiale de paramètre p_e. Ainsi $P_{start}(K)$ est la probabilité d'avoir exactement K trajectoires sachant que la probabilité de démarrer une trajectoire à partir d'une détection vaut p_e. L'*a priori* sur le nombre de trajectoires se définit alors de la manière suivante :

$$P_{start}(K) = \binom{D}{K} p_e^K (1-p_e)^{D-K} \tag{3.3}$$

La même approche peut être utilisée pour définir l'*a priori* concernant les faux positifs. On suppose une loi binomiale sur le nombre de faux positifs, avec pour paramètre le taux de faux positifs (β) du détecteur :

$$P_{fp}(K) = \binom{D}{k} \beta^k (1-\beta)^{D-k} \tag{3.4}$$

Cette modélisation est à prendre avec précaution, car le taux β n'est pas uniquement dépendant du détecteur lui-même : certaines vidéos font intervenir plus de faux positifs que d'autres. Le choix de β peut s'avérer difficile, car il ne peut être fixé indépendamment des vidéos. Ce terme permet tout de même d'écarter les détections trop isolées en ne les faisant pas intervenir dans les regroupements. En pratique, le taux de faux positifs (β) est sous-estimé pour éviter de traiter de bonnes détections comme fausses positives. Étant donné que l'objectif final est la construction d'un album photo (constitué d'un représentant de chaque

trajectoire), un seuil sur le nombre d'éléments dans la trajectoire (et aussi un terme de vraisemblance de détection) permet de considérer comme faux positifs les détections des petites trajectoires. C'est pour cela que le terme P_{fp} a souvent été omis dans les expérimentations.

Voici ce que nous appellerons par la suite *forme simplifiée de l'a priori* :

$$\tilde{P}_{start}(K) = p_e^K (1-p_e)^{D-K} \tag{3.5}$$
$$\tilde{P}_{fp}(k) = \beta^k (1-\beta)^{D-k} \tag{3.6}$$

Cette représentation est proche de celle utilisée par L. Zhang et al. 2008, à l'exception du terme $(1-p_e)^{D-K}$ qui n'apparait pas dans leur définition. En général le nombre de détections est grand devant le nombre de trajectoires et les valeurs de p_e deviennent négligeables. La forme simplifiée est donc quasiment identique à ce qui est fait par L. Zhang et al. 2008.

Nous montrerons dans la section 3.2 qu'il est possible d'obtenir une solution exacte (et de façon rapide) au problème du MAP défini ici. Pour le cas de la forme non simplifiée de l'*a priori* faux positifs, nous n'avons pas réussi à mettre en place de méthode exacte et rapide.

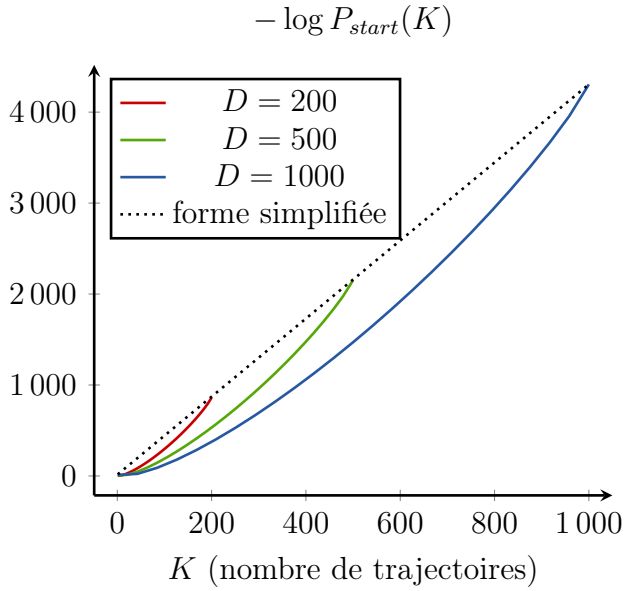

Figure 3.2. Coût issu de l'*a priori* P_{start} sur la proportion de trajectoires, il est défini par une loi binomiale et est représenté pour différents nombres de détections.

La figure 3.2 illustre le coût associé à l'*a priori* défini par une loi binomiale ainsi que sa simplification. On voit que, en utilisant le logarithme de l'*a priori*, la forme simplifiée correspond à une linéarisation de la forme binomiale.

Devant une situation réelle, le paramètre p_e n'est pas facile à fixer. Comme le montre le diagramme de la figure 3.3, le paramètre p_e conduisant à une bonne solution est parfois loin de celui correspondant à la proportion de trajectoires de la vérité-terrain. Ce diagramme nous montre aussi qu'il parait plus judicieux d'utiliser l'*a priori* de la loi binomiale plutôt que la version simplifiée, car le paramètre p_e correspond mieux à la réalité.

3.1.2.2. *A Priori* sans paramètres

Pour définir un *a priori* sur un ensemble de trajectoires sans autre information extérieure, seuls deux paramètres sont utilisables : le nombre de trajectoires et leur longueur en terme

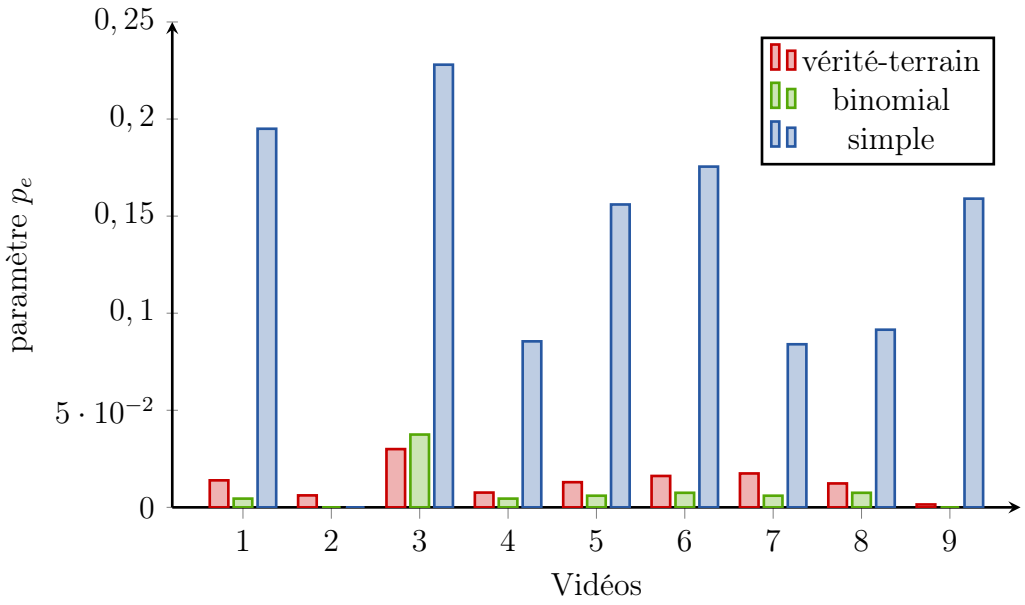

FIGURE 3.3. Valeurs du paramètre p_e conduisant aux meilleurs regroupements de détections (meilleur au sens du maximum de F-pureté avec la vérité-terrain), pour les neuf vidéos décrites à la section 2.5. *simple* : paramètre correspondant à la meilleure solution obtenue par la forme simplifiée de l'*a priori*, *binomial* : celui obtenu par P_{start} (loi binomiale) et *vérité-terrain* : proportion de trajectoires (ratio nombre de personnes sur nombre de détections) obtenue par la vérité-terrain.

de nombre de détections. Comme expliqué précédemment, les tailles des trajectoires pouvant être délicates à prendre en compte étant donné qu'elles peuvent être très variables, il ne reste plus que le nombre de trajectoires à disposition pour définir l'*a priori*. Ainsi, l'*a priori* $P(T)$ peut se caractériser par un *a priori* (noté $P_{struct}(K)$) qui sera simplement le rapport entre le nombre de solutions comportant K trajectoires sur le nombre total de solutions admissibles. En faisant l'hypothèse d'une répartition uniforme des ensembles de trajectoires, cet *a priori* représente la probabilité d'avoir un ensemble T avec K trajectoires.

Pour cela, il nous faut donc dénombrer tous les ensembles de trajectoires $t \in \mathcal{T}$ qui ont $K \doteq |T|$ trajectoires ainsi que le nombre total de tous les ensembles de trajectoires $|\mathcal{T}|$.

Une première approche serait de considérer T comme un simple partitionnement d'un ensemble à D éléments et de ne pas tenir compte du graphe des dépendances entre les détections. Le nombre total de partitions possibles est donné par le nombre de Bell (noté B_D où D représente le nombre d'éléments de l'ensemble à partitionner), et le nombre de partitions comportant K parties est donné par le nombre de Stirling de seconde espèce (noté $\left\{ {D \atop K} \right\}$). Ainsi, on pourrait approcher l'*a priori* structurel par :

$$P^s_{struct}(T) = \frac{\left\{ {D \atop K} \right\}}{B_D} \propto \left\{ {D \atop K} \right\} \tag{3.7}$$

Les nombres $\left\{ {D \atop K} \right\}$ et B_D, de par leur grandeur, sont difficilement calculables en pratique, mais comme B_D est une constante du problème et que l'on ne s'intéresse qu'au maximum *a posteriori*, il n'est pas nécessaire de calculer B_D. Pour ce qui est du nombre $\left\{ {D \atop K} \right\}$, il faudrait l'approcher par son comportement asymptotique quand D devient grand et ne calculer que son logarithme. Plutôt que d'utiliser cette approche, nous avons préféré définir plus

précisément le nombre de solutions à k trajectoires parmi l'ensemble de toutes les solutions, en prenant en compte les regroupements qui peuvent être considérés comme impossibles en se basant sur le graphe des dépendances.

Nous présentons maintenant comment est estimé le nombre de solutions à K trajectoires qui sera noté $NT_K \doteq |\{t \in \mathcal{T} \ , \ |t| = K\}|$.

En premier lieu, pour déterminer un t de $\mathcal{T}$, il faut fixer les détections sur lesquelles vont démarrer les K trajectoires, ce qui nous donne $\binom{D}{K}$ choix. Ensuite, en supposant le graphe des causalités (représentant la relation d'ordre temporelle sur les détections), il faut déterminer de combien de manières différentes il est possible de choisir les arcs. Ce qui est intéressant, c'est que l'on sait déjà qu'il faut sélectionner $D - k$ arcs pour avoir k trajectoires (mais la réciproque est fausse). Ainsi, pour approcher NT_k, la solution mise en œuvre consiste à déterminer le nombre de possibilités pour le choix des arcs, ce qui équivaut au nombre de combinaisons de $D - k$ arcs parmi tous les arcs du graphe (soit : $\binom{A}{D-k}$, avec $A \doteq |\mathcal{A}|$ nombre d'arcs du graphe). En réalité, cette énumération surestime le nombre d'ensembles de trajectoires à K trajectoires, puisque la sélection de $D-k$ arcs ne conduit pas forcément à une solution valide au sens des trajectoires. Par exemple, sélectionner deux arcs qui aboutissent à la même détection ne donnera pas un ensemble de trajectoires au sens où nous l'avons défini précédemment.

Finalement, la représentation sans paramètres pour représenter l'*a priori* s'écrit de la façon suivante :

$$P_{struct}(T) = \frac{\binom{D}{K}\binom{A}{D-K}}{|\mathcal{T}|} \tag{3.8}$$

où $|\mathcal{T}|$ représente le nombre d'ensembles de trajectoires possibles. Ce nombre n'est pas facilement calculable, toutefois, en utilisant la même approximation que pour le calcul de NT_K, il est possible de l'estimer l'*a priori* par :

$$P_{struct}(T) \propto \frac{\binom{D}{K}\binom{A}{D-K}}{\sum_{i=1}^{D}\binom{D}{i}\binom{A}{D-i}} \tag{3.9}$$

En réalité, comme on ne s'intéresse qu'au maximum *a posteriori* et que le nombre $|\mathcal{T}|$ est constant en fonction des solutions recherchées, il n'est pas nécessaire de calculer $|\mathcal{T}|$.

La figure 3.4 donne un aperçu de cette représentation de l'*a priori*. Au maximum A vaut $\frac{D(D-1)}{2}$, empiriquement A est de l'ordre de $\frac{1}{30}$ à $\frac{1}{60}$ de ce maximum. En comparant les courbes continues et discontinues, on voit que le coût s'adapte au nombre d'arcs du graphe ; quand il y a moins d'arcs, l'*a priori* va autoriser plus de trajectoires.

Pour plus de précision sur le nombre possible de solutions à K trajectoires, il peut être envisagé que le graphe utilisé pour représenter les liens potentiels entre les détections ne prenne pas que l'aspect temporel en compte, mais aussi le fait que certains liens soient impossibles car leur probabilité est trop faible.

Quelques expérimentations ont été menées pour montrer l'intérêt de l'*a priori* structurel. Avec neuf vidéos (présentées à la section 2.5), nous avons testé la qualité de l'ensemble de trajectoires obtenu en utilisant successivement les trois *a priori* : P_{start}, sa version simplifiée et P_{struct}. Les expérimentations utilisant P_{start} nécessitent de fixer le paramètre p_e qui représente la proportion de trajectoires par rapport aux détections. Pour fixer ce paramètre et pour qu'il soit le même pour tous les tests, la même moyenne de ces proportions a été prise sur les neuf vidéos.

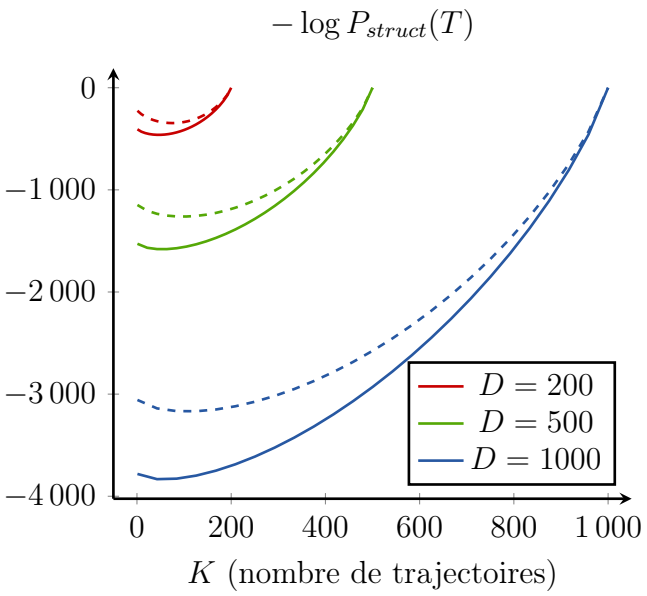

FIGURE 3.4. Coût associé à l'*a priori* structurel en fonction du nombre de trajectoires K, avec $A = \alpha\frac{D(D-1)}{2}$. Les lignes continues correspondent à $\alpha = \frac{1}{30}$ et les discontinues à $\alpha = \frac{1}{60}$.

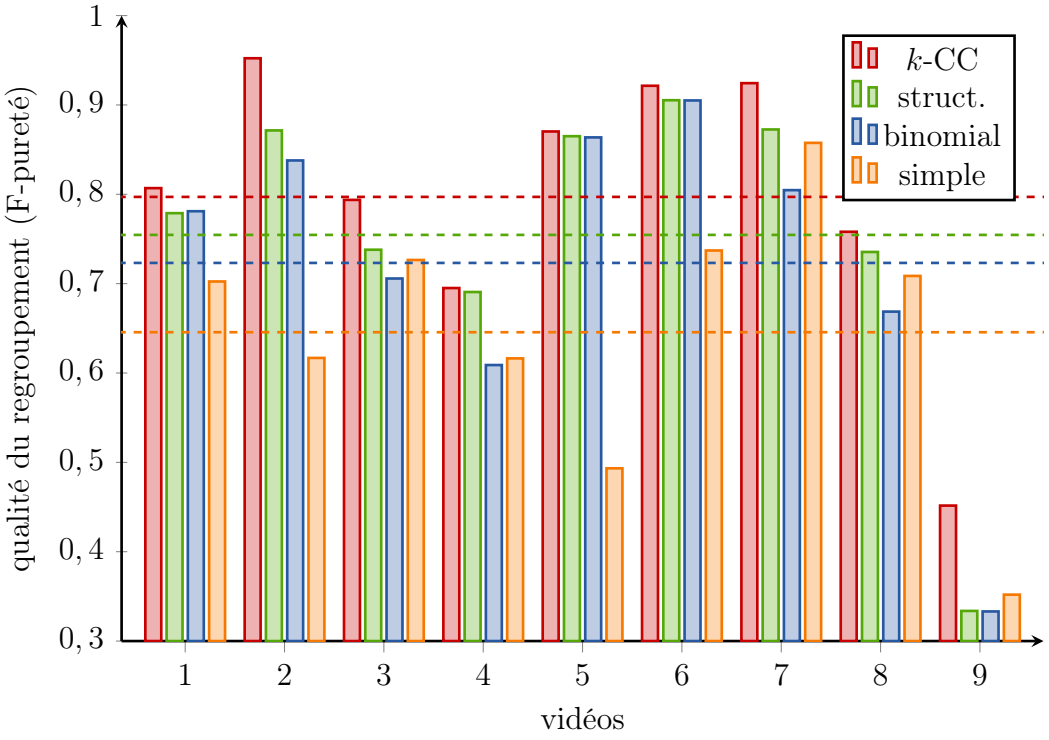

FIGURE 3.5. Comparatif des performances des différents *a priori* sur neuf vidéos. *k-CC* correspond à la meilleure solution obtenue en testant tous les nombres de trajectoires, *struct.* correspond à l'utilisation de l'*a priori* structurel, *binomial* est celui utilisant P_{start} et *simple* la version simplifiée de P_{start}. Pour *binomial* et *simple*, nous avons choisi le paramètre $p_e = 1.35\%$ correspondant à la moyenne pour les neuf vidéos des ratios nombre de trajectoires sur nombre de détections. Les lignes en pointillés représentent les moyennes des F-puretés sur les neuf vidéos pour les différents *a priori*.

La pertinence d'une solution obtenue par l'algorithme est mesurée en terme de qualité de partitionnement. Cette qualité est donnée par la F-pureté issue de la comparaison du partitionnement des détections obtenue par l'algorithme et les regroupements de la vérité-terrain. Au plus la F-pureté est proche de 1, au mieux l'algorithme a réussi à regrouper les détections. Plus de détails sur la F-pureté sont donnés à la section 2.4.3.

Les résultats de ces expérimentations sont présentés à la figure 3.5. Ce diagramme montre que, sans fixer de paramètres, l'*a priori* structurel donne une meilleure stabilité des résultats par rapport aux autres *a priori* qui sont parfois très éloignés des meilleurs solutions obtenues. De plus les paramètres p_e, correspondant aux meilleures solutions trouvées, sont bien éloignés des proportions rencontrées avec la vérité-terrain, et le choix d'un bon paramètre reste expérimental. Sur ce diagramme figure aussi la solution qui serait la meilleure en testant tous les nombres de trajectoires (de 1 à D). Pour cela l'algorithme k-CC présenté à la section 4.2.4.1 est utilisé. Cela permet de voir que dans certains cas (vidéos 4, 5 et 6) l'*a priori* structurel permet de sélectionner une solution qui est très proche de la meilleure solution que l'on puisse trouver en testant exhaustivement tous les nombres de trajectoires.

Ce résultat est à prendre avec précaution, les conclusions peuvent être différentes avec une autre fonction de similarité. La fonction de similarité correspond à une mesure de vraisemblance entre les détections, elle intervient dans la définition de la vraisemblance décrite par la suite. Dans le cadre de ces expérimentations une similarité particulière (similarité $s_{\boldsymbol{txaof}}$ équation 4.25 de la section 4.2.2) a été employée. Des tests menés avec quelques autres similarités n'ont pas été aussi concluants : cela signifie qu'il reste difficile de généraliser les performances des différentes définitions de l'*a priori*.

3.1.3. Vraisemblance

Le terme de vraisemblance ($P(Z|T)$) permet de caractériser le fait qu'un ensemble de trajectoires représente bien les observations. Ce terme correspond à ce qui est plus couramment appelé *attache aux données* dans le cadre des méthodes non probabilistes de partitionnement. Il définit l'écart entre le modèle et les données. En supposant l'indépendance entre les trajectoires, la vraisemblance peut s'écrire de la manière suivante :

$$P(Z|T) = \prod_{k=1}^{K} P_{traj}(Z|T_k) \tag{3.10}$$

où $P_{traj}(Z|T_k)$ représente la vraisemblance d'une trajectoire. L'indépendance entre les trajectoires ne parait pas déraisonnable, car les trajectoires ne s'influencent pas véritablement (si ce n'est par des occultations réciproques) et l'approche globale permet d'avoir une gestion d'ensemble sur elles.

La vraisemblance d'une trajectoire sera représentée comme une chaine de Markov :

$$P_{traj}(Z|T_k) = P_f(z_{T_k^1}) \prod_{i=2}^{|T_k|} P_{link}(z_{T_k^i}|z_{T_k^{i-1}}) P_f(z_{T_k^i}) \tag{3.11}$$

où $P_f(z_i)$ représente la vraisemblance d'une détection (*ie* la probabilité que la détection i soit bien un visage, au vu de son état z_i). Cette vraisemblance sera détaillée à la prochaine section.

Cette représentation suppose que l'observation d'une détection ne dépend que de l'observation de la détection précédente dans la trajectoire. Principalement pour des raisons de complexité calculatoire, il est difficile d'augmenter l'ordre du processus Markovien (*ie* faire dépendre une détection de ses n détections précédentes dans la trajectoire) dans le cadre d'une approche globale. C'est une pratique courante pour le suivi global, les travaux de L. Zhang et al. 2008 ; C. Huang et al. 2008 ; Q. Yu et G. Medioni 2009 ; B. Benfold et I. Reid 2011, font aussi intervenir une dépendance à l'ordre 1 entre les détections d'une trajectoire. De plus, avec des détections de visages ou de piétons, il est difficile d'utiliser un modèle d'évolution de par le caractère erratique des déplacements et de l'apparence. Cela nous a conduit à rester avec une modélisation par une chaine de Markov et à ne pas ajouter plus de précédents dans la représentation des transitions au sein d'une trajectoire.

Il est à noter que la vraisemblance décrite ici est bien différente de la plupart des vraisemblances utilisées couramment pour le partitionnement de données. Pour une trajectoire donnée, nous ne faisons finalement intervenir que les probabilités de liens entre les détections de manière à n'avoir qu'une chaine de probabilités et non pas toutes les inter-distances pouvant intervenir au sein d'une trajectoire. Nos expérimentations ont montré que cette représentation est plus adaptée à notre problématique que le sont des algorithmes plus génériques de partitionnement de données (cf 5.1).

En nous basant sur la modélisation présentée par les équations 3.10 et 3.11, il reste à définir la probabilité de transition P_{link} et la vraisemblance d'une détection P_f. La probabilité de transition est un élément crucial de la vraisemblance et de la modélisation en général, elle fait intervenir plusieurs grandeurs (le temps, la position, l'apparence). Plusieurs approches ont été mises en œuvre dans le cadre de cette thèse pour estimer au mieux cette probabilité, elles font l'objet du chapitre 4.

3.1.3.1. Vraisemblance des détections

La vraisemblance d'une détection n'est pas facile à représenter dans le cas général. Cette probabilité devrait pouvoir indiquer une confiance donnée à une détection. En d'autres termes, cette vraisemblance devrait pouvoir déterminer les chances qu'a une détection de ne pas être un faux positif du détecteur.

La couleur n'est en général pas utilisée dans l'apprentissage du détecteur et comme, historiquement, les premières approches de détections de visages se basaient sur l'extraction de couleur de peau, la segmentation de couleur de peau est une piste à envisager. L'extraction utilisée consiste simplement en une segmentation de l'espace colorimétrique issu de N. Rahman et al. 2006. Cette information a été intégrée en assimilant P_f à la proportion de pixels de couleurs de peau par rapport au nombre total de pixels de la détection.

Une autre approche consisterait à utiliser un terme de confiance qu'il est parfois possible d'obtenir avec un détecteur provenant d'apprentissage supervisé. Toutefois, on a pu observer des faux positifs ayant des termes de confiance plus élevés qu'un vrai positif, et ce paramètre ne semblait pas apporter une information suffisamment fiable.

Des expérimentations ont été menées pour juger de l'apport de la gestion des faux positifs. Cette gestion passe par deux aspects : l'*a priori* faux positifs (cf. 3.1.2) et la vraisemblance des détections. Pour rendre compte de ces deux aspects, trois cas sont comparés :

1. sans *a priori* faux positif ni vraisemblance des détections : les termes P_{fp} et P_f ont été omis
2. avec *a priori* et sans vraisemblance des détections : le terme P_f n'est pas utilisé

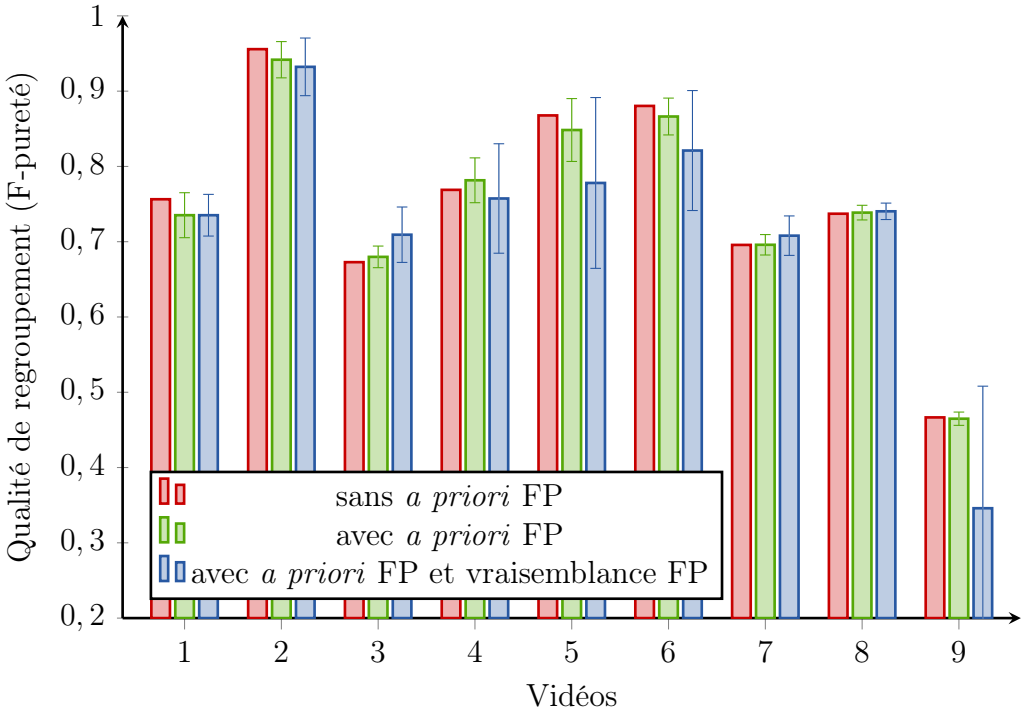

FIGURE 3.6. Apport de la gestion des faux positifs. Cas 1 : sans P_{fp} ni P_f, cas 2 : avec P_{fp} mais pas P_f et cas 3 : avec P_{fp} et P_f basé sur la proportion de couleur de peau.

3. avec *a priori* et vraisemblance des détections : le terme P_f est fixé par la proportion de pixels couleurs de peau.

Dans chacun de ces cas, le même terme de vraisemblance des liens (P_{link}) est utilisé. L'*a priori* P_{start} est pris en compte avec comme paramètre p_e celui calculé par la vérité-terrain. Pour le cas 1, l'algorithme de recherche du MAP est exécuté une seule fois alors que pour les cas 2 et 3, l'algorithme a été exécuté 100 fois avec des valeurs du paramètre β (le taux de faux positifs) allant de 10^{-6} à 0.4.

Les résultats sont représentés à la figure 3.6 en termes de qualité de regroupement des détections (en moyenne et écarts-types). Ces tests ont été menés sur les neuf vidéos présentées au chapitre 2.5. Cette expérimentation montre que la gestion des faux positifs par l'*a priori* ou la vraisemblance des détections permet dans certains cas d'améliorer la qualité des regroupements obtenus. Les termes d'*a priori* faux positifs et de vraisemblance des détections, ne donnent que des paramètres de réglages supplémentaires qui permettent d'atteindre de meilleures solutions. De manière générale, si rien ne permet de bien fixer les paramètres de faux positifs, les résultats ne sont pas forcément meilleurs avec les gestions de faux positifs proposés.

Étant donné que l'objectif final est de construire un album photo, la vraisemblance peut s'utiliser à l'étape de sélection des représentants qui constitueront l'album. C'est pourquoi, cette vraisemblance de détection n'a finalement pas été exploitée directement dans la modélisation mais plutôt à l'étape finale de la construction de l'album photo.

3.2. Recherche d'une solution au MAP

Afin de trouver une solution au problème du Maximum *A Posteriori* tel que défini à la section 3.1.1, plusieurs types d'approches peuvent être employés : des échantillonnages basés sur des méthodes de Monte Carlo (Q. Yu et G. Medioni 2009 ; B. Benfold et I. Reid 2011 ; W. Ge et R. Collins 2008), des méthodes génériques d'optimisation telles que les programmes linéaires (J. Berclaz et al. 2009), l'algorithme Hongrois (C. Huang et al. 2008) ou encore les flots de coût minimal (L. Zhang et al. 2008).

Les travaux présentés dans ce rapport se sont focalisés sur une méthode de résolution se basant sur un ensemble d'algorithmes issus de la théorie des graphes (cf. 2.2.3.1 pour les raisons de ce choix). Le lien entre la recherche du MAP et le problème du flot de coût minimal est décrit à la section 3.2.1. Nos travaux se basent sur l'idée proposée par L. Zhang et al. 2008 qui consiste à représenter le problème comme des recherches successives du flot de coût minimal sur un graphe, les détails sont donnés à la section 3.2.2. Ensuite nous expliquons comment, toujours en nous basant sur le problème du flot de coût minimal, un problème de convexité a été résolu tout en réduisant la complexité de l'algorithme.

3.2.1. Lien entre le MAP et le flot d'un graphe

En utilisant la modélisation des trajectoires par des chaines de transitions entre détections, un graphe pour représenter ces trajectoires est tout particulièrement adapté. L'ensemble $\mathcal{D}$ des détections sera représenté par un graphe ($G = (\mathcal{V}, \mathcal{A})$ avec $\mathcal{V}$ l'ensemble des nœuds et $\mathcal{A}$ l'ensemble des arcs) où chaque nœud correspond à une détection et chaque arc correspond à un lien possible entre deux détections. Étant donné qu'une détection ne peut pas être reliée à une détection passée ou une détection de la même frame, les arcs entre les nœuds correspondant à des détections sont orientés pour bien représenter les transitions. Plus mathématiquement parlant, le graphe orienté construit représente la relation d'ordre temporelle entre détections.

Ayant défini le graphe G représentant les détections, une trajectoire correspond à un chemin sur le graphe. Toutefois, si l'on veut représenter un ensemble T de trajectoires, il nous faut aussi représenter la contrainte de non-recouvrement issue de la définition d'un partitionnement (*ie* une détection ne peut appartenir à deux trajectoires). En se basant sur le graphe précédemment décrit, la contrainte de non-recouvrement peut s'exprimer de la façon suivante : un seul chemin peut passer par un nœud. Cette contrainte peut s'exprimer en terme de flot sur le graphe. En donnant des capacités binaires aux arcs, un flot $f = \{f_{ij}\}_{(i,j)\in\mathcal{A}}$ sur G se définit de la manière suivante :

$$\begin{aligned} f_{ij} &\in \{0,1\} \qquad \forall (i,j) \in \mathcal{A} \\ \sum_{(i,k)\in\mathcal{A}} f_{ik} = \sum_{(k,i)\in\mathcal{A}} f_{ki} &\leq 1 \qquad \forall i \in \mathcal{V} \end{aligned} \tag{3.12}$$

Ces contraintes permettent d'imposer qu'il ne puisse passer qu'une ou aucune unité de flot par nœud et cette unité de flot n'utilise qu'un arc pour entrer et qu'un autre pour sortir du nœud. En sélectionnant les arcs par lesquels passe une unité de flot, un ensemble de trajectoires (satisfaisant la contrainte de non-recouvrement provenant de la définition d'un partitionnement) est construit. Les nœuds au travers desquels ne circule pas de flot,

permettront de définir l'ensemble T_{FP} des détections considérées comme fausses positives. Ainsi, un flot f sur le graphe G permet de représenter un ensemble T de trajectoires basé sur l'ensemble $\mathcal{D}$ de détections.

Maintenant qu'une correspondance existe entre une ensemble T de trajectoires et un flot f sur G, il nous faut établir un lien entre l'*a posteriori* de T et un coût du flot f. En fixant des coûts $E_{ij} \doteq -\log P_{link}(z_j|z_i)$ aux arcs du graphe, la maximisation du terme de vraisemblance $P(Z|T)$ peut se rapporter à une recherche de flot de coût minimal :

$$\begin{aligned} \textbf{min} \sum_{(i,j)\in\mathcal{A}} f_{ij}E_{ij} & \\ f_{ij} \in \{0,1\} \quad & \forall (i,j) \in \mathcal{A} \\ \sum_{(i,k)\in\mathcal{A}} f_{ik} = \sum_{(k,i)\in\mathcal{A}} f_{ki} \leq 1 \quad & \forall i \in \mathcal{V} \end{aligned} \tag{3.13}$$

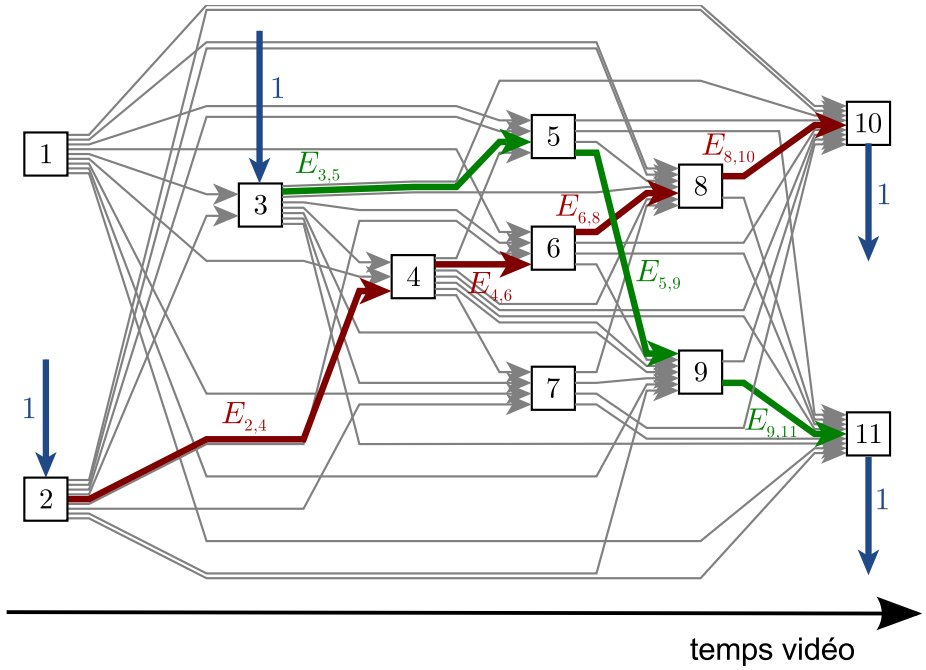

FIGURE 3.7. Illustration du graphe utilisé pour représenter la vraisemblance d'un ensemble de trajectoires. Les nœuds rectangulaires représentent les détections et les arcs montrent les liens potentiels. Les arcs bleus illustrent une demande de flot. Dans le cas illustré, comme les capacités des arcs sont 0 ou 1, le coût du flot vaut $(E_{3,5}+E_{5,9}+E_{9,11})+(E_{2,4}+E_{4,6}+E_{6,8}+E_{6,8})$. Les arcs verts et rouges représentent les deux trajectoires construites $\{3,5,9,11\}$ et $\{2,4,6,8\}$. Les détections 1 et 7 sont considérées comme fausses positives et n'interviennent pas dans la vraisemblance.

Quand le terme d'*a priori* $P(T)$ est mis de côté, le coût du flot se rapproche de la vraisemblable et ainsi le maximum de vraisemblance se trouve au minimum de coût du flot. Cela est illustré à la figure 3.7.

Cette modélisation par circulation de flot sur un graphe, permet ainsi de trouver un maximum de vraisemblance en se basant sur une recherche de flot de coût minimal, problème connu de la Théorie des Graphes.

La section suivante décrit comment les auteurs de L. Zhang et al. 2008 ont mis en œuvre cette idée et comment ils intègrent l'*a priori*.

3.2.2. L'approche de L. Zhang et al. 2008 pour la recherche du MAP

Nos travaux présentent une approche similaire à celle de L. Zhang et al. 2008, pour ce qui est de la recherche du Maximum *A Posteriori*. La principale différence est que leur méthode ne modélise pas directement l'*a priori* par le graphe et la valeur du flot (correspondant à K, le nombre de trajectoires) doit être connue pour pouvoir trouver une solution par algorithme de flot de coût minimal, alors que nous proposons une représentation qui intègre l'*a priori* dans le coût du flot.

Un point important, qui n'a pas été abordé dans la section précédente, est que le problème du flot de coût minimal nécessite une demande de flot fixée à l'avance. Sachant qu'une trajectoire peut démarrer à chaque détection, il faudrait pouvoir introduire une demande de flot à chaque détection. Pour résoudre ce problème, il est couramment introduit deux nœuds (une source s et un puits t) qui sont reliés à tous les nœuds par des arcs de capacité 1 et auxquels une offre ou une demande est ajoutée. L'ajout des deux nœuds est illustré à la figure 3.8.

En définissant le coût F comme l'opposé du logarithme de l'*a posteriori*, on peut le développer de la façon suivante :

$$\begin{aligned} F(T) &= -\log(P(T)P(Z|T)) \\ &= -\log P(T) - \sum_{i=1}^{D}\sum_{j=1}^{D} f_{ij} \log P_{link}(z_j|z_i) \end{aligned} \tag{3.14}$$

en omettant la vraisemblance d'une détection P_f et où :

$$f_{ij} = \begin{cases} 0 \text{ si il existe une transition de } i \text{ à } j \text{ dans } T \\ 1 \text{ sinon} \end{cases} \tag{3.15}$$

Pour l'*a priori* $P(T)$, les auteurs prennent :

$$P_{start}(K) = p_e^{2K} \tag{3.16}$$

$$P_{fp}(|T_{FP}|) = \beta^{|T_{FP}|}(1-\beta)^{D-|T_{FP}|} \tag{3.17}$$

ce qui est très proche de ce que nous avons appelé la forme simplifiée de l'*a priori* binomial, quand p_e est petit (ce qui est le cas en pratique). Le coût s'exprime alors de la façon suivante :

$$\begin{aligned} F(T) &= -\log P_{start}(K) - \log(P_{fp}(|T_{FP}|)) - \sum_{i=1}^{D}\sum_{j=1}^{D} f_{ij} \log P_{link}(z_j|z_i) \\ &= -\log P_{start}(K) - \sum_{i=1}^{D} f_i \log \frac{1-\beta}{\beta} \\ &\quad - \sum_{i=1}^{D}\sum_{j=1}^{D} f_{ij} \log P_{link}(z_j|z_i) + cste \end{aligned} \tag{3.18}$$

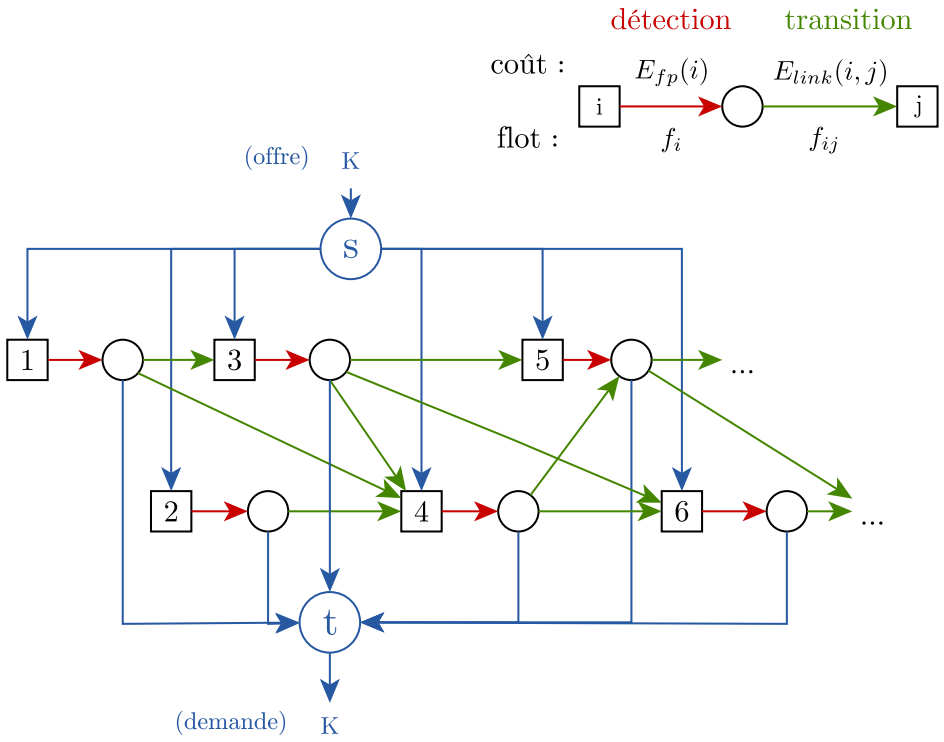

FIGURE 3.8. Illustration du graphe utilisé pour résoudre le MAP par flot de coût minimal, les six premières détections ont été représentées. Les nœuds rectangulaires représentent les détections à partitionner, les nœuds circulaires et les arcs rouges permettent d'introduire l'*a priori* P_{fp}. Les arcs verts représentent les transitions possibles pour regrouper deux détections dans une même trajectoire, ils permettent de représenter le coût de transition. Les nœuds s et t sont introduits pour imposer une demande de flot correspondant au nombre de trajectoires.

où *cste* désigne une constante ne dépendant pas de T.

Le terme d'*a priori* P_{fp} est représenté par le graphe en ajoutant un arc et un nœud à chaque nœud correspondant à une détection, ce qui permet d'ajouter un coût fixe par détection (cf. figure 3.8). Le flot passant par ces arcs est noté :

$$f_i = \begin{cases} 0 \text{ si } i \in T_{FP} \\ 1 \text{ sinon} \end{cases} \tag{3.19}$$

En prenant le graphe tel qu'il est illustré à la figure 3.8, les coûts des arcs, liés au flot f définit par les f_i et f_{ij}, sont donnés par :

$$E_{fp}(i) = -\log \frac{1-\beta}{\beta} \tag{3.20}$$

$$E_{link}(i,j) = -\log P_{link}(z_j|z_i) \tag{3.21}$$

Ainsi, le coût total d'un flot vaut :

$$F(f) = \sum_{i=1}^{D} f_i E_{fp}(i) + \sum_{i=1}^{D}\sum_{j=1}^{D} f_{ij} E_{link}(i,j) \tag{3.22}$$

et la recherche du flot de coût minimal permet de trouver :

$$\tilde{T} = \underset{\substack{f \text{ flot sur } G \\ \text{avec } K\text{unités de flot}}}{\arg\min} F(f) = \underset{\substack{T\in\mathcal{T} \\ |T|=K+1}}{\arg\min} -\log P(T)P(Z|T) \tag{3.23}$$

pour un K fixé.

Finalement, trouver le K qui permet d'atteindre le MAP nécessite une succession de recherches de flot de coût minimal. Notons f_m la suite du coût minimal en fonction du nombre de trajectoires m :

$$f_m = \min_{\substack{T\in\mathcal{T} \\ |T|=m+1}} -\log P_{fp}(T)P(Z|T) \tag{3.24}$$

Si f_m ne possède aucune propriété de convexité, il faudrait exécuter D recherches de flot de coût minimal pour trouver un MAP. Cependant, en supposant que cette suite soit uni-modale, la recherche du K permettant d'atteindre le MAP peut se faire en un maximum de $\log(D)$ itérations avec un algorithme basé sur la recherche de Fibonacci.

Ainsi, l'algorithme employé par L. Zhang et al. 2008 est une recherche de Fibonacci sur le nombre de trajectoires, où à chaque étape de la recherche un algorithme de flot de coût minimal est employé pour trouver le MAP ayant le nombre de trajectoires fixé.

Pour pouvoir dire que l'algorithme trouve bien un MAP, il faut montrer que la fonction coût f_m relative à la probabilité *a posteriori* forme une suite de m (nombre de trajectoires de T) qui soit uni-modale. Dans notre cas, comme l'objectif est de trouver un minimum de f_m, montrer que f_m est uni-modale revient à montrer qu'il existe un $m^\star \in \{1, ..., D\}$ tel que :

$$\forall i < m^\star : f_i \leq f_{i+1} \tag{3.25}$$
$$\forall j \geq m^\star : f_j \geq f_{j+1} \tag{3.26}$$

La démonstration ne semble pas aussi triviale que le suggère L. Zhang et al. 2008 page 3. Toutefois, en pratique, on observe que cette suite semble bien uni-modale, notamment en prenant comme définition de l'*a priori* celle des équations 3.16 et 3.17.

3.2.3. Méthode proposée

Afin de gérer plus efficacement le terme d'*a priori* et de contourner le problème de convexité de la méthode décrite précédemment, nous avons décidé de représenter l'*a priori* sous une forme qui permette de l'intégrer directement dans le problème du flot de coût minimal. Ainsi il n'y aurait plus besoin de lancer plusieurs fois l'algorithme du flot de coût minimal (la solution donnée par l'algorithme correspond directement à une solution du maximum *a posteriori*). La partie de l'*a priori* qui représente les faux positifs (cf. P_{fp} de l'équation 3.2) est déjà pris en compte par le graphe comme nous l'avons expliqué à la section précédente. L'objectif est alors de pouvoir exprimer $P_{start}(K)$ (*a priori* sur le nombre de trajectoires) de manière à ce qu'il soit représentable par le coût du flot sur le graphe.

En prenant la forme simplifiée de l'*a priori* des trajectoires ($\tilde{P}_{start}(K)$ cf. équation 3.5), on peut écrire son coût équivalent :

$$-\log \tilde{P}_{start}(K) = -K \log p_e - (D-K)\log(1-p_e) \tag{3.27}$$

Étant donné que la quantité de flot circulant sur le graphe correspond au nombre de trajectoires, en ajoutant un coût $E_{start} \doteq -\log(p_e)$ sur un arc entre le graphe représentant la vraisemblance et la source, le terme $-K \log p_e$ est ainsi bien pris en compte. En imposant à la source une demande de D (au lieu de K), le deuxième terme de l'*a priori* peut être intégré dans le coût du flot par un arc allant directement aux nœuds puits. La quantité de flot traversant cet arc est de $D-K$, et donc, avec un coût de $E_{nstart} \doteq -\log(1-p_e)$ sur cet arc, le coût lié à la version simplifiée de l'*a priori* binomial est bien pris en compte.

La modélisation présentée possède aussi un terme représentant la vraisemblance d'une détection prise isolément. Ce terme est en fait la probabilité qu'une détection soit bien un vrai positif. Comme décrit précédemment, l'*a priori* de faux positifs $P_{fp}(|T_{FP}|)$ est comptabilisé dans le coût par des arc internes (de flot f_i) qui permettent de sommer les coûts des détections qui ne sont pas comptés comme fausses positives. Pour intégrer la vraisemblance des détections, sur chacun de ces arcs correspondant à chaque détection i, il suffit d'ajouter un coût $E_f(i) \doteq -\log P_f(z_k)$ à $E_{fp}(i)$ défini précédemment.

En définissant les coûts sur les arcs de la façon suivante :

$$E_{start} = -\log(p_e) \tag{3.28}$$
$$E_{nstart} = -\log(1-p_e) \tag{3.29}$$
$$E_{fp} = -\log \tfrac{1-\beta}{\beta} \tag{3.30}$$
$$E_f(k) = -\log P_f(z_k) \tag{3.31}$$
$$E_{link}(k,l) = -\log P_{link}(z_l|z_k) \tag{3.32}$$

la fonction objectif rattachée au MAP s'écrit de la façon suivante :

$$\tilde{F}(T) = f_s E_{start} + (D-f_s)E_{nstart} + \sum_{i=1}^{D} f_i(E_f(i)+E_{fp}) + \sum_{i=1}^{D}\sum_{j=1}^{D} f_{ij}E_{link}(i,j) \tag{3.33}$$

où f représente le flot :
$f_s = K$: flot entrant dans la partie *vraisemblance* du graphe
$f_i = 0$: i est considéré comme faux positif
$f_i = 1$: i est considéré comme vrai positif
f_{ij} : flot entre la détection i et la détection j

La figure 3.9 décrit le graphe permettant de trouver un MAP par la résolution d'un unique problème de flot de coût minimal.

L'utilisation de la version non simplifiée de l'*a priori* P_{start} nécessite l'ajout du coût $-\log \binom{D}{K}$ sur le graphe. Le problème de ce coût est qu'il n'est pas linéaire en fonction du nombre K de trajectoires et ne peut donc pas être intégré dans E_{start} ni dans E_{nstart}. Il est toutefois possible de représenter ce coût. Cela demande l'ajout d'arcs et de nœuds comme le décrit l'annexe B.1.

Ce même problème interviendra si l'on veut passer à la version non simplifiée de l'*a priori* P_{fp}, mais l'intégration du coefficient binomial prenant en compte les nombres de faux positifs est plus difficile. Comme ce nombre n'est pas lié au flot sur le graphe, il n'a pas pu être intégré directement dans le coût.

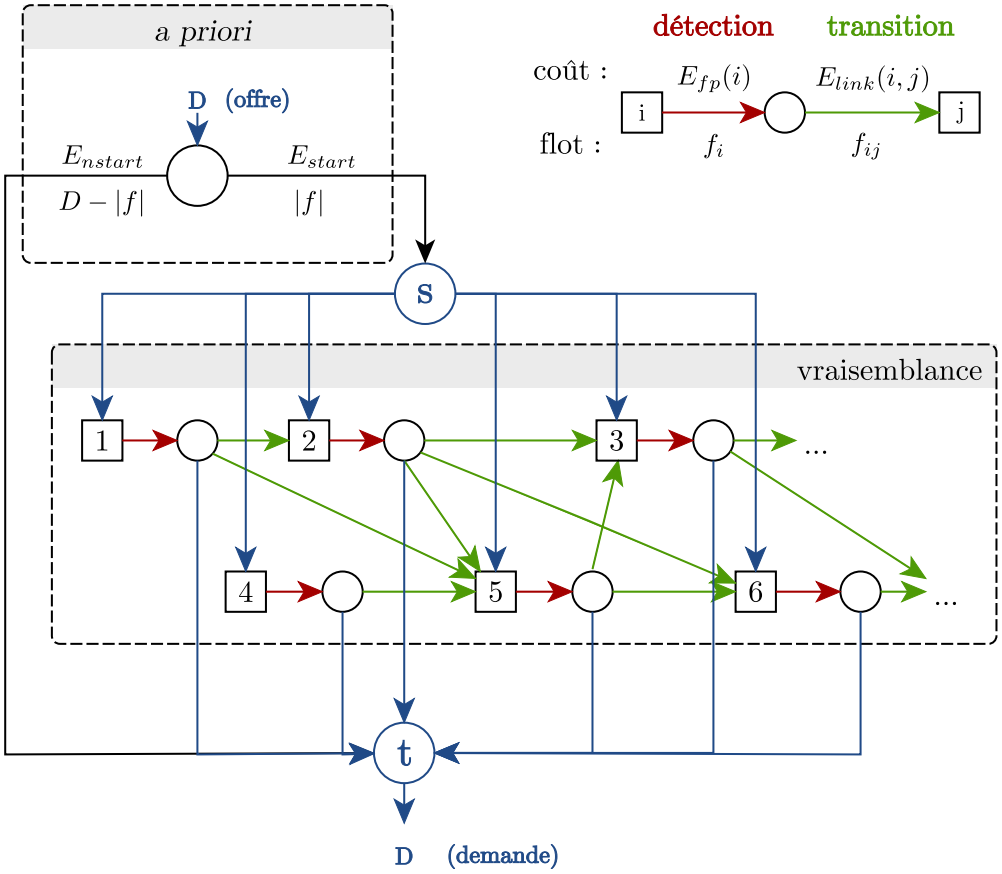

FIGURE 3.9. Illustration du graphe utilisé pour résoudre le MAP par flot de coût minimal avec gestion directe de l'*a priori*. Grâce à la linéarisation du coût lié à l'*a priori* P_{start}, il est possible de trouver le MAP en résolvant un seul flot de coût minimal.

3.2.4. Comparatif des temps de calcul

En plus de sa capacité à trouver une solution optimale sans avoir recours à une recherche de Fibonacci, la méthode que nous proposons pour gérer l'*a priori* directement permet aussi de gagner en temps de calcul. Afin de montrer cela, nous avons exécuté quatre versions de l'algorithme permettant de regrouper les trajectoires. Les performances sont données à la figure 3.10. Pour les quatre versions, nous avons comptabilisé uniquement le temps de calcul utilisé pour regrouper les détections avec une matrice des similarités inter-détections déjà calculées. Les neuf vidéos utilisées pour les tests correspondent à celles présentées au chapitre 5. Elles ont un nombre de détections allant de 200 à 2700. Pour chaque vidéo, chaque algorithme est exécuté 10 fois avec des valeurs de p_e différentes (sauf pour la version *struct.* qui ne fait pas intervenir ce paramètre), la moyenne et l'écart-type sont représentés. Tous les tests ont été menés avec la même implémentation de la recherche du flot de coût minimal, à savoir l'algorithme *Capacity Scaling* de la bibliothèque C++ LEMON [1]. Cet algorithme se base sur les travaux de J. EDMONDS et R. KARP 1972 recensés notamment par R. AHUJA et al. 1993. Les temps présentés sont ceux obtenus sur un PC de bureau standard avec une fréquence de processeur de 2,4 GHz et sans parallélisation particulière.

La première version (nommée *fibonacci*) correspond à l'approche utilisant une recherche de Fibonacci pour trouver le nombre de trajectoires qui maximisera l'*a posteriori*. Elle

1. http ://lemon.cs.elte.hu

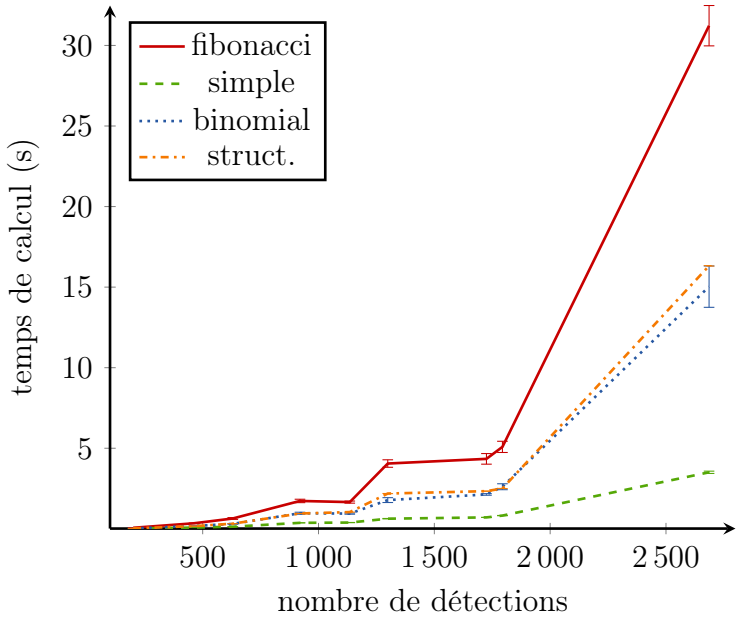

FIGURE 3.10. Temps de calcul du regroupement des détections par flot de coût minimal. *fibonacci* : méthode faisant intervenir une recherche de Fibonacci sur le nombre de trajectoires de la solution avec la version simplifiée de l'*a priori*, *simple* : méthode proposée avec la version simplifiée de l'*a priori*, *binomal* : méthode proposée avec l'*a priori* P_{start} binomial et *struct.* : méthode proposée avec l'*a priori* sans paramètre.

correspond à celle présentée par L. ZHANG et al. 2008 et emploie la version simplifiée de l'*a priori*. Les trois autres versions (*simple*, *binomial* et *struct.*) correspondent à la méthode que nous présentons, où l'*a priori* est directement intégré dans la représentation par flot de coût minimal. Pour le cas *simple*, l'*a priori* utilisé est celui ne faisant pas intervenir de coefficient binomial, alors que la version *binom* est celle utilisant l'*a priori* binomial $P_{start}(K)$ et un ajout de nœuds permettant de représenter le logarithme du coefficient binomial (cf annexe B.1). La dernière version (*struct.*) utilise la version structurelle de l'*a priori* qui nécessite de représenter deux coefficients binomiaux (cf section 3.1.2.2). L'annexe B.2 montre comment intégrer l'*a priori* P^s_{struct} défini comme étant proportionnel à $\left\{ {D \atop K} \right\}$.

Les temps de calcul de la figure 3.10 permettent de comparer ces quatre versions de l'algorithme de regroupement. Les résultats montrent que la méthode que nous employons pour représenter l'*a priori* directement dans le problème de flot de coût minimal permet de gagner en temps de calcul. La version *fibonacci* demande plus de temps de calcul que les autres versions, d'autant plus quand le nombre de détections augmente. Les versions prenant en compte des coefficients binomiaux dans l'*a priori* sont plus rapides que celle utilisant une recherche de Fibonacci, mais restent toutefois plus lentes que la version *simple* qui donne les résultats les plus intéressants. Sur le graphique figurent les écarts-types des temps de calcul estimés. Il est donc possible de voir que le paramètre p_e impacte le temps de calcul, notamment pour les versions *binomial* et *fibonacci*.

3.3. Version séquentielle

Une extension séquentielle de la méthodologie décrite jusqu'ici est maintenant présentée. Les principaux problèmes liés à une approche globale pour le regroupement des détections

sont les suivants :

— **la complexité combinatoire** : sur des cas réels de longues vidéos, le traitement de l'ensemble des détections fait intervenir des temps de calcul qui peuvent être rédhibitoires.

— **le traitement d'un flux** : l'approche globale nécessite l'acquisition de toute la vidéo avant de pouvoir donner un résultat de regroupement. Dans de nombreux cas d'utilisation, il peut être nécessaire d'avoir une analyse temps réel afin de pouvoir directement intervenir sur l'évènement. Cela n'est pas envisagé directement par la méthode globale.

Afin de pallier ces deux limitations, il est possible d'utiliser un découpage de la vidéo afin de traiter chaque partie globalement et de fusionner ensuite les résultats de chaque partie. Cette section décrit une solution mise en œuvre pour traiter une vidéo de façon séquentielle, en la divisant en sous-séquences disjointes.

3.3.1. Algorithme

Dans le cadre de l'analyse de films cinématographiques ou de vidéos issues de la télévision, il peut être intéressant de ne pas forcer un découpage en séquences de tailles fixes, mais de faire coïncider les coupes avec des transitions naturelles de la vidéo (typiquement les changements de plans). Pour ce qui est de la vidéosurveillance, n'ayant pas de découpage "naturel" de la vidéo, nous avons utilisé un découpage de la vidéo en séquences d'images consécutives de tailles fixes. Une autre solution pourrait consister en un découpage qui construirait des séquences ayant un nombre de détections fixé. Cette solution n'a pas été mise en œuvre dans le cadre de cette thèse.

Pour chaque sous-séquence de la vidéo, l'algorithme décrit au chapitre précédent peut être utilisé pour regrouper les détections de visages. Ensuite, ce même algorithme sera réutilisé non plus pour regrouper directement les détections, mais pour regrouper les différents groupes déjà créés sur chaque sous-séquence. À ce niveau, deux stratégies ont été envisagées. La première tient compte d'un regroupement issu de la sous-séquence précédente pour le fusionner avec le regroupement de la sous-séquence courante. La seconde stratégie consiste à effectuer en dernière étape un regroupement global des groupes créés à chaque sous-séquence (cf. section 3.3.2).

Dans le premier cas, on procède à un partitionnement prenant en compte les groupes créés à la sous-séquence courante et ceux intervenant à la sous-séquence suivante (cf figure 3.11). Ce procédé est décrit à l'algorithme 1.

Algorithme 1 : Méthode de traitement par sous-séquences.

pour *toutes les sous-séquences de la vidéo* **faire**
  détecter et extraire les caractéristiques des détections de la séquence courante;
  partitionnement intra-séquence :
  résoudre le MAP sur la sous-séquence courante;
  mettre à jour les dissimilarités en intégrant les nouvelles trajectoires;
  mettre à jour les paramètres Pe_I et β_I;
  partitionnement inter-séquences :
  résoudre le MAP avec les trajectoires de la séquence précédente;
fin

Le *partitionnement intra-séquence* correspond à la méthode présentée précédemment, non plus appliquée à toute la vidéo mais en se concentrant sur un seul paquet d'images. Pour ce qui est du *partitionnement inter-séquences*, la même méthode est utilisée à la différence que les observations $(\boldsymbol{z}_i)$ ne sont plus de simples détections mais peuvent être des groupes de détections construits à l'étape précédente. Ce partitionnement découle des groupes issus du *partitionnement intra-séquence* sur la sous-séquence courante et des groupes ayant au moins un élément dans la séquence précédente (voir figure 3.11). Cela permet de ne pas perdre la continuité entre deux sous-séquences successives. Dans le cas du partitionnement intra-séquence, les paramètres β et p_e sont fixés empiriquement. Pour ce qui est du partitionnement inter-séquences, ces paramètres (notés Pe_I et β_I) sont ré-estimés à chaque itération :

$$Pe_I = \frac{\text{nb personnes}}{\text{nb groupes actifs}} = p_e \frac{\text{nb détections}}{\text{nb groupes actifs}} \tag{3.34}$$

car p_e est estimé par $\frac{\text{nb personnes}}{\text{nb détections}}$. Le terme *groupes actifs* désigne les regroupements issus du partitionnement intra-séquences ou issus des partitionnements précédents dont au moins une détection figure dans la séquence de l'itération précédente (cf figure 3.11). Le paramètre du taux de faux positifs est estimé comme suit :

$$\beta_I = \frac{\text{nb groupes FP}}{\text{nb groupes actifs}} = p_e \frac{\text{nb détections FP}}{\text{nb détections}} = \beta Pe_I \tag{3.35}$$

avec β estimé par $\frac{\text{nb détections FP}}{\text{nb détections}}$ (où FP signifie Faux Positifs) et en approximant $\frac{\text{nb groupes FP}}{\text{nb détections FP}}$ par p_e.

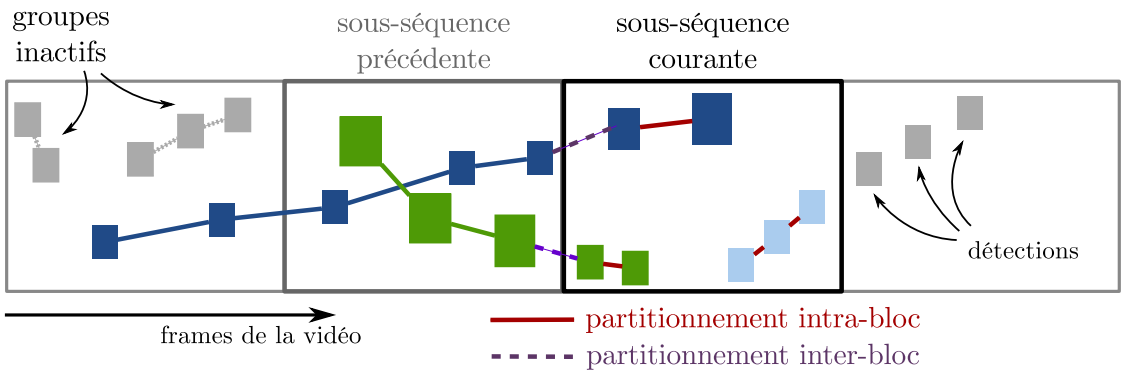

FIGURE 3.11. Schéma illustrant les sous-séquences d'images utilisées dans le traitement par blocs.

Ayant décrit la méthode, il nous reste à définir les dissimilarités entre deux groupes de détections. En utilisant toujours les distances de Hellinger déjà définies, les similarités entre deux groupes peuvent s'estimer par une moyenne des similarités se situant aux nouvelles transitions. Ces nouvelles transitions sont définies comme les transitions supplémentaires qui apparaitraient à la fusion des deux groupes (cf schéma figure 3.12).

Cette méthode permet d'utiliser les similarités définies précédemment et d'être assez robuste aux cas des groupes de tailles différentes.

3.3.2. Variante hiérarchique

Plusieurs travaux (C. HUANG et al. 2008 ; A. A. PERERA et al. 2006 ; J. HENRIQUES et al. 2011) se basent sur la construction de *tracklets* pour ne pas traiter directement l'ensemble

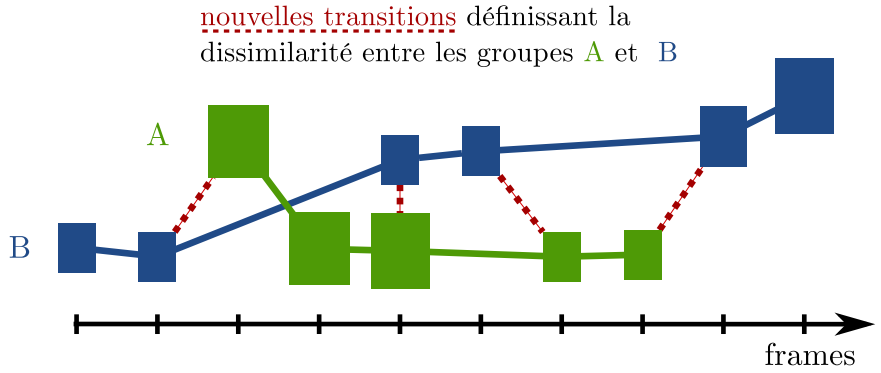

FIGURE 3.12. Schéma illustrant les dissimilarités entre groupes de détections.

des détections, mais effectuer un premier regroupement liant les détections très proches et ne présentant pas d'ambiguïté. Cela permet de diminuer considérablement la complexité combinatoire du problème tout en permettant de travailler sur des objets plus complexes (*tracklets*) que de simples détections et ainsi utiliser plus d'information pour l'association. Toutefois, cela implique une confiance importante au premier regroupement.

Utilisant cette idée de construction de *tracklets*, nous avons mis en œuvre une méthode qui est proche de celle présentée à la section précédente, mais qui ne traite pas la vidéo par sous-séquences d'images. Plus précisément, l'algorithme de regroupement des détections est employé une première fois pour construire des petites trajectoires puis les similarités entre ces *tracklets* sont ré-estimées et une nouvelle fois l'algorithme de regroupement est exécuté.

3.4. Conclusion

Ce chapitre a présenté le modèle probabiliste mis en œuvre pour définir le problème du regroupement des détections et traiter le suivi visuel. Le maximum *a posteriori* défini offre deux possibilités pour préciser l'*a priori*. La première fait intervenir un paramètre lié au détecteur qui permet d'influer sur le nombre de trajectoires. La seconde ne prend en compte aucun paramètre et est utile dans le cas où aucun *a priori* sur le paramétrage n'est disponible.

Un terme de vraisemblance de détection a aussi été intégré au modèle et à la résolution, mais il reste délicat à définir et dépend de l'application visée.

L'algorithme déterministe de recherche du maximum *a posteriori* présenté permet d'obtenir une solution optimale rapidement. En étendant des travaux antérieurs, le partitionnement optimal est obtenu par une seule recherche de flot de coût minimal et nous avons montré comment intégrer les deux modélisations proposées pour l'*a priori*.

Se basant sur la modélisation et l'algorithme de résolution définis, des stratégies de traitement séquentiel ont aussi été introduites. Elles permettent une subdivision du problème en sous-séquences, ce qui engendre un gain en temps de calcul et autorise une extension au cas des flux vidéos.

Chapitre 4.

Vraisemblance et agrégation des informations

Après avoir décrit, au chapitre précédent, la modélisation utilisée pour représenter de manière probabiliste le problème du regroupement des détections issues d'une vidéo, ce chapitre présente les différentes similarités employées pour caractériser l'éloignement d'une détection par rapport à une autre. Ces similarités sont ensuite employées pour calculer le terme de vraisemblance d'un regroupement de détections.

Tout au long de ce chapitre, des expérimentations sont présentées. C'est notamment pour cela que la base de test des neuf vidéos et une méthode de mesure de qualité ont été présentées au chapitre 2. Ces expérimentations permettent de justifier les choix présentés dans ce chapitre.

Dans le cadre du partitionnement de données, les similarités (ou dissimilarités) entre objets jouent un rôle prépondérant, et c'est pourquoi nous y accordons une attention particulière. La définition de telles similarités nécessite la sélection d'informations permettant de décrire au mieux les détections et ensuite la fusion de ces informations.

La première partie du chapitre traite des différentes descriptions envisagées et utilisées ainsi que les mesures employées pour les comparer.

La seconde partie expose une représentation des données permettant de fusionner les descriptions ainsi qu'une méthode d'estimation des similarités pour agréger les différentes mesures.

4.1. Description et similarités des détections

Pour rendre compte des liens entre les détections, il est important de sélectionner les informations utiles pour les décrire et aussi de définir une distance entre ces descriptions. Utilisant de simples vidéos, il est possible d'extraire trois types d'information à chaque détection :

- temporelle : même si la date absolue de chaque image de la vidéo peut être inconnue, il est a minima possible d'avoir un index de l'image en question au sein de la vidéo.
- colorimétrique : l'apparence d'une personne traversant la scène filmée peut être très variable, toutefois, la couleur des vêtements ou encore celle des cheveux reste relativement stable.
- spatiale : là encore, la position absolue dans la scène observée d'un pixel d'une image n'est souvent pas accessible (à moins d'avoir une calibration de la caméra et de pouvoir reconstruire la scène), toutefois la position de la détection dans l'image est accessible.

Ces informations ne sont pas absolues et, idéalement, il faudrait avoir accès à une calibration complète du système d'acquisition pour accéder à des informations plus réalistes. Pour ne pas perdre en généralité, nous supposons n'avoir accès à aucune calibration. Même si les information extraites ne sont pas absolues, elles restent toutes primordiales pour décrire les détections.

Comme expliqué au chapitre précédent, nous nous sommes placés dans un cadre probabiliste pour exposer le problème. La définition du terme de vraisemblance $P(Z|T)$ (où Z représente les observations et T l'ensemble des regroupements de détections) passe par l'écriture du terme $P_{link}(\boldsymbol{z}_j|\boldsymbol{z}_i)$ qui décrit la probabilité de passer de l'observation de la détection i à celle de la détection j. Pour définir cette probabilité de transition, il nous faut aussi définir comment est décrite l'information $\boldsymbol{z}_i$ extraite de la détection i.

Les descriptions attachées aux détections sont représentées par des distributions, et la similarité entre les détections est exprimée par une mesure comparant des densités de probabilités. Notre choix s'est porté sur la distance de Hellinger qui se base sur le coefficient de Bhattacharyya et en fait une distance. Comme cette distance est dans $[0, 1]$, la similarité entre deux mêmes caractéristiques (notée x) issues de deux détections (i et j) est définie comme :

$$s_x(i, j) = 1 - D_H(p_x^i, p_x^i) \tag{4.1}$$

où $p_x^i : x \in \mathcal{X} \rightarrow [0, 1]$ représente la densité liée à la probabilité d'avoir x comme caractéristique pour la détection i.

Cette similarité n'est pas forcément une probabilité de i, il n'est donc pas directement possible de l'assimiler à $P_{link}(.|\boldsymbol{z}_i)$. La section 4.2.1 décrit ce que nous avons envisagé pour définir une probabilité à partir de ces similarités.

4.1.1. Aspect temporel

Afin d'intégrer la notion de temps dans la vraisemblance d'une suite de détections, il nous faut estimer la probabilité que deux détections apparaissant à des temps différents se suivent dans une trajectoire. En pratique, en ayant à faire à des vidéos au taux d'images par seconde fixe, il n'est pas déraisonnable de représenter le temps comme un index de frame entier.

Plusieurs façons de représenter la vraisemblance temporelle d'un lien entre deux détections peuvent être mises en avant.

Les deux premières représentent la vraisemblance comme la probabilité $P_{time}(t|t_i)$ que la détection i (apparaissant au temps t_i) soit directement rattachée à une détection apparaissant au temps t.

La manière la plus simple est de prendre une loi uniforme avec un écart temporel maximum fixé (t_{max}) :

$$P_{time}^u(t|t_i) = \begin{cases} \frac{1}{\Delta t_{max}} & \text{si } 0 < t - t_i \leq \Delta t_{max} \\ 0 & \text{sinon} \end{cases} \tag{4.2}$$

Cette loi représente relativement bien le cas où l'écart entre t et t_i est élevé, car quand une personne n'est plus détectée, aucune information, concernant le moment où elle pourrait être re-détectée, n'est disponible. Toutefois, quand deux détections sont proches dans le temps, il est important de quantifier cette proximité.

La loi de probabilité la plus couramment utilisée (dans le cadre du suivi d'objets basé détections) pour représenter la probabilité de relier deux objets espacés en temps, est la loi

géométrique. On pourrait l'exprimer de la façon suivante :

$$P^g_{time}(t|t_i) = \begin{cases} (1-p_t)p_t^{t-t_i-1} & \text{si } t \geq t_i + 1 \\ 0 & \text{sinon} \end{cases} \tag{4.3}$$

Cette loi suppose un paramètre p_t qui peut s'interpréter comme le taux d'oubli du détecteur. Le principal problème rencontré avec une telle modélisation est que l'absence de détection sur une période de temps (alors que l'objet est toujours présent) n'est pas forcément due au défaut du détecteur lui-même. Cette absence de détection peut provenir du fait que l'objet ne soit plus détectable (par occultation partielle, ou, dans le cas des visages, par rotation de la tête). En fixant un taux d'oubli du détecteur avec la loi géométrique, ces cas ne sont pas représentés et la probabilité $P_{time}(t|t_i)$ est finalement sous-estimée.

En résumant, P^g_{time} représente assez bien la probabilité recherchée quand l'écart entre t et t_i est faible, alors que pour la loi uniforme c'est le contraire. Cela nous a amené à combiner ces deux lois afin d'utiliser la représentation géométrique pour un $t - t_i$ faible (inférieur à un seuil fixé τ) et la loi uniforme quand cet écart est plus important (supérieur à τ). Cette probabilité s'écrit :

$$P_{time}(t|t_i) = \begin{cases} (1-p_t)p_t^{t-t_i-1} & \text{si } t - t_i \in \{1, ..., \tau\} \\ \frac{p_t^\tau}{t_{max}-\tau+1} & \text{si } t - t_i \in \{\tau+1, ..., t_{max}\} \\ 0 & \text{sinon} \end{cases} \tag{4.4}$$

Cette loi représente assez bien la probabilité de lier des détections en fonction de leur date d'apparition dans la vidéo, mais le problème est qu'elle fait intervenir beaucoup de paramètres (p_t,τ et t_{max}) qui peuvent être variables en fonction des séquences vidéo.

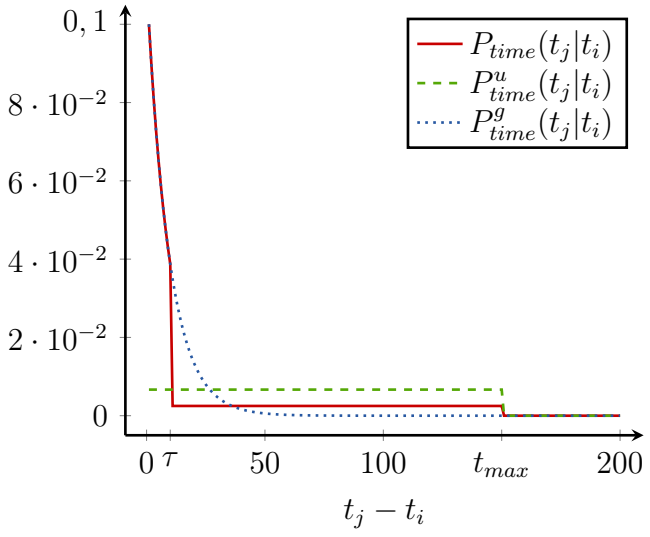

FIGURE 4.1. Aperçu des lois utilisées pour représenter la probabilité $P_{time}(t|t_i)$ en fonction de l'écart $t - t_i$. P^u_{time} représente la densité de la loi uniforme avec $t_{max} = 150$, P^g_{time} la densité de la loi géométrique avec $p_t = 0.9$ et P_{time} celle qui est géométrique sur $\{0, ..., \tau\}$ et uniforme sur $\{\tau + 1, ..., t_{max}\}$.

Pour illustrer les performances de ces trois représentations possibles, une expérimentation a été faite sur la vidéo 1 (cf. description de la base de tests). Cette vidéo fait intervenir

des personnes circulant avec de nombreux changements de directions. Ainsi il est possible d'observer des moments où les personnes sont de face, avec des détections de visages très rapprochées en temps, et aussi des périodes où il n'y a plus de détection pendant plusieurs centaines de frames. C'est précisément pour cela que cette vidéo a été sélectionnée.

En effectuant des regroupements se basant sur l'information temporelle seule, les résultats ne sont pas exploitables. Afin de présenter tout de même des résultats comparatifs, à cette probabilité ont été associées deux mesures basées sur l'apparence (histogramme HS-V et loi normale RGB présentés à la section 4.1.2) et une mesure combinant position et apparence (loi normale XYRGB présentée à la section 4.1.2). Pour chacun de ces trois cas, les trois représentations (*uniforme*, *géométrique* et P_{time}) pour la probabilité temporelle d'association ont été testées. Ces tests consistent à utiliser un algorithme de regroupement avec 20 consignes différentes sur le nombre de groupes (de 25 à 45 groupes sachant que la vidéo 1 fait intervenir 25 passages) avec les 9 mesures d'association (les 3 différentes pour représenter le temps combinées avec les 3 mesures d'apparence). L'algorithme utilisé permet de trouver une solution optimale au sens de la minimisation des similarités inter-détections des trajectoires (cf. 4 section 4.2.4). La mesure de la qualité d'un regroupement par rapport à la vérité-terrain se fait par la F-pureté présentée à la section 2.4.3. L'algorithme est lancé 20 fois (pour 20 nombres de groupes différents), les résultats de la figure 4.2 présentent les moyennes et écarts-types des F-puretés correspondant à ces tests.

Ces résultats montrent bien que, même associées avec des mesures différentes (HS-V, RGB et XYRGB), les représentations de la probabilité temporelle employant une loi géométrique donnent de meilleurs résultats que la loi uniforme. Mais la loi joignant les deux (P_{time}) permet encore d'obtenir des solutions plus proches de la vérité-terrain.

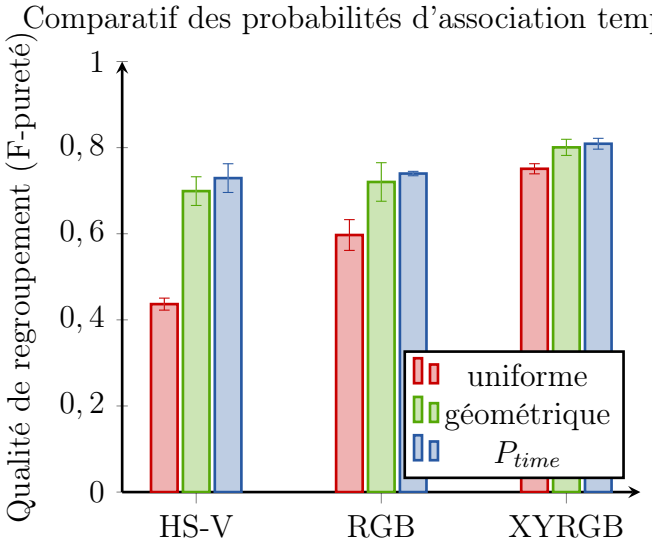

FIGURE 4.2. Moyennes et écarts-types des F-puretés obtenues lors du regroupement des détections en utilisant différentes représentations de la probabilité d'associer des détections au vu de leurs temps. Les différentes lois testées sont les suivantes : *uniforme* : loi uniforme avec $t_{max} = 100$, *géométrique* : loi géométrique avec $p_t = 0.9$ et P_{time} loi liant les deux précédentes avec $\tau = 10$. Les résultats sont obtenus en associant à chacune de ces lois : des mesures d'apparence colorimétrique (*HS-V* out *RGB*) ou encore une mesure faisant intervenir la couleur et la position (*XYRGB*).

Pour la majorité des expérimentations effectuées, la loi géométrique a été utilisée parce

qu'elle présente de bon résultats et n'utilise qu'un paramètre (p_t) (contrairement à P_{time} qui fait intervenir le paramètre τ en plus).

4.1.2. Descriptions et similarités d'apparence

La caractérisation de l'apparence des images dans le but de mesurer des similarités entre elles est un vaste problème qui a déjà mené à des méthodes très diverses.

Pour construire les descripteurs et une mesure de similarité entre eux, il faut premièrement définir la zone image utilisée. Si la simple zone de détection de visage est prise en compte, il est difficile d'utiliser un descripteur fiable, si ce n'est ceux de la reconnaissance faciale qui s'avèrent n'être pas très robustes dans le cas de la vidéosurveillance en environnement non contrôlé. Comme le descripteur recherché ici peut se limiter au cadre d'une séquence vidéo, l'apparence vestimentaire des individus est une information à ne pas négliger pour les décrire. Le premier problème rencontré est que le piéton n'est pas toujours vu en entier par la caméra et que l'ensemble de l'apparence de ses vêtements n'est pas visible. Ensuite, il faudrait un système de segmentation pour pouvoir définir précisément la forme du piéton et ne pas intégrer la couleur de l'arrière plan dans le descripteur. Face à ces deux difficultés, une solution relativement simple et efficace consiste à agrandir vers le bas la zone issue du détecteur de visages pour inclure la couleur des vêtements au niveau du torse.

La comparaison des apparences fait généralement intervenir trois aspects : un espace de description, un modèle probabiliste et une métrique. Les différentes combinaisons des possibilités sur ces trois aspects ont donné une grande variété d'approches pour mesurer la similarité entre apparences. Notre étude se concentre sur les méthodes les plus simples ne nécessitant pas de paramétrage particulier. L'objectif est de ne pas encombrer le système avec des techniques complexes qui n'améliorent pas toujours significativement les résultats finaux. Dans le cadre de cette thèse, principalement trois types d'apparence ont été testés :

- statistique colorimétrique : histogramme HS-V et loi normale à trois dimensions (RGB)
- statistique position et colorimétrique : matrices de covariance (XYRGB), corrélation croisée normalisée (ZNCC) et lois normales positions couleurs (XYRGB)
- descripteur spécifique aux visages : descripteur POEM utilisé en reconnaissance faciale.

Ces approches sont décrites dans les sections suivantes et ensuite comparées dans notre cadre applicatif.

4.1.2.1. Distributions colorimétriques

Pour représenter l'apparence d'une détection, il est possible de simplement utiliser un histogramme des couleurs rencontrées. L'histogramme HS-V, issu des travaux de P. Perez et al. 2002, est relativement bon pour représenter la couleur. Il se base sur l'espace colorimétrique HSV (resp. teinte, saturation et valeur) et est constitué de la concaténation d'un histogramme 2D des composantes H et S ainsi que d'un histogramme 1D sur la composante V. Pour le construire, un filtrage est utilisé pour mettre dans la partie HS de l'histogramme les pixels faisant intervenir une valeur et une saturation suffisamment élevées, et dans la partie V les valeurs des pixels à faible saturation ou faible valeur (*ie* trop proche du noir, blanc ou d'un gris). Ainsi la partie 2D représentera la colorimétrie de l'image et la partie V représentera plutôt les niveaux de gris. Pour le filtrage nous avons utilisé le seuil empirique présenté par P. Perez et al. 2002.

La dissimilarité d'apparence utilisée est simplement la distance de Hellinger entre les histogrammes HS-V calculés à partir des détections :

$$d_{hsv}(\boldsymbol{a}_i, \boldsymbol{a}_j) = \sqrt{1 - \sum_{k=1}^{n} \sqrt{a_i^k a_j^k}} \tag{4.5}$$

où $\boldsymbol{a}_i = (a_i^1, ..., a_i^n)$ représente l'histogramme HS-V de la détection i.

Une autre approche consiste à représenter la répartition des couleurs par une loi paramétrique. Cette voie a été testée en utilisant une loi normale à trois dimensions sur les canaux rouge, vert et bleu. La dissimilarité est calculée en utilisant la distance de Hellinger entre deux lois normales :

$$d_{rgb}(\boldsymbol{a}_i, \boldsymbol{a}_j) = D_H(\mathcal{N}(\mu_{RGB}^i, \Sigma_{RGB}^i), \mathcal{N}(\mu_{RGB}^j, \Sigma_{RGB}^j)) \tag{4.6}$$

où l'apparence $\boldsymbol{a}_i = (\mu_{RGB}^i, \Sigma_{RGB}^i)$, extraite d'une détection, est constituée de la moyenne RGB et de la matrice de covariance. Elles sont estimées par les estimateurs classiques sur l'ensemble des pixels constituant la détection.

Le point important à prendre en compte avec ces descriptions colorimétriques est qu'elles ne font pas intervenir la répartition spatiale des couleurs. Comme la position des pixels n'est pas prise en compte, deux images présentant les mêmes couleurs, mais à des positions différentes, sont considérées comme identiques. Pour pallier à ce défaut, il est possible d'utiliser une représentation qui tienne compte des aspects colorimétriques et spatiaux.

4.1.2.2. Corrélation entre intensités lumineuses

La première métrique employée pour la comparaison d'images est souvent la corrélation croisée normalisé (ZNCC). Cette métrique peut être vue comme le coefficient de corrélation entre les intensités lumineuses des deux images. Elle permet de comparer deux images de tailles identiques avec une invariance aux changements affines d'illumination, ce qui nécessite un redimensionnement à une taille canonique (50×50 pixels pour les expérimentations). Comme elle se base sur une comparaison pixel à pixel, le moindre décalage ou la moindre déformation de l'image peut avoir des conséquences importantes sur la métrique. Ce type de métrique n'est pas suffisamment robuste aux déformations du visage lié à la rotation de la tête pour qu'elle soit réellement applicable.

4.1.2.3. Covariances spatio-colorimétriques

Une autre approche pour représenter l'apparence d'un objet, consiste à utiliser la matrice de covariance entre des caractéristiques spatiales et colorimétriques de l'objet. L'intérêt est d'utiliser conjointement des statistiques spatiales et colorimétriques d'une image.

T. Oncel et al. 2006 sont les premiers à utiliser les matrices de covariance de cette manière. L'idée est de ne pas avoir des statistiques pour chaque caractéristique prise indépendamment, mais d'observer leurs corrélations. L'ensemble des caractéristiques fait généralement intervenir : la position (en pixels) dans l'image, l'intensité lumineuse (ou les composantes couleurs), les valeurs absolues des gradients en x et y, la norme et l'orientation du gradient et les dérivées au second ordre des intensités lumineuses en x et y.

L'utilisation d'une matrice de covariance entre différentes caractéristiques pour décrire une zone d'une image est relativement récente. Plusieurs auteurs ont montré son intérêt,

notamment pour la détection de piétons (J. Yao et J.-M. Odobez 2008), mais aussi comme descripteur pour le suivi d'objets (F. Porikli et al. 2005).

Plus formellement, en considérant un ensemble P de pixels définissant la zone d'intérêt de l'image et en décrivant chaque pixel $\boldsymbol{p}_i \in P$ par un vecteur de caractéristiques, la matrice de covariance est estimée de la façon suivante :

$$\Sigma_P = \frac{1}{|P|-1} \sum_{i \in P} (\boldsymbol{p}_i - \boldsymbol{\mu}_P)^\top (\boldsymbol{p}_i - \boldsymbol{\mu}_P) \tag{4.7}$$

où $\boldsymbol{\mu}_P$ correspond à la moyenne des $\boldsymbol{p}_i$ de P. La principale difficulté liée à l'utilisation des matrices de covariance est que l'ensemble de ces matrices (symétriques définies positives, noté $\mathcal{M}$) ne forme pas un espace euclidien et le calcul des distances entre ces matrices n'est pas simple. Les travaux de X. Pennec et al. 2006 proposent de voir $\mathcal{M}$ comme une variété et d'utiliser une métrique riemannienne. Les auteurs présentent un cadre à la mesure de distances entre les matrices de covariance ; ce cadre est notamment utilisé par J. Yao et J.-M. Odobez 2008 pour détecter des piétons.

Cette mesure a ensuite été utilisée pour être appliquée à des méthodes plus particulières, qui ne nécessitent pas que la définition d'une distance. C'est le cas de A. Cherian et al. 2011 qui utilisent les matrices de covariance dans le cadre du partitionnement et présentent des résultats intéressants dans le cadre de la vidéosurveillance (regroupement de piétons ou de visages). Pour ce qui est du suivi d'objets, A. Tyagi et J. Davis 2008 mettent en place un formalisme qui permet de définir un filtre de Kalman dans le cas particulier des matrices de covariance.

L'ensemble des publications citées ici et les premiers résultats obtenus ont encouragé l'utilisation des matrices de covariance dans notre cadre applicatif.

Au lieu de n'utiliser que la matrice de covariance des caractéristiques de l'image à décrire, il peut être envisagé d'utiliser aussi les valeurs moyennes de ces caractéristiques. En faisant cela, l'image est alors décrite par une loi normale multivariée représentant la répartition de ces caractéristiques. Cette idée a aussi été développée et illustrée avec l'espace colorimétrique RGB (cf. section suivante et section 4.2.2).

4.1.2.4. Loi normale spatio-colorimétrique

Comme les détections sont extraites d'une même séquence vidéo, et que l'on cherche à les regrouper, il est important de rechercher des descriptions plus discriminantes que celles employées, par exemple, pour des détecteurs de piétons. Ainsi, il peut être intéressant de ne pas utiliser uniquement la covariance pour décrire l'image (comme cela est fait pour certains détecteurs de piétons), mais aussi la moyenne sur ces caractéristiques. Cela conduit à représenter les images par une loi normale multivariée sur les différentes caractéristiques. Si l'on ne recherche qu'à décrire une image, la moyenne spatiale en coordonnées XY n'est pas utile. Mais comme les détections sont des sous-images des images de la vidéo et que leur position importe, la moyenne spatiale permet d'intégrer directement une information de la position de la détection dans la vidéo.

Pour comparer deux détections selon ce principe, la distance de Hellinger est utilisée :

$$d_{xyrgb}(\boldsymbol{a}_i, \boldsymbol{a}_j) = D_H(\mathcal{N}(\mu^i_{XYRGB}, \Sigma^i_{XYRGB}), \mathcal{N}(\mu^j_{XYRGB}, \Sigma^j_{XYRGB})) \tag{4.8}$$

où les moyennes et covariances sont estimées sur les pixels des détections en utilisant leur position XY en pixels et leur colorimétrie RGB.

FIGURE 4.3. Exemple de lois normales XYRGB extraites à partir de trois détections de visages. L'image de gauche présente les détections sur l'image de la vidéo, l'image de droite présente trois ensembles de pixels tirés suivant les trois lois normales XYRGB construites par trois détections.

La figure donne un aperçu de la description des détections par lois normales XYRGB. Elle présente des ensembles de pixels tirés suivant les lois normales construites par les détections de l'image.

4.1.2.5. Utilisation d'un descripteur de la reconnaissance faciale

Le but étant de regrouper des visages par identité, il peut sembler plus pertinent d'utiliser des outils spécifiques, notamment ceux de la biométrie faciale. Cette voie a été envisagée en testant le descripteur POEM (*Patterns of Oriented Edge Magnitude*) de N.-S. VU, H. M. DEE et al. 2012.

Le descripteur POEM hérite de différents aspects des descripteurs couramment utilisés : LBP (*Local Binary Pattern*) décrivant bien les aspects textures et SIFT ou HOG qui caractérisent la forme du voisinage local. Un autre point crucial que prend en compte le descripteur POEM est le temps de calcul. Ce descripteur reste plus rapide que la plupart des systèmes de description faciale les plus performants.

La majorité des descripteurs de visages font intervenir une première étape de recalage pour faire correspondre les éléments du visage à une position canonique. Dans notre cadre applicatif, avec des tailles de visage de l'ordre de 60×60 pixels, le recalage n'a pu être exploité.

Les performances auraient pu être améliorées avec une décomposition en composantes principales (PCA ou *Whitened* PCA) ou encore un pré-traitement pour normaliser l'illumination (N.-S. VU 2010). Toutefois, les tests menés ont découragé l'utilisation d'un tel descripteur et ces voies n'ont pas été explorées.

Les principaux obstacles rencontrés avec les descripteurs de reconnaissance faciale sont la faible résolution et la mauvaise qualité des détections issues de la vidéosurveillance non-contrainte. De plus, ne cherchant pas à faire de la ré-identification (qui ferait intervenir des caméras et des dates différentes), la colorimétrie des vêtements paraît plus discriminante qu'un système de reconnaissance faciale difficilement exploitable.

4.1.2.6. Comparatif des descriptions d'apparence

Pour comparer la pertinence des descripteurs d'apparence des détections, ceux-ci ont été employés pour regrouper les détections de visages issues d'une vidéo. L'algorithme utilisé pour regrouper les détections permet de trouver un partitionnement optimal avec un nombre de groupes fixé et une matrice des similarités fixée. Cet algorithme donne une solution optimale au sens de la vraisemblance décrite au chapitre précédent. Plus précisément, le partitionnement considéré comme optimal est celui qui regroupe les détections, de telle sorte que le produits des probabilités de transition apparaissant au sein des groupes soit maximal (cf. k-CC algorithme 4 section 4.2.4).

Les différentes distances entre les descriptions d'apparences des détections sont les suivantes :

XYRGB : distance entre les lois normales position XY et couleur RGB (d_{xyrgb}).

- *cov-XYRGBgI* : distance entre les matrices des covariances position (XY), couleur (RGB) et norme du gradient niveaux de gris (gI). La distance est celle basée sur la variété riemannienne des matrices de covariance.
- *cov-XYRGB* : distance (sur la variété riemannienne) entre les matrices de covariance position (XY) et couleur (RGB).
- *RGB* : distance de Hellinger entre les lois normales RGB (d_{rgb}).
- *HS-V* : distance de Hellinger entre histogrammes HS-V (d_{hsv}).
- *ZNCC* : distance basée sur la corrélation croisée normalisée entre images 50×50 en niveaux de gris.

Les résultats sont exprimés sous la forme d'une qualité de regroupements en comparaison avec la vérité-terrain. La grandeur utilisée est une F-pureté qui est décrite à la section 2.4.3. Pour chacun des descripteurs d'apparence utilisés, l'algorithme est exécuté 30 fois avec des nombres de groupe allant de 10 à 40. Les moyennes et les écarts-types des F-puretés sont présentés au graphique 4.4.

La figure 4.4 montre que les approches joignant des statistiques spatiales et de colorimétrie permettent de mieux décrire les apparences des détections. En comparant les deux approches basées sur les matrices de covariance, on voit que l'ajout de caractéristiques (ici la norme du gradient de l'intensité) n'améliore pas forcément les performances en terme de regroupement des détections.

Le fait que les performances de la modélisation par loi normale XYRGB soient au-dessus des autres peut s'expliquer par le fait qu'elle fait intervenir la position de la détection dans l'image. Comme elle n'utilise pas que l'apparence d'une détection, il est normal que ses performances dépassent les autres.

4.1.3. Représentation de la position

Pour caractériser la localisation d'une détection dans l'image, deux aspects entrent en compte : la position de la détection et sa taille. Envisager des détections non-rectangulaires nécessiterait un *a priori* supplémentaire : par exemple supposer que les contours de la personne soient marqués et employer un détecteur de contours, ou que l'arrière plan de la détection soit fixe et effectuer une extraction d'arrière plan. Étant donné que les détecteurs de visages (et aussi de piétons) non-rectangulaires sont plutôt rares, les détections seront supposées rectangulaires.

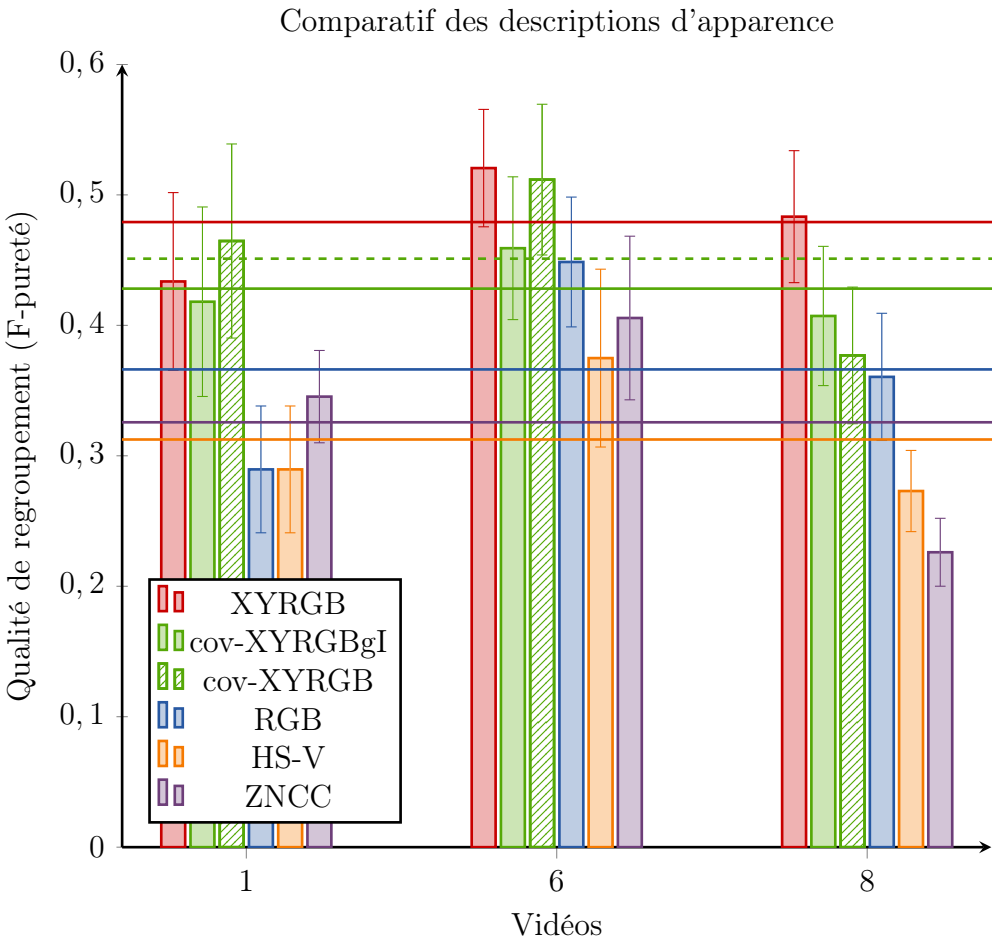

FIGURE 4.4. Moyennes et écarts-types des F-puretés obtenues lors du regroupement des détections en utilisant différentes descriptions de l'apparence des détections. Les traits horizontaux montrent les moyennes sur les trois vidéos des performances des six représentations de l'apparence.

En définitive, la représentation retenue est celle qui représente la zone détectée par une loi normale à deux dimensions (estimée par l'ensemble des positions intervenant à la détection).

La première approche, pour représenter la proximité spatiale des détections par une loi de probabilité, est d'utiliser une ou plusieurs lois normales. C'est notamment ce qui est présenté par les travaux (B. WU et R. NEVATIA 2007 ; L. ZHANG et al. 2008) déjà cités précédemment. Les paramètres de ces lois normales sont fixés empiriquement et la taille de la détection est prise en compte séparément (B. WU et R. NEVATIA 2007) ou elles sont apprises sur des données de test (L. ZHANG et al. 2008). Le problème qui survient vite avec de tels paramètres est leur sur-ajustement. Cela peut conduire à des résultats qui présentent plus la qualité de l'ajustement des paramètres que la pertinence de la représentation probabiliste.

Cherchant à être parcimonieux en termes de paramètres à fixer empiriquement, nous avons recherché une loi normale qui représente la position, dont la matrice de covariance soit directement liée à la taille de la détection. Pour ce faire, les coordonnées des pixels constituant la détection sont prises en compte pour estimer la moyenne et la matrice de covariance de la loi normale recherchée.

Plus formellement, en considérant des détections rectangulaires, la position d'une détection i de centre $\boldsymbol{x}_i = (x_i, y_i)$ et de taille $\boldsymbol{s}_i = (w_i, h_i)$ sera représentée par une loi normale. En

utilisant les estimateurs standards de la moyenne et de la matrice de covariance sur toutes les coordonnées des pixels intervenant sur la zone de détection, la position de la détection suit la loi suivante :

$$\boldsymbol{x} \sim \mathcal{N}\left(\boldsymbol{x}_i, \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}\right) \tag{4.9}$$

où σ_x^2 correspond à la variance en x des pixels et σ_y^2 celle en y. En développant les calculs, on se rend compte que :

$$\sigma_x^2 = \frac{h_i}{w_i h_i - 1} \sum_{j=1}^{w_i} \left(j - \frac{w_i - 1}{2}\right)^2 = \frac{h_i w_i}{12(h_i w_i - 1)}(w_i^2 - 1) \tag{4.10}$$

$$\sigma_y^2 = \frac{h_i w_i}{12(h_i w_i - 1)}(h_i^2 - 1) \sim \frac{h_i^2}{12} \tag{4.11}$$

ce qui se rapproche du lien recherché entre les écarts-types des positions (σ_x et σ_y) et la taille de la détection (w_i,h_i).

La matrice de covariance ayant σ_x^2 et σ_y^2 sur la diagonale sera notée : Σ_{xy}^i. Comme les coordonnées x et y sont indépendantes, leur covariance est bien nulle.

Ayant représenté la position d'une détection par une loi normale, la comparaison en position entre deux détections peut se faire en utilisant une distance entre leur distribution. Là encore, une similarité est définie à l'aide de la distance de Hellinger :

$$s_{pos}(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 - D_H\left(\mathcal{N}(\boldsymbol{x}_i, \Sigma_{xy}^i),\ \mathcal{N}(\boldsymbol{x}_j, \Sigma_{xy}^j)\right) \tag{4.12}$$

4.1.4. Description du mouvement

Le mouvement d'une détection au sein de la séquence vidéo représente un indice important pour améliorer le regroupement des détections, mais reste difficile à utiliser de manière fiable. Le cas typique, où une information de direction de déplacement est intéressante, est le cas d'un croisement de personnes faisant intervenir des directions différentes. Toutefois, étant donné le caractère parfois erratique des mouvements des piétons, il reste toujours difficile de déterminer si deux personnes se croisent ou changent brusquement de direction.

Deux approches ont été employées pour extraire une information de vitesse. La première s'appuie sur le suivi mono-cible d'une image au cours de la vidéo, alors que la deuxième se base sur le flot optique.

4.1.4.1. Suivi avec matrices de covariance

La description de l'apparence étant utilisée tant dans le cadre de la détection d'objets que pour le suivi, nous avons envisagé un suivi des détections de visages en s'appuyant sur ce type de descripteurs. En prenant une région de référence (région à suivre) pour l'image au temps t d'une vidéo, l'objectif du suivi est de déterminer où se situe cette région à l'image au temps $t + 1$. Pour se faire, la région de référence est décrite par sa matrice de covariance et la métrique riemannienne est utilisée pour trouver la région la plus ressemblante de l'image suivante.

Les caractéristiques recommandées par F. Porikli et al. 2005 pour construire les covariances, sont la position x et y, le gradient en x et y, la norme du gradient et la couleur. Ce sont les caractéristiques que nous avons utilisées pour mettre en œuvre le suivi de visages.

Algorithme 2 : Suivi d'une région de référence

calcul de la matrice de covariance de référence C_r à la position X_r;
pour $t \in frames$ **faire**
  $X_t \leftarrow \arg\min_{X \in W_X}(d_{\mathcal{M}}(C_X, C_r))$;
  si $d_{\mathcal{M}}(C_t, C_r) < \alpha\ d_{\mathcal{M}}(C_1, C_r)$ **alors**
    arrêt du suivi;
  fin
  mise à jour de C_r;
fin

L'algorithme 2 décrit le procédé du suivi à partir d'une région de référence où $frames$ représente l'ensemble des temps considérés, W_x une fenêtre de recherche autour de la position X et α un critère d'arrêt. Dans leur approche, F. Porikli et al. 2005 effectuent une recherche sur la totalité de l'image pour déterminer la position de l'objet suivi, ce qui entraine des coûts de calcul non négligeables. Même avec des décompositions en images intégrales pour un calcul rapide des matrices de covariance, le suivi d'une cible dépasse difficilement deux images par seconde pour une résolution de 320×240 pixels.

Pour accélérer le procédé, nous avons utilisé une optimisation (basée sur la méthode Nelder-Mead) pour trouver la position correspondant à la distance minimale. Cette méthode n'est pas exacte, elle donne un minium local proche de l'initialisation. Cela n'est toutefois pas aberrant étant donnée la continuité des positions, au cours du temps, de l'objet suivi. Comme de t à $t+1$ l'objet ne s'est en général que peu déplacé, initialiser l'optimisation à X_t pour trouver X_{t+1} donne de bons résultats. Pour rechercher X_{t+1}, au lieu de calculer les matrices de covariance sur tout un voisinage (*eg* 40×40) de X_t, elles ne sont calculées qu'une cinquantaine de fois au cours de l'optimisation. Cela nous a permis de limiter le temps de calcul et de l'utiliser dans le cadre du suivi de détections.

La stratégie utilisée par F. Porikli et al. 2005 pour mettre à jour la description de référence paraît pertinente. Elle consiste à calculer la moyenne (au sens de la métrique riemannienne) des K précédentes matrices de covariance $C_{t-1}, ..., C_{t-K}$. Cette stratégie a été implémentée mais sa pertinence n'a pas été mesurée, dans notre cadre applicatif, étant donné que les suivis restent relativement courts.

Disposant de la vidéo entière, il est donc possible d'utiliser le suiveur dans les deux directions : vers le futur et vers le passé, ce qui permet de gagner en robustesse. Si, par exemple, le suivi vers le futur vient à défaillir, il peut arriver que celui provenant d'une détection ultérieure reste fiable vers son passé, et ainsi ces deux détections peuvent se raccorder.

Pour tester le potentiel de ce type de suivi dans le cadre du regroupement de détections, nous avons initié des suiveurs (vers le passé et vers le futur) à partir de chaque détection rencontrée. Bien sûr, cette solution demande beaucoup de temps de calcul et possède de nombreuses redondances. Mais cela permet de ne pas introduire d'erreur de regroupement à cette étape, car toutes les détections sont encore séparées, aucune fusion n'est effectuée à cette échelle plus locale. L'intérêt est de découpler les suivis, qui se déroulent de manière locale, d'avec le regroupement qui cherche à être le plus global possible et peut envisager toutes les partitions des détections.

L'intégration du suivi local, par une similarité robuste entre les détections, a finalement été mise de côté. Les principales raisons sont : le temps de calcul qui n'est pas négligeable, la difficulté d'ajustement des paramètres du suiveur qui permettent de l'arrêter quand il n'est plus fiable et finalement l'apport n'est pas aussi important qu'on pourrait le penser (cf. 4.1.4.3).

4.1.4.2. Flot optique

Une autre manière de prendre en compte une information de mouvement issue des images, est le flot optique qui permet d'avoir une estimation dense du mouvement. Il consiste à extraire les mouvements des pixels entre deux images, en associant à chaque pixel de l'image un déplacement 2D. Le flot optique est une approximation du mouvement dans l'image qui devrait représenter la projection des mouvements réels de la scène observée par la caméra. Dans la littérature, il existe de nombreuses méthodes pour estimer le flot optique. Les trois grandes familles d'algorithmes d'estimation du flot optique sont :

— *bloc matching* : mise en correspondance d'imagettes pour estimer leur déplacement.
— variationnelles : calcul de la vitesse par dérivations spatio-temporelles de l'intensité lumineuse de l'image. Deux approches se sont bien rependues dans le domaine, l'une locale (Lucas-Kanade) et l'autre globale (Horn-Schunck).
— méthode éparse : mise en correspondance de points d'intérêts (Lucas-Kanade-Tomasi).

Comme le suggèrent les résultats d'une évaluation (J. Barron et al. 1994) de différentes techniques d'estimation de flot optique, nous avons choisi de nous baser sur la méthode variationnelle locale provenant des travaux de B. Lucas, T. Kanade et al. 1981. Elle donne une bonne précision locale, mais rend difficilement compte des larges déplacements rencontrés sur une séquence d'images. Ce point a été traité par différentes techniques, comme la décomposition de l'image en sous-régions (T. Brox et al. 2009 ; D. Sun et al. 2010), ou l'estimation multi-échelles (J. Marzat et al. 2009).

Comme la plupart des méthodes variationnelles, la méthode de Lucas et Kanade se base sur la conservation de l'illumination pour estimer le déplacement des pixels entre deux images consécutives d'une séquence. Cette conservation peut s'écrire de la manière suivante :

$$I(\boldsymbol{x} + \boldsymbol{v}_{\boldsymbol{x}}, t+1) - I(\boldsymbol{x}, t) = 0 \tag{4.13}$$

où $I(\boldsymbol{x}, t)$ représente l'intensité lumineuse de l'image à la position $\boldsymbol{x}$ à l'instant t et $\boldsymbol{v}_{\boldsymbol{x}}$ le flot optique en $\boldsymbol{x}$. Ceci conduit, après développement au 1er ordre, à :

$$\nabla I_{xy}(\boldsymbol{x}, t)^\top \boldsymbol{v}_{\boldsymbol{x}} = -I_t(\boldsymbol{x}, t) \tag{4.14}$$

où ∇I_{xy} est le gradient de l'image et I_t la dérivée partielle de I en fonction du temps. Dans le cadre de la méthode de Lucas et Kanade, le flot est supposé constant sur un voisinage de la position $\boldsymbol{x}$ considéré. Ainsi, pour chaque point $\boldsymbol{x}$ de l'image I, le flot optique $\boldsymbol{v}_{\boldsymbol{x}}$ minimisera :

$$\sum_{\boldsymbol{p} \in \mathcal{V}(\boldsymbol{x})} (\nabla I_{xy}(\boldsymbol{p}, t)^\top \boldsymbol{v}_{\boldsymbol{x}} + I_t(\boldsymbol{p}, t))^2 \tag{4.15}$$

où $\mathcal{V}(\boldsymbol{x})$ est un voisinage de $\boldsymbol{x}$. Une résolution par moindre carré est bien adaptée à ce problème et permet de trouver $\boldsymbol{v}_{\boldsymbol{x}}$ rapidement.

En supposant le flot constant sur un voisinage, le problème est soit de perdre en précision si le voisinage est trop large ou soit de ne pas réussir à représenter de grands déplacements.

Afin de pallier à cela, J. Marzat et al. 2009 proposent une approche multi-échelle de grossier à fin. Pour se faire, une pyramide d'images à différentes échelles est construite, le flot optique est calculé à la plus grande échelle (image de résolution la plus faible) et répercuté sur l'image à l'échelle inférieure. On procède ainsi de suite jusqu'à obtenir le flot optique à pleine résolution (cf. J. Marzat et al. 2009). À l'échelle la plus importante, de larges déplacements sont pris en compte et ensuite répercutés pour raffiner le flot optique. Les larges et petits déplacements sont donc pris en compte. En pratique, trois à cinq échelles sont employées avec des voisinages carrés d'une dizaine de pixels de côté. Ensuite, afin d'avoir plus de résistance au bruit, les flot optiques obtenus avec l'image précédente et avec la suivante sont moyennés.

La figure 4.5 donne un aperçu du flot optique calculé avec deux images consécutives sur une scène de vidéosurveillance. Même si des erreurs apparaissent au niveau des occultations (dues à l'hypothèse de conservation de l'illumination qui n'est pas respectée), le flot optique permet d'obtenir une bonne estimation du mouvement. Cette figure illustre aussi le fait que l'extraction du flot optique permet de distinguer différentes personnes, cela se voit sur la représentation colorimétrique du flot optique de l'image en bas à gauche.

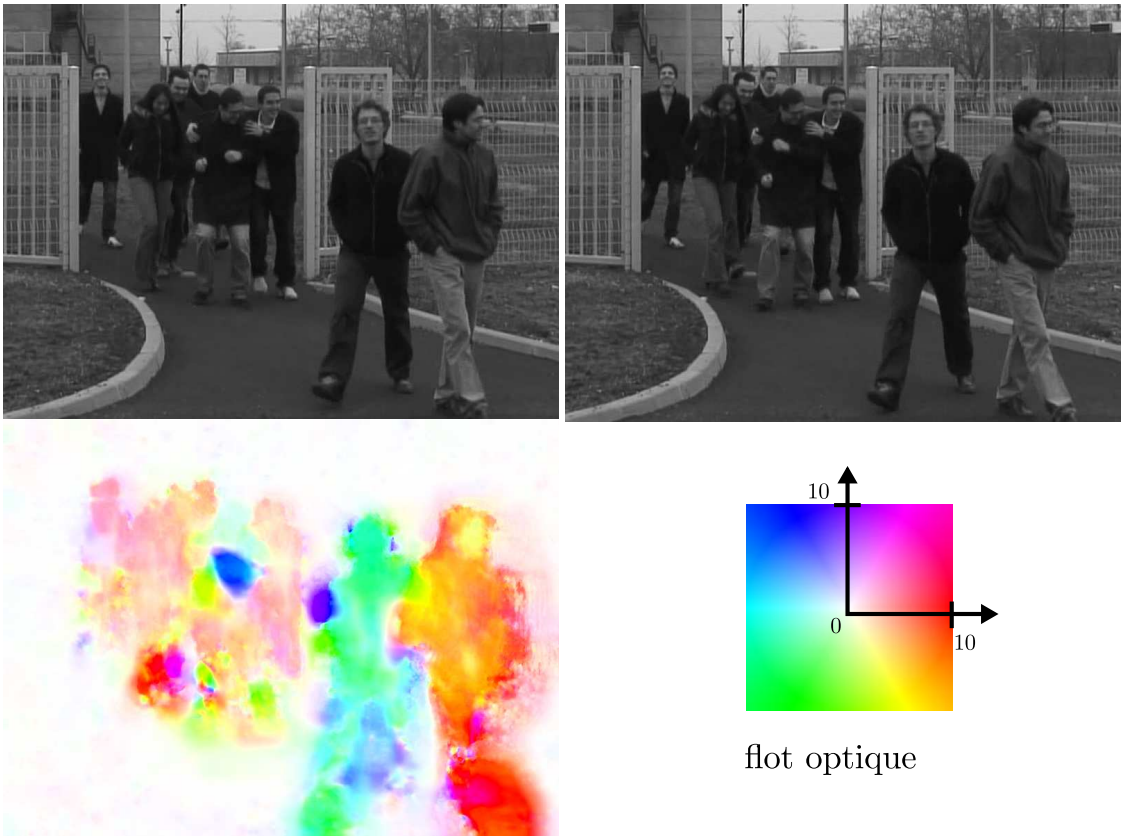

Figure 4.5. Exemple de flot optique calculé avec une méthode Lucas-Kanade pyramidale. *En haut* : deux images ayant servi à calculer le flot optique, *en bas à gauche* : flot optique calculé, *en bas à droite* : signification des couleurs servant à représenter un vecteur de flot (en pixels).

Une critique souvent faite au flot optique est son temps de calcul. Étant dense, il est en général plus lent à l'exécution que le serait une approche éparse avec des appariements de points d'intérêts. Dans notre cas, le flot optique n'est pas à calculer sur chaque image de la séquence, nous nous concentrons sur les zones de l'image issues des détections. Ces zones

sont légèrement agrandies pour ne pas avoir de perte de précision aux bords de la zone.

L'information dynamique donnée par le flot optique permet de pouvoir prédire la position d'une détection à un instant différent de son apparition. Comme le flot optique donne une information de vitesse à un instant donné, il parait difficile d'utiliser un modèle dynamique différent de celui à vitesse constante. En pratique, étant donné que le mouvement d'un piéton peut être particulièrement erratique, il n'est pas forcément raisonnable d'utiliser un modèle plus complexe.

Ainsi, pour mettre en place une métrique utilisant le flot optique extrait à chaque détection, il est possible de se baser sur ce qui a été défini pour la distance en position (cf. 4.1.3). Nous utiliserons une loi normale à deux dimensions pour représenter la position d'une détection i :

$$\boldsymbol{x}, t \sim \boldsymbol{x} + (t - t_i) \begin{pmatrix} u \\ v \end{pmatrix} \sim \mathcal{N}_{\boldsymbol{x}}(\boldsymbol{x}_i + (t - t_i) \begin{pmatrix} \bar{u}_i \\ \bar{v}_i \end{pmatrix}, \Sigma^i_{xy} + (t - t_i)^2 \Sigma^i_{uv}) \tag{4.16}$$

où Σ^i_{xy} correspond à la covariance en position (calculée à partir des coordonnées des pixels pour la zone de détection, cf. 4.1.3), $(\bar{u}, \bar{v})$ (resp. Σ^i_{uv}) est la moyenne (resp. matrice de covariance) du flot optique calculé par les valeurs du flot dans la zone détectée.

De cette manière, grâce au flot optique, la position est estimée à un temps différent de celui où la détection apparait. Cela va permettre de comparer deux positions de détection à un même temps donné. Le temps le plus intéressant pour comparer deux détections est le temps moyen, car c'est là où il y a le moins d'erreurs d'estimation dues à la non-linéarité du mouvement. Les deux lois comparées pour obtenir une similarité entre leurs deux détections respectives (i et j) sont les suivantes :

$$\mathcal{N}_{\boldsymbol{x}} \left(\boldsymbol{x}_i + \frac{t_j - t_i}{2} \begin{pmatrix} \bar{u}_i \\ \bar{v}_i \end{pmatrix}, \Sigma^i_{xy} + \left(\frac{t_j - t_i}{2} \right)^2 \Sigma^i_{uv} \right) \tag{4.17}$$

et

$$\mathcal{N}_{\boldsymbol{x}} \left(\boldsymbol{x}_j + \frac{t_i - t_j}{2} \begin{pmatrix} \bar{u}_j \\ \bar{v}_j \end{pmatrix}, \Sigma^j_{xy} + \left(\frac{t_i - t_j}{2} \right)^2 \Sigma^j_{uv} \right) \tag{4.18}$$

Comme pour les similarités définies précédemment, la distance de Hellinger est utilisée. La similarité issue de cette distance sera notée : s_{of}. Plus de détails sur le calcul de la distance de Hellinger entre deux lois normales multivariées sont donnés à l'annexe B.3.

4.1.4.3. Comparatif entre suivi et flot-optique

Afin de choisir entre une approche basée sur un algorithme de suivi et une approche basée sur l'estimation du flot optique, quelques expérimentations ont été menées. Sur six vidéos, des similarités ont été construites en utilisant l'apparence, la position et le temps. À cette fonction de similarité ont été ajoutées des mesures de similarité de mouvement basées sur : 1) une estimation du mouvement local par flot optique ou 2) un suivi sur plusieurs images (jusqu'à une centaine).

Partant de ces similarités, l'algorithme (MAP) de regroupement des détections a été employé. Pour présenter des résultats en moyennes et écarts-types, l'algorithme a été exécuté avec 100 valeurs différentes du paramètre p_e de l'*a priori*. Les différents résultats obtenus sont présentés à la figure 4.6.

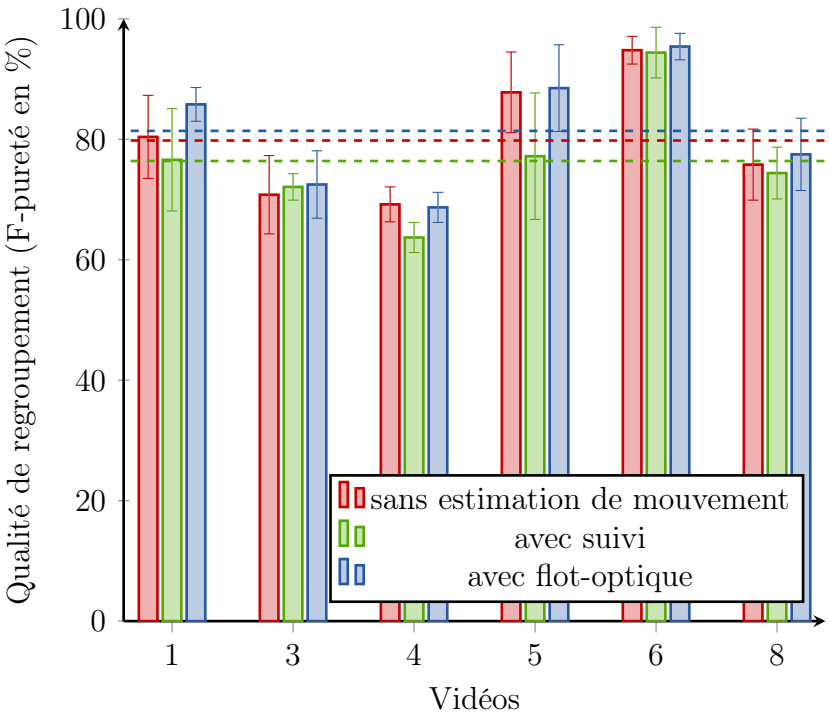

FIGURE 4.6. Résultats montrant l'apport d'une information basée sur le mouvement, et la différence entre une approche suivi et une approche flot-optique. Les pointillés montrent les moyennes des 3 cas sur les 6 vidéos.

Ces résultats montrent qu'en utilisant une estimation du flot optique, l'information de mouvement permet d'améliorer légèrement les performances. Le suivi a permis dans un seul cas (vidéo 3) d'obtenir une performance supérieure à celle n'utilisant aucune information de mouvement. Cette conclusion mériterait une analyse plus poussée. Avec notamment une bonne gestion de l'arrêt du suiveur, le suivi pourrait apporter une information plus pertinente.

Sur ces vidéos, l'approche basée sur le suivi donne donc de moins bons résultats que celle utilisant le flot optique. Cela peut s'expliquer par le fait que, avec des détections de visages, le suiveur se perd. Plusieurs situations peuvent causer la perte du suiveur : une occultation (par un autre piéton ou par un objet de la scène) ou lors de mouvements brusques de la tête (particulièrement quand les piétons changent d'orientation). Or c'est justement dans ces cas que l'information de mouvement aurait été utile. Plus généralement, le suiveur fonctionne bien là ou il y a justement de nombreuses détections et où l'association est plus facile, alors que, lors de longues absences de détections (où l'association est difficile), le suiveur ne parvient pas non plus à relier des détections.

Ces résultats, additionnés au fait que les temps de calcul des suivis sur toutes les détections sont plus élevés que l'estimation du flot optique, font que l'estimation du flot optique a été préférée au suivi.

4.2. Agrégation des différentes similarités entre détections

Même si les définitions présentées permettent une assez bonne cohabitation des différentes similarités, les premières expérimentations effectuées ont montré qu'un bon "réglage" de ces similarités permet d'accéder à de meilleurs résultats.

Cette section décrit premièrement les problématiques attachées à la définition de P_{link} (qui correspond à la probabilité de relier deux détections au sein d'une même trajectoire) à partir des différentes similarités précédemment décrites. Face à ces problématiques soulevées, trois solutions sont ensuite détaillées.

La première traite de la fusion des différentes similarités en représentant les détections par une loi normale joignant les caractéristiques (temps, position, apparence et mouvement).

Ensuite est décrit la façon dont les similarités peuvent s'adapter au problème par un pré-traitement basé sur le partitionnement spectral.

En dernier lieu nous présentons une méthode d'estimation qui permet de fusionner les différentes similarités et de définir une probabilité P_{link}.

4.2.1. Lien entre probabilité de liaison et similarité

Parmi les mesures définies précédemment, figurent des similarités et distances qui ne sont pas directement assimilables à des probabilités et donc à P_{link}.

Les différentes définitions des similarités entre deux observations $\boldsymbol{z}_i$ et $\boldsymbol{z}_j$ sont bien dans l'intervalle $[0, 1]$, toutefois elles ne définissent pas forcément une probabilité assimilable à P_{link}. Le lien entre une similarité s et P_{link} peut être fait de différentes manières, suivant les interprétations que l'on donne à P_{link}.

En supposant que $P_{link}(i \rightarrow j)$ soit la probabilité de relier directement la détection i à la détection j, il est possible de l'assimiler à $s(i, j)$ à condition de normaliser pour que $\sum_{(i,j)\in\mathcal{D}\times\mathcal{D}} P_{link}(i \rightarrow j) = 1$. La solution la plus simple consiste à définir la probabilité de lien de la façon suivante :

$$P_{link}(i \rightarrow j) \doteq \frac{s(i, j)}{\sum_{(k,l)\in\mathcal{D}\times\mathcal{D}} s(k, l)} \tag{4.19}$$

Le principal problème soulevé par cette normalisation est qu'elle définit une probabilité de lien entre i et j parmi tous les liens potentiels. Ces probabilités sont généralement bien plus faibles que celles issues de la probabilité conditionnelle $P_{link}(.|i)$. Dans le cadre du MAP, il en résulte une sous-estimation de la vraisemblance, et les solutions ainsi obtenues favorisent trop fortement l'*a priori*. En pratique, cette approche a été difficilement exploitable.

Une autre possibilité serait de garder l'aspect conditionnel de la probabilité de transition et de la définir comme $P_{link}(.|i)$: probabilité d'atteindre directement une détection sachant que l'on vient de la détection i. Cela pourrait conduire à une normalisation de type :

$$P_{link}(j|i) \doteq \frac{s(i, j)}{\sum_{l\in\mathcal{D}} s(i, l) + cste} \tag{4.20}$$

le terme *cste* étant une constante à rajouter pour représenter le fait que la trajectoire contenant i puisse s'arrêter en i (i peut se relier à tous les éléments de $\mathcal{D}$ ou s'arrêter). Cette

représentation semble mieux correspondre au cadre des chaines de Markov, mais, en pratique, cette normalisation présente quelques difficultés. Le premier problème est de pouvoir fixer *cste*. Comme s ne présente pas une probabilité, la *cste* ne peut pas être interprétée comme la probabilité que la trajectoire s'arrête à une détection donnée. L'autre problème est que, si une détection est plutôt isolée des autres (typiquement un faux positif), la normalisation va artificiellement donner une forte probabilité aux transitions qui en découlent (par une somme des $s(i,.)$ faible). À l'inverse, une détection très proche de nombreuses détections provenant de la même personne, va voir ses probabilités de transitions fortement diminuées (par une somme des $s(i,.)$ élevée).

Étant donné la structure de graphe décrite précédemment pour représenter les dépendances entre les détections, l'utilisation d'un réseau bayésien a aussi été envisagée. Cette modélisation est présentée à l'annexe A.2 qui expose aussi les raisons pour lesquelles elle n'a pas été exploitée.

Se placer dans le cas continu revient à définir la probabilité P_{link} par une loi continue sur les descripteurs de détections et non plus sur les index. En cherchant toujours à s'appuyer sur les similarités définies précédemment, la normalisation la plus simple consisterait à définir :

$$P_{link}(\boldsymbol{z}_j|\boldsymbol{z}_i) \doteq \frac{s(\boldsymbol{z}_i, \boldsymbol{z}_j)}{\int_{\boldsymbol{z}} s(\boldsymbol{z}_i, \boldsymbol{z}) d\boldsymbol{z}} \tag{4.21}$$

où ici s ne représente pas une similarité définie sur des paires d'index de détections mais directement sur les descripteurs des observations issues des détections. Le principal problème rencontré avec cette approche est le calcul du dénominateur. Comme $\boldsymbol{z}_i$ peut faire intervenir des objets très hétérogènes (entier, vecteur, histogramme, matrice de covariance ...), le calcul ou même l'estimation du dénominateur ne paraît pas facile à obtenir.

Suite à ces observations, la première approche discrète (équation 4.19) semble la plus abordable et raisonnable. Toutefois, cela conduit à une trop forte sensibilité au paramètre d'*a priori* p_e due à une sous-estimation de la vraisemblance. C'est pourquoi, en pratique, nous avons évité ces types de normalisations.

Face à ce constat, nous nous sommes penchés sur plusieurs solutions permettant de fusionner les différentes similarités et de définir une probabilité de lien entre les détections. Ces trois solutions font l'objet des trois sections suivantes.

4.2.2. Représentation par une loi normale jointe

La méthode présentée ici consiste à représenter une détection par une loi conjointe sur les différents aspects présentés précédemment (temps, position dans l'image et couleur au sein d'une détection et, en prenant en compte les images passées et futures, le flot optique).

Comme expliqué auparavant (section 4.1.2.3), les matrices de covariance faisant intervenir les différents aspects d'une détection, donnent une information intéressante pour la décrire. Toutefois, en ne prenant qu'une matrice de covariance, on perd la notion des valeurs en absolu des variables de la détection : la matrice de covariance ne représente que la corrélation linéaire entre les variables. Or, si l'on veut comparer deux détections qui sont situées dans une vidéo, il est primordial de garder une notion de position dans l'image et du temps dans la vidéo. Suite à cette constatation, une loi normale multivariée (faisant donc intervenir moyennes et covariances) construite par l'ensemble des pixels de la détection, semble mieux convenir. En se basant sur les différentes informations qu'il est possible d'extraire d'une détection et que nous avons présentées précédemment, nous avons décidé de faire intervenir :

— le temps : t
— la position absolue des pixels dans la frame (x, y)
— les valeurs des canaux couleurs (r, g, b) (ou dans un autre système colorimétrique)
— la valeur du flot optique (u, v)

Une détection sera alors représentée par une loi normale de dimension 6 (t, x, y, r, g, b) et le flot optique utilisé pour estimer la position à un temps différent de celui de la détection (cf section 4.1.4.2). Plus formellement, nous définissons l'observation $\boldsymbol{z}_i = (t, x, y, r, g, b, u, v)$ d'une détection i par une loi normale de la façon suivante :

$$\boldsymbol{z}_i \sim \mathcal{N}\left(\begin{pmatrix} t_i \\ \boldsymbol{x}_i \\ \boldsymbol{a}_i \end{pmatrix}, \Sigma^i_{t\boldsymbol{xa}}\right) \tag{4.22}$$

où $\boldsymbol{x}_i = (x_i, y_i)^\top$ et $\boldsymbol{a}_i = (r_i, g_i, b_i)^\top$. À un temps t_e différent de t_i, la détection sera représentée à l'aide de son flot optique par :

$$\boldsymbol{z}_i, t_e \sim \mathcal{N}\left(\begin{pmatrix} t_i \\ \boldsymbol{x}_i \\ \boldsymbol{a}_i \end{pmatrix} + (t_e - t_i)\begin{pmatrix} 1 \\ \boldsymbol{of}_i \\ 0_{3\times1} \end{pmatrix}, \Sigma^i_{t\boldsymbol{xa}} + (t_e - t_i)^2 \begin{pmatrix} 0 & 0_{1\times2} & 0_{1\times3} \\ 0_{2\times1} & \Sigma^i_{uv} & 0_{2\times3} \\ 0_{3\times1} & 0_{3\times2} & 0_{3\times3} \end{pmatrix}\right) \tag{4.23}$$

où :

$$\Sigma^i_{t\boldsymbol{xa}} = \begin{pmatrix} \sigma_t^2 & 0_{1\times5} \\ 0_{5\times1} & \Sigma^i_{\boldsymbol{xa}} \end{pmatrix} \tag{4.24}$$

et $\Sigma^i_{\boldsymbol{xa}}$ s'estime comme sont classiquement estimées les matrices de covariance d'images (cf 4.1.2.3). La moyenne de cette loi normale sera notée $\mu^i_{t\boldsymbol{xa}}(t_e)$ et sa matrice de covariance : $\Sigma^i_{t\boldsymbol{xa}}(t_e)$.

Tout comme pour les autres similarités définies à partir des lois régissant les différents éléments d'une observation, le calcul des similarités entre les différentes détections est fait en utilisant la distance de Hellinger entre leurs lois de probabilités. Ainsi, la similarité prenant en compte le temps, la position, l'apparence et le flot optique, sera notée $s_{t\boldsymbol{xaof}}(i, j)$ et définie comme l'opposé de la distance de Hellinger entre les lois

$$s_{t\boldsymbol{xaof}}(i, j) = 1 - D_H\left(\mathcal{N}(\mu^i_{t\boldsymbol{xa}}(t_e), \Sigma^i_{t\boldsymbol{xa}}(t_e))\ ,\ \mathcal{N}(\mu^j_{t\boldsymbol{xa}}(t_e), \Sigma^j_{t\boldsymbol{xa}}(t_e))\right) \tag{4.25}$$

où $t_e = \frac{t_j - t_i}{2}$. L'un des avantages de cette représentation est qu'elle ne fait intervenir qu'un paramètre à estimer : σ_t, l'écart-type représentant la précision de la localisation temporelle de la détection.

En utilisant une seule loi normale multivariée, il est important de noter que l'on ne représente les différents aspects que par un seul mode, ce qui est faible, notamment pour la couleur. Pour réussir à représenter différents modes, nous avons tenté de filtrer par des gaussiennes positionnées régulièrement sur l'intervalle de valeurs des caractéristiques. Ainsi, en prenant l'exemple du canal rouge, il n'y aurait plus une seule valeur r de rouge mais N valeurs r_i correspondant à la proximité de r avec la i-ème référence de rouge. Cette proximité peut être construite par la réponse à un filtre gaussien centré en i/N et d'écart-type proportionnel à $1/N$. On peut aussi envisager une estimation de la densité par des noyaux (*Kernel Density Estimation*) ou encore d'utiliser la Théorie des Copules qui donne un cadre général aux corrélations possibles entre variables. Or, par l'usage, ces deux approches font

intervenir des complexités calculatoire assez importantes notamment quand la dimension s'élève et demande beaucoup d'observations. Une autre manière de voir consisterait à utiliser une représentation non-paramétrique, *ie* estimer la "distance" entre deux détections en comparant directement les deux ensembles d'observations. Mais là aussi, les temps de calculs deviennent vite rédhibitoires. C'est pourquoi, même si en pratique la loi régissant (t, x, y, r, g, b) n'est pas une loi normale, elle présente tout de même un intérêt.

Dans le cadre des travaux présentés, l'ajout d'une description d'apparence plus complète n'a pas été testé. Comme pour ce qui est fait avec les descriptions par les simples matrices de covariance, il est tout à fait envisageable de rajouter à (r, g, b) les gradients en intensité lumineuse (ou même couleurs) et les orientations. Cela pourrait encore accroître la robustesse sans faire intervenir de paramétrage particulier à ajuster.

4.2.3. Approche spectrale

L'approche spectrale est plutôt jeune mais commence à être bien reconnue dans le domaine du partitionnement de données. Elle se base sur le graphe des inter-distances entre les objets et, à l'aide d'algèbre linéaire, permet de placer les données dans un espace où les regroupements sont plus discernables. Partant de cette transformation des données, des algorithmes classiques de partitionnement sont employés. La théorie fondatrice du partitionnement spectral est complexe et peut être interprétée de manières très diverses. Par la suite, nous prendrons le point de vue de la théorie des graphes pour expliquer comment nous avons appliqué ce principe à notre problématique. Nous n'avons pas la prétention de décrire précisément la théorie de la méthode, cette section décrira simplement le procédé employé. Pour plus de détails nous renvoyons à la section 2.3.2.1, au tutoriel de U. Von Luxburg 2007 et aux travaux de A. Ng et al. 2002.

Pour appliquer le principe du partitionnement spectral, il faut premièrement représenter les données comme les nœuds d'un graphe dont les arcs (non-orientés) sont pondérés par les similarités entre les différentes données. Dans le cadre du partitionnement de détections issues d'une vidéo, un graphe similaire a déjà été construit (cf. chapitre 3). Ce graphe peut très bien être employé dans le cadre du partitionnement spectral, à condition de prendre les arcs non-orientés.

En prenant $G = (\mathcal{V}, \mathcal{E})$ le graphe construit à partir des n détections d'une vidéo, et supposant une matrice des similarités quelconque (notée $W = (w_{ij})_{i,j} \in \{1, ..., n\}$), la matrice laplacienne de G se définira de la façon suivante :

$$L = D - W \tag{4.26}$$

où D est une matrice diagonale dont chaque élément d_i correspond au degré du nœud $v_i \in \mathcal{V}$:

$$d_i = \sum_{j=1}^{n} w_{ij} \tag{4.27}$$

Ayant la matrice laplacienne L, plusieurs normalisations sont possibles. Elles conduisent à différents algorithmes. La normalisation de A. Ng et al. 2002 a ici été choisie, elle peut se définir de la façon suivante :

$$L_{sym} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \tag{4.28}$$

Partant de cette matrice, la recherche des k plus petites valeurs propres permet de construire un espace de représentation des données qui soit plus apte au regroupement.

L'algorithme 3 décrit l'ensemble de la procédure. Cet algorithme nécessite de renseigner le nombre k de valeurs propres à sélectionner qui est en relation avec le nombre de parties. Le nombre de parties peut être spécifié par l'algorithme utilisé au final pour le partitionnement, mais un bon choix de k proche de ce nombre permet d'obtenir de meilleurs résultats.

Algorithme 3 : Algorithme décrivant l'approche spectrale pour la construction des similarités (méthode de A. Ng et al. 2002).

Entrées : matrice des similarités entre les détections $S \in \mathbb{R}^{n \times n}$ et k nombre de valeurs propres sélectionnées

Calculer la matrice laplacienne normalisée L_{sym} ;
Calculer les k premiers vecteurs propres $u_1, ..., u_k$ de L_{sym} ;
Construire $U \in \mathbb{R}^{n \times k}$ contenant les vecteurs u_i en colonnes ;
Normaliser U pour obtenir T : $t_{ij} = \frac{u_{ij}}{\sqrt{\sum_{l=1}^{k} u_{il}^2}}$;
Construire les vecteurs $y_i \in \mathbb{R}^k$ pour $i \in \{1, ...n\}$;
Appliquer un algorithme de partitionnement pour classer les données décrites par leur vecteur y_i ;

Afin de décrire brièvement et de manière simpliste le pourquoi de la matrice laplacienne, on peut dire qu'il existe un lien entre la minimisation des coupes du graphe construisant un partitionnement des nœuds et l'espace construit par les vecteurs propres associés aux faibles valeurs propres de la matrice laplacienne. Plus précisément, ce lien provient de la relaxation du problème de minimisation de coupes. Relâcher ce problème revient à rechercher une matrice $H \in \mathbb{R}^{n \times k}$ indiquant l'appartenance d'un nœud à l'une des k parties d'une partition de $\mathcal{V}$. A. Ng et al. 2002 ont montré qu'en prenant pour H la matrice U de l'algorithme 3 (constituée de vecteurs propres), la solution est proche du minimum du problème relâché. Ainsi la recherche de vecteurs propres permet d'approcher le problème de la recherche d'une partition de $\mathcal{V}$ minimisant la somme "normalisée" des poids intervenant aux arrêtes des coupes.

Finalement, cette approche spectrale permet de se placer dans un espace où les regroupements des détections sont mieux discernables. Ensuite, il est possible d'utiliser soit un algorithme de type *k-means* sur cet espace ou encore de reconstruire une matrice des similarités issue des distances entre les éléments, en considérant leurs coordonnées basées sur les vecteurs propres. Quelques expérimentations de l'application de cette méthode avec différents algorithmes de partitionnement sont présentées à la section 5.1.1 du chapitre des expérimentations.

4.2.4. Estimation des similarités par *ensemble clustering*

Cette section présente une approche permettant d'estimer la probabilité ($P_{link}(z_i|z_j)$) de lier directement les détections i et j au sein d'une trajectoire. Comme expliqué précédemment, une des principales difficultés dans le problème de l'estimation des similarités est l'hétérogénéité des grandeurs décrivant une détection. Et derrière ce problème se cache la question de l'ajustement des pondérations que l'on pourrait donner à chacune des grandeurs (temps, position, apparence ...). Plutôt que de représenter P_{link} par une conjonction de lois

paramétriques et chercher à estimer ses paramètres, il peut être intéressant d'estimer directement la valeur de $P_{link}(z_i|z_j)$ en se basant sur les partitionnements que l'on peut obtenir en prenant chacune des grandeurs séparément.

Cela conduit à une classe de problèmes de partitionnement que sont les *cluster ensembles* (parfois appelés *consensus clustering*) tels que décrits par A. Strehl et J. Ghosh 2003. Ce sont des approches assez récentes qui cherchent à combiner un ensemble de partitionnements (partitionnements faibles) pour aboutir à un partitionnement plus robuste. En considérant chaque type de similarité (temps, position, apparence ...) séparément et en recherchant des partitionnements sur ces similarités, on obtient des partitionnements différents. Ceux-ci peuvent ensuite servir de base à un algorithme de type *cluster ensemble*. C'est ce principe qui a motivé l'utilisation de techniques d'*ensemble clustering*.

L'objectif du *cluster ensemble* tel que défini par A. Strehl et J. Ghosh 2003 est de trouver le partitionnement qui soit le plus proche en moyenne de l'ensemble des partitionnements faibles. Plus précisément, il correspond au partitionnement dont la moyenne des informations mutuelles (entre lui-même et chacun des partitionnements faibles) est minimale. Étant donnée la complexité combinatoire intervenant dans la recherche exhaustive directe de ce partitionnement idéal, les auteurs de A. Strehl et J. Ghosh 2003 ont mis en avant plusieurs heuristiques.

Parmi ces algorithmes, qui permettent uniquement de trouver une solution approchée, le *Cluster-based Similarity Partitioning Algorithm* (CSPA) paraît utilisable parce qu'il présente de bons résultats tout en étant relativement facile à mettre en œuvre. Cette heuristique se base sur la construction d'une matrice des similarités élaborée à partir d'un ensemble de matrices des similarités issues de partitionnements "faibles". Concrètement, la matrice des similarités finale représente le nombre de fois qu'une paire d'éléments a été regroupée par les partitionnements faibles. Une fois cette matrice construite, un algorithme standard de partitionnement peut être appliqué avec celle-ci. Cependant la principale difficulté réside dans la définition des partitionnements faibles, de manière à ce qu'ils soient les plus représentatifs du problème.

Dans notre cas, il n'est pas intéressant d'utiliser un algorithme générique de partitionnement, mais plutôt de bien prendre en compte que les groupes constituent des trajectoires, en ne faisant intervenir que les transitions correspondant aux trajectoires. Pour cela, il est convenable d'utiliser un algorithme dans la même idée que celui écrit au chapitre précédent.

Plus formellement, la probabilité de transition est définie par la matrice des similarités estimées $\hat{S}$:

$$P_{link}(j|i) = \hat{S}(i,j) = \frac{1}{|\mathcal{T}_{sample}|} \sum_{T \in \mathcal{T}_{sample}} s_T(i,j) \tag{4.29}$$

où s_T correspond à une matrice de similarités construite à partir d'un partitionnement faible T de la façon suivante :

$$s_T(i,j) = \begin{cases} 1 & \text{if } \exists k \mid T_k(i+1) = T_k(j) \\ 0 & \text{else} \end{cases} \tag{4.30}$$

et $\mathcal{T}_{sample}$ correspond à l'ensemble des partitionnements faibles ; sa construction sera décrite par la suite.

Notons que $\hat{S}(i,j)$ est bien assimilable à une probabilité $P_{link}(j|i)$ de passer en j étant en i car :

1. $\hat{S}(i,j) \in [0,1]$ grâce à la normalisation par $|\mathcal{T}_{sample}|$
2. $\sum_j \hat{S}(i,j) = 1$ car :

$$\sum_j \hat{S}(i,j) = \sum_j \frac{1}{|\mathcal{T}_{sample}|} \sum_{t \in \mathcal{T}_{sample}} s_t(i,j) \tag{4.31}$$

$$= \frac{1}{|\mathcal{T}_{sample}|} \sum_{t \in \mathcal{T}_{sample}} \sum_j s_t(i,j) \tag{4.32}$$

$$= \frac{1}{|\mathcal{T}_{sample}|} \sum_{t \in \mathcal{T}_{sample}} 1 = 1 \tag{4.33}$$

Le point crucial est maintenant de réussir à extraire un ensemble des partitionnements "faibles", qui soit représentatif du problème tout en restant vraisemblable (en n'allant pas chercher des solutions trop extrêmes). Pour l'aspect représentatif, différents types de similarités et différents nombres de trajectoires ont été envisagés, et pour ce qui de l'aspect vraisemblable, une méthode permettant d'obtenir une solution "optimale" a été développée pour un nombre de groupes et un type de similarité donné.

4.2.4.1. Algorithme d'estimation des similarités

Avec une matrice de dissimilarité d contenant les dissimilarités pour chaque paire de détections, et un nombre C de trajectoires fixé, le partitionnement $U_C^d \in \mathcal{T}$ minimise la somme des distances issues des transitions. Plus exactement, le partitionnement U_C^d, considéré comme optimal au vu de d (matrice $D \times D$ des distances) et de C (nombre de trajectoires), se définit de la manière suivante :

$$U_C^d = \underset{T \in \mathcal{T}, |T|=C}{\arg\min} \sum_{T_k \in T} \sum_{i=2}^{|T_k|} d(T_k^{i-1}, T_k^i) \tag{4.34}$$

où T_k^i est l'indice correspondant au i-ème élément de la k-ième trajectoire de l'ensemble T de trajectoires. Défini de cette façon, U_C^d peut être interprété comme le maximum de vraisemblance, en supposant avoir un nombre de trajectoires fixé et une matrice de similarités donnée. Cette vraisemblance reste proche de celle intervenant dans l'*a posteriori* défini au chapitre précédent. Cela permettra de se baser sur la méthode de résolution utilisant les circulations de flots sur un graphe pour obtenir U_C^d. Cette méthode reprend le principe décrit à la section 3.2 du chapitre précédent et permettra de construire l'ensemble $\mathcal{T}_{sample}$ des partitionnements faibles utilisés par l'algorithme d'*ensemble clustering*.

En reprenant les notations définies au chapitre précédent (cf. section 3.2.1), les détections seront représentées par un graphe G sur lequel circule un flot binaire. Si le coût d'une transition $i \to j$ est fixé par la distance $d(i,j)$ envisagée, et que l'on impose d'avoir une quantité de flot circulant par nœud représentant une détection, le coût d'un flot représente bien ce que l'on cherche à minimiser à l'équation 4.34. En ayant une demande de flot valant C (nombre de trajectoires demandé) à la source du graphe, le flot de coût minimal sélectionne bien les arcs permettant de définir U_c^d. La figure 4.7 illustre la circulation de flot employée pour trouver un partitionnement U_c^d optimal. L'algorithme 4 détaille la procédure utilisée

Algorithme 4 : Algorithme permettant d'obtenir un partitionnement P optimal avec un nombre de parties C fixé et une matrice des distances d donnée. Cet algorithme sera désigné par k-CC (*k-Chain Clustering*).

$V \leftarrow \emptyset$: ensemble des nœuds;
$E \leftarrow \emptyset$: ensemble des arcs;
ajouter s et t à V;
pour *détection i de $\mathcal{D}$* **faire**
    ajouter un nœud v_i à V;
    ajouter un arc $s \rightarrow v_i$ de coût 0 à E;
    ajouter un arc $v_i \rightarrow t$ de coût 0 à E;
fin
pour $v_i \in V - \{s, t\}$ **faire**
    pour $v_j \in V - \{s, t, v_i\}$ **faire**
        si $d(i, j) < \infty$ **alors**
            ajouter un arc $v_i \rightarrow v_j$ de coût $d(i, j)$ à E;
        fin
    fin
fin
initialiser la demande : $u_v \leftarrow 0$, $\forall v \in V$;
$u_s \leftarrow -C$ et $u_t \leftarrow C$;
imposer une circulation binaire sur chaque arc;
imposer une circulation de 1 pour les nœuds de $V - \{s, t\}$;
trouver le flot de coût minimal F sur le graphe $G = (V, E, u)$;
initialiser le partitionnement : $P = \{\{1\}, \{2\}, ..., \{D\}\}$;
pour $f_{ij} \in F$ **faire**
    si $f_{ij} = 1$ **alors**
        fusionner la partie de P contenant i avec celle contenant j;
    fin
fin

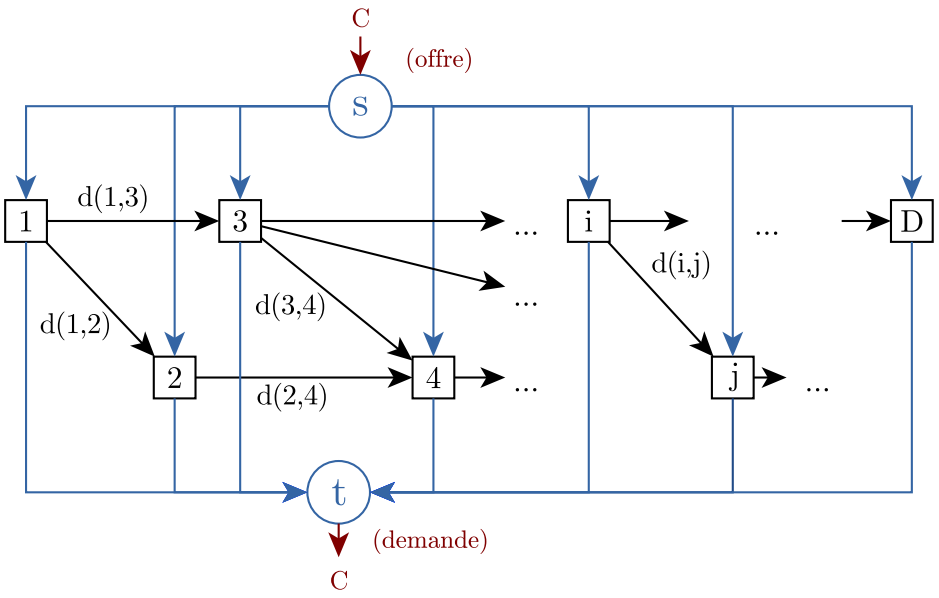

FIGURE 4.7. Graphe représentant la circulation de flot permettant d'obtenir un partitionnement optimal en ayant un nombre C de trajectoires imposé et une matrice des inter-distances d donnée. Les nœuds correspondant à des détections sont représentés par des carrés. Le flot circulant est binaire, et sur chaque nœud de détection on impose qu'il y ait une quantité de flot le traversant.

pour construire le graphe et comment le partitionnement optimal est construit à partir du flot de coût minimal.

Cet algorithme sera aussi utilisé pour la comparaison des résultats avec d'autres méthodes de partitionnement (comme le *k-means* ou *k-medoids*), il sera désigné par la suite comme l'algorithme k-CC (*k-Chain Clustering*). Notons que si l'on prend d tel que $d(i,j) = -\log P_{link}(z_i|z_j)$, l'algorithme permet de trouver un maximum de vraisemblance au sens de la définition de la vraisemblance donnée à la section 3.1.3 (sans terme $P_f(z_i)$). Cela permet d'obtenir une solution la plus vraisemblable en n'utilisant pas de modélisation de l'*a priori* mais un nombre fixé (k) de trajectoires.

Ayant ainsi un moyen d'obtenir un partitionnement optimal U_c^d pour une matrice d et un nombre de trajectoires C fixé, il est possible de construire $\mathcal{T}_{sample}$ comme un ensemble de ces partitionnements pour différents d et C.

En pratique, C prendra toutes les valeurs entières allant de 1 à C_{max}, où C_{max} est un paramètre permettant de fixer le nombre maximum de trajectoires demandé. Théoriquement, il est possible de prendre $C \in \{1, ..., D\}$, mais $C_{max} < D$ est utilisé pour éviter d'envisager des solutions contenant un nombre excessif de trajectoires.

Il faut ensuite définir quel va être l'ensemble des matrices de distances (noté $\mathcal{H}_{sample}$). Pour construire ces matrices, principalement quatre dissimilarités ont été choisies : d_{ts} pour rendre compte du temps, de la position et de la taille des détections, d_a pour l'apparence, d_{of} pour le mouvement basé sur une estimation du flot optique. D'autres distances seront aussi testées et présentées dans le chapitre des expérimentations.

L'algorithme 5 décrit la procédure employée pour calculer une estimation de la matrice $\hat{S}$ des similarités.

Algorithme 5 : Algorithme d'estimation des probabilités de transition.

$N \leftarrow 0$;
$S \leftarrow 0$;
pour $d \in \mathcal{H}_{sample}$ **faire**
  pour $c \in \{1, ..., C_{max}\}$ **faire**
    rechercher $\hat{u}_c^d$ *via* l'algorithme 4 permettant de trouver :

$$\hat{u}_c^d = \underset{T \in \mathcal{T}, |T|=C}{\arg\min} \sum_{T_k \in T} \sum_{i=2}^{|T_k|} d(T_k^{i-1}, T_k^i)$$

    si $\hat{u}_c^d$ *est trouvé* **alors**
      $N \leftarrow N + 1$;
      pour *transition* $i \rightarrow j$ *de* $\hat{u}_c^d$ **faire**
        $s(i,j) \leftarrow s(i,j) + 1$;
      fin
    fin
  fin
fin
$\hat{S} \leftarrow \frac{S}{N}$;

À titre d'exemple, la figure 4.8 montre une matrice des similarités estimée à partir de trois similarités en entrée. Les matrices présentent les similarités entre les détections. Comme elles sont triées par ordre d'apparition des détections dans la vidéo, elles sont triangulaires supérieures. La matrice estimée par l'algorithme décrit précédemment représente plus fidèlement une probabilité de transition entre les détections et tire parti de la globalité des matrices de similarités initiales. Là où les matrices de similarités avaient un bloc de fortes similarités pour représenter des détections proches, la matrice estimée ne représente plus que les transitions en chaine correspondant à la trajectoire.

Cette estimation demande un temps de calcul non négligeable, mais permet d'obtenir une matrice plus creuse et donc de réduire la complexité de la recherche du MAP (le graphe où circule le flot comporte peu d'arcs).

4.3. Conclusion

Ce chapitre a exposé les représentations des différents aspects servant à décrire une détection. Pour ce qui est de l'aspect temporel, des modélisations basées sur des lois géométriques ont été employées et une expérimentation a permis de se rendre compte de leurs convenances respectives. Les descriptions de l'apparence qui paraissent les plus efficaces sont celles traitant conjointement les aspects spatiaux et colorimétriques, c'est le cas des matrices de covariance et des lois normales joignant position et couleurs. Au vu des expérimentations, l'information de mouvement parait plus accessible et exploitable par une estimation du flot optique que par du suivi visuel de chaque détection.

Pour la construction des similarités finales utilisées pour définir P_{link} et faire le lien avec la vraisemblance, plusieurs solutions sont envisagées. Une première solution consiste à joindre

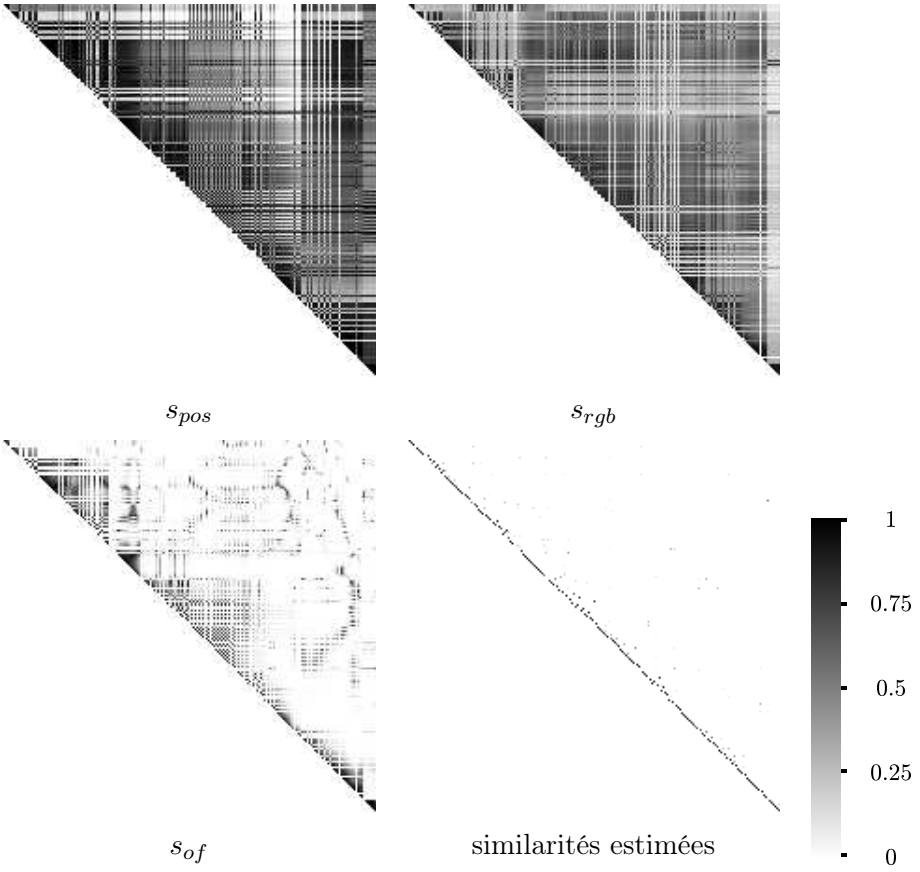

FIGURE 4.8. Exemple d'estimation d'une matrice des similarités. s_{pos}, s_{rgb} et s_{of} sont les matrices utilisées pour l'estimation de la matrice en bas à droite.

les informations temporelles, spatiales, colorimétriques et dynamiques dans une loi normale multivariée. Ensuite, des méthodes issues du partitionnement de données ont aussi pu être mises en œuvre pour construire au mieux les probabilités de transition entre les détections. Le partitionnement spectral a permis de mieux conditionner les similarités pour que les différents regroupements soient plus discernables, tandis que la méthode d'estimation (inspirée de l'*ensemble clusering*) donne une fusion de différentes similarités en estimant directement les probabilités P_{link}. Les expérimentations du chapitre suivant permettent de juger la pertinence de ces différentes solutions.

Chapitre 5.

Expérimentations et résultats

Ce chapitre présente les différentes expérimentations menées pour évaluer le système élaboré au cours de cette thèse. Elles s'appuient sur la base de test présentée à la section 2.5 et les critères de qualité définis préalablement (section 2.4.3).

En premier lieu, la stratégie de regroupement a été évaluée. Ces tests permettent de déterminer si la stratégie de regroupement par la modélisation MAP reste mieux adaptée au cas du regroupement de détections d'une vidéo que des algorithmes plus standard de partitionnement. L'approche issue du partitionnement spectral est aussi évaluée par la même occasion.

Ensuite, les différentes mesures de similarité inter-détections (qui restent un point crucial) sont comparées. Des expérimentations sont présentées pour tester la pertinence des différentes stratégies de fusion et d'estimation de ces mesures.

Dans un troisième temps, la méthode est testée dans sa globalité et d'un point de vu plus applicatif. La méthode de regroupement exposée a permis de construire des albums photos en sélectionnant des représentants et des résultats quantitatifs sont donnés sur la base de vidéos précédemment présentées.

Puis, des résultats qualitatifs de suivi multi-cibles résultant du regroupement de détections sont montrés. Ils font intervenir des cadres applicatifs très différents.

Pour finir, nous présentons un aperçu des temps de calcul nécessaires à l'exécution des différentes étapes de la méthode.

5.1. Évaluation de différentes méthodes de partitionnement

Cette section compare l'étape de partitionnement de la méthode présentée (Maximum *A Posteriori* avec résolution par flot de coût minimal) avec quelques autres algorithmes génériques de partitionnement.

L'expérimentation menée ici met en œuvre trois algorithmes de partitionnement de données s'appuyant sur les mêmes mesures de similarités. Ces similarités font intervenir le temps, la position dans l'image, l'apparence et le mouvement à l'aide du flot optique estimé. Les meilleurs performances ont été obtenues en faisant varier le paramètre jouant sur le nombre de groupes. Dans le cas du partitionnement ascendant hiérarchique, ce paramètre correspond au seuil de troncature de la hiérarchie. En le faisant varier, il est possible de tester toutes les solutions en envisageant tous les nombres de parties possibles. Pour ce qui est de la méthode *Affinity Propagation*, le paramètre de préférence (*global preference parameter*) a été employé et testé sur une large gamme de valeurs. Cet algorithme n'est pas décrit dans

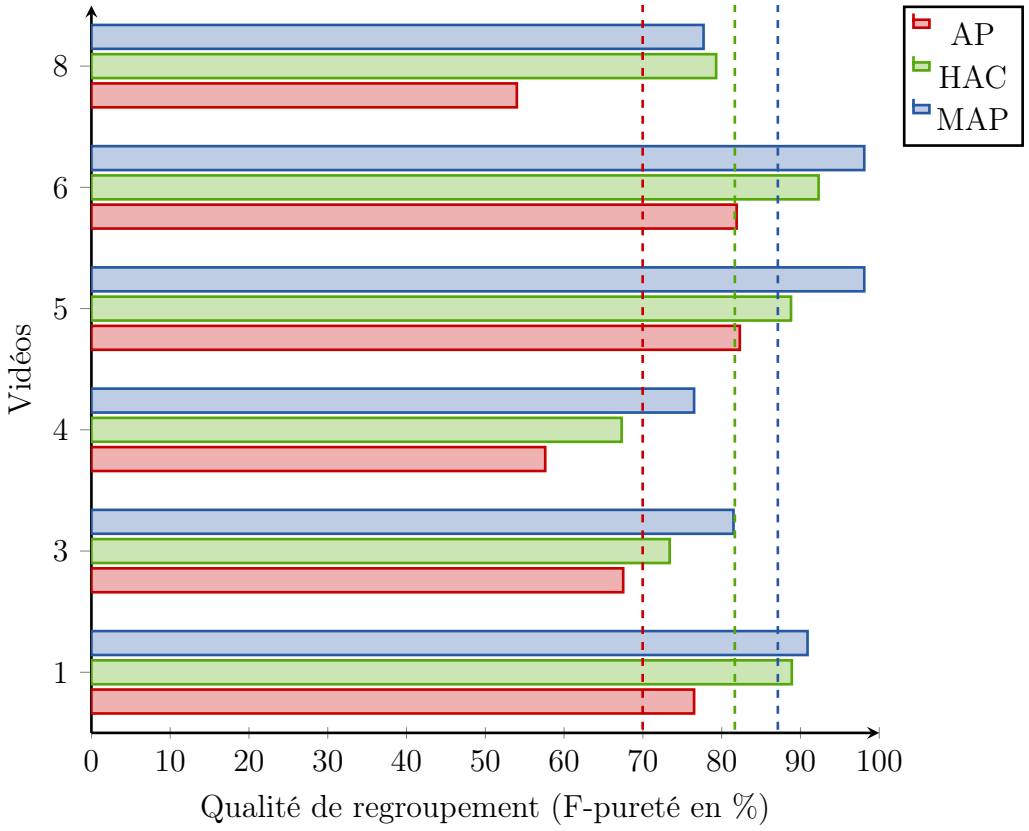

FIGURE 5.1. Meilleures performances obtenues par trois algorithmes de partitionnement en se basant sur la même matrice des similarités. *MAP* : méthode présentée au chapitre 3, *HAC* : partitionnement ascendant hiérarchique (*Hierarchical Ascendant Clustering*) et *AP* : *affinity propagation clustering*. Les lignes en pointillés représentent la moyenne des F-puretés pour les trois méthodes.

ce rapport, plus de détails sont donnés par la publication de D. DUECK et B. FREY 2007. La troisième méthode est celle décrite au chapitre 3, en utilisant la forme simplifiée de l'*a priori* et en faisant varier le paramètre p_e qui influence le nombre de regroupements obtenus. L'expérimentation a été faite sur six vidéos.

La figure 5.1 montre les meilleures performances (exprimées en F-pureté) obtenues par ces trois algorithmes de partitionnement de données en se basant sur la même matrice des similarités. Les résultats montrent que l'approche s'appuyant sur la maximisation des probabilités de transition au sein des groupes de détections, donne de meilleurs résultats que les deux autres méthodes plus génériques. Ensuite, le HAC passe devant les performances de l'*Affinity-Propagation*. Cela vient probablement du fait que le partitionnement hiérarchique (avec la stratégie *single-link* pour mesurer la distance entre groupes) souffre d'un "effet de chaine" qui peut amener à des groupes de formes très allongées. Dans notre cas, cet "effet de chaine" ne s'avère pas problématique. Au contraire, la forme de chaine est assez proche de ce qui est recherché. En effet, une détection se retrouve en général proche de deux autres détections : celle provenant de l'image précédente et celle provenant de l'image suivante. Cette représentation en chaine est d'ailleurs utilisée dans la modélisation MAP avec une représentation des trajectoires comme chaines de Markov d'ordre 1.

Les performances supérieures obtenues avec la modélisation par MAP viennent du fait que les trajectoires sont mieux représentées. Par ailleurs, la forme des regroupements effectués

correspond mieux à la réalité rencontrée dans le cas du regroupement de détections extraites d'une vidéo. Des résultats en relation avec l'approche spectrale vont aussi dans ce sens, ils sont présentés à la section 5.1.1.

5.1.1. Évaluation de l'approche spectrale

Des expérimentations ont aussi été menées pour tester le potentiel des approches dites *spectrales* (cf. 2.3.2.1). Ce type d'approches permet de se placer dans un espace où les groupes sont plus aisément identifiables tout en se basant sur les similarités entre les objets et la structure du graphe liant les objets. La méthode est décrite en détails à la section 4.2.3.

Afin de montrer, par l'expérimentation, l'impact de l'approche spectrale, quatre algorithmes différents ont été testés avec ou sans l'étape spectrale appliquée à la matrice des similarités. La fonction de similarité utilisée pour cette matrice est celle utilisant une loi normale joignant la position (x,y) et la couleur (r,g,b), avec en plus le flot optique pour estimer le mouvement. La définition de la similarité est donnée par $s_{\boldsymbol{txaof}}(.,.)$ à l'équation 4.25 de la section 4.2.2.

Les quatre algorithmes utilisés sont les suivants :

— HAC : partitionnement ascendant hiérarchique avec *single-linkage* et choix du nombre de groupes pour découper la hiérarchie
— *k-means* : algorithme des k moyennes
— *k-medoids* : algorithme des k médianes (variante de *k-means* ne calculant pas de moyenne mais prenant un élément comme centre de classe)
— *kcc* : algorithme de recherche du partitionnement par chaines optimal avec consigne sur le nombre de chaines (algorithme 4 section 4.2.4.1).

Ces algorithmes prennent en entrée une matrice des similarités (excepté le *k-means* dont on n'utilisera que la version spectrale) et un nombre de groupes k. Pour ces expérimentations, tous les k possibles (de 1 à D) ont été envisagés. Le nombre de groupes (noté $k^\star$) permettant d'obtenir le meilleur résultat (meilleure F-pureté) est sélectionné, et ensuite les k de $k^\star - 10$ à $k^\star + 10$ sont pris pour calculer la moyenne et l'écart-type des F-puretés obtenues. Cela permet de rendre compte du potentiel de chacune des méthodes tout en ayant un aperçu de leur sensibilité à k grâce à l'écart-type.

Les versions spectrales des algorithmes HAC et k-medoids font intervenir les distances euclidiennes entre vecteurs descripteurs issus des vecteurs propres (cf. section 4). Le *k-means* spectral fait directement intervenir ces vecteurs descripteurs et le k-CC prend une normalisation linéaire des distances entre vecteurs descripteurs afin d'obtenir une similarité dans $[0, 1]$. Le nombre de valeurs propres sélectionnées au niveau de l'étape spectrale a été arbitrairement fixé à 25 pour toutes ces expérimentations. Le choix de ce nombre peut influencer la qualité des résultats. L'impact de ce choix n'est pas présenté dans le cadre des expérimentations, mais il peut être important de s'en soucier.

Pour les algorithmes *k-means* et *k-medoids*, les k centres initiaux sont choisis aléatoirement et uniformément parmi les détections à classer. Ces algorithmes sont exécutés avec 10 initialisations différentes pour chaque valeur k envisagée, les F-puretés obtenues sont ensuite moyennées. Comme l'algorithme *k-means* ne permet pas de travailler directement sur des similarités, mais nécessite un espace où des moyennes soient calculables, la version non-spectrale n'a pas pu être testée.

D'après les résultats résumés à la figure 5.2, on peut voir que l'approche spectrale permet, en moyenne, d'avoir une qualité de regroupement légèrement supérieure à l'approche simple,

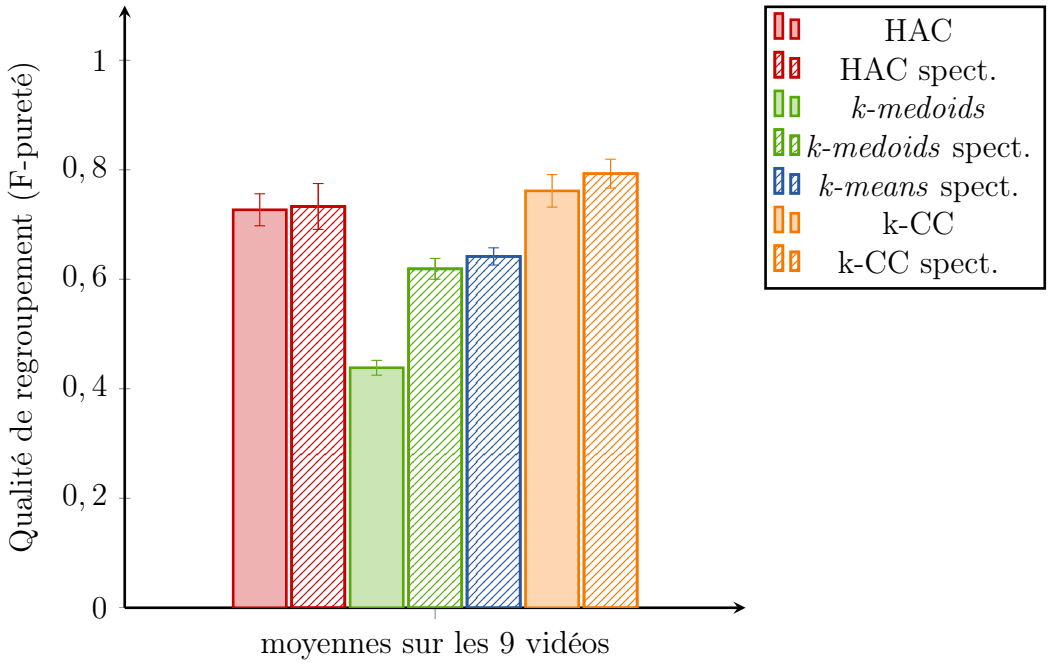

FIGURE 5.2. Résumé de l'apport de l'approche spectrale appliquée à différents algorithmes. Les moyennes et moyennes des écarts-types sur les 9 vidéos sont présentés. Les barres hachurées représentent les moyennes des résultats obtenus en ajoutant l'étape spectrale avant l'exécution des différents algorithmes.

et cela pour les trois algorithmes concernés (HAC, *k-medoids* et *k*-CC). Toutefois, compte tenu des écarts-types importants dûs à la variabilité des situations représentées par les neuf vidéos, il faut prendre les résultats avec précaution. Même si l'approche spectrale améliore considérablement les résultats sur les algorithmes de type k-moyenne, ils restent en dessous de ce que l'on peut obtenir avec les deux autres méthodes.

Pour ce qui est du partitionnement hiérarchique, il concurrence assez bien l'approche des regroupements en chaines utilisée par la méthode du MAP, mais les résultats restent très variables selon les situations. L'impact de l'approche spectrale ne se voit pas réellement sur le HAC. La méthode *k*-CC présente en moyenne des résultats légèrement supérieurs à ceux du HAC et l'approche spectrale permet un léger mieux sur les performances. On peut noter aussi une plus grande stabilité des résultats de l'approche *k*-CC par rapport au HAC et elle est encore un peu accrue avec l'approche spectrale. Le fait que le *k*-CC passe devant le HAC, qui lui-même devance les méthodes de type k-moyennes, s'explique par le fait que les formes des groupes construits soient différentes. Dans le cas du regroupement de détections issues de vidéos, les groupes ont naturellement une forme de chaine, ceci venant de l'aspect temporel de la vidéo. Comme les k-moyennes donnent des formes plutôt convexes, il est normal que leurs résultats soient en dessous du *k*-CC qui modélise justement les groupes par des chaines de similarités. Toutefois, l'approche spectrale pallie quelque peu cette forme inadaptée des groupes. Le HAC (notamment avec la stratégie *single-linkage*) favorise les formes chainées, mais le *k*-CC se rapproche plus de ce qui est rencontré en pratique, ce qui est aussi reflété par les résultats.

Les résultats plus détaillés de la figure 5.3 montrent les performances sur chaque vidéo prise séparément. À première vue, les résultats présentent une forte disparité de performances en fonction de la situation rencontrée sur les différentes vidéos. Par exemple, en

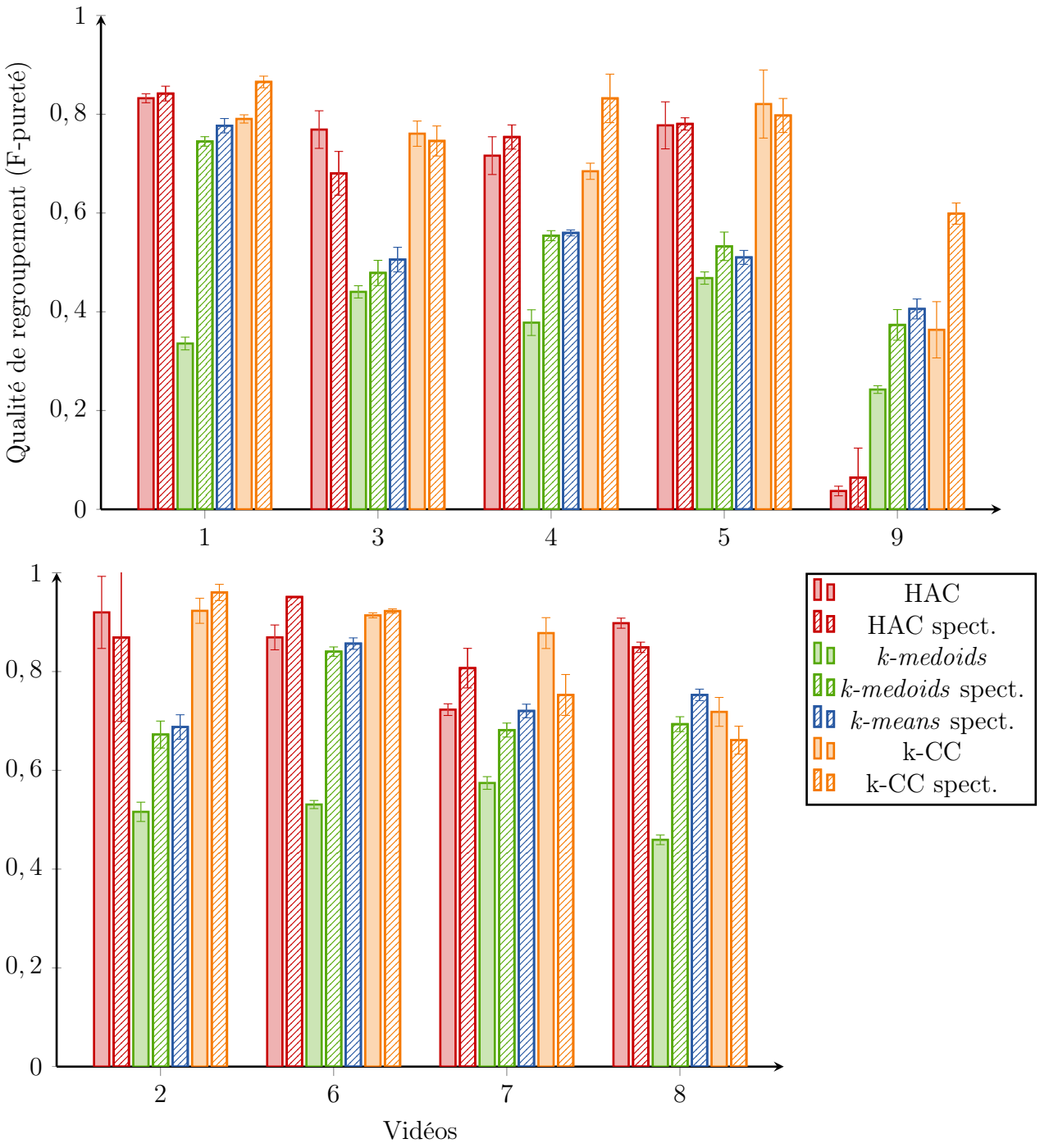

FIGURE 5.3. Apport de l'approche spectrale appliquée à différents algorithmes. Les barres hachurées représentent les résultats obtenus en ajoutant l'étape spectrale avant l'exécution des différents algorithmes.

ne comparant que les HAC et les k-CC des vidéos 8 et 9, on ne peut dire qu'une méthode soit meilleure qu'une autre : le HAC surpasse le k-CC dans le premier cas alors que c'est l'inverse pour l'autre vidéo. Cela illustre bien les difficultés à sélectionner une seule méthode qui fonctionne dans tous les cas rencontrés en vidéosurveillance. Cette différence de résultats peut s'expliquer par le fait que ces deux vidéos (8 et 9) présentent les situations les plus différentes (cf. table 2.2). Une autre explication est que la similarité utilisée dans le cadre de

ces expérimentations est moins adaptée à la situation de la vidéo 8 qu'aux autres situations. La variation d'illumination est particulièrement importante (ce qui impacte aussi la colorimétrie), les tailles des détections varient aussi beaucoup et l'on observe peu de changements de direction. Cela peut expliquer que la similarité joignant colorimétrie RGB et position avec flot optique soit moins adaptée à cette situation et que les résultats auraient pu être différents avec une similarité différente.

Pour résumer, on peut dire qu'en moyenne, l'approche spectrale améliore bien les résultats obtenus avec les différents algorithmes l'utilisant. Cette expérimentation montre aussi que la modélisation des groupes avec des chaines de similarités (par l'algorithme k-CC) est en moyenne plus intéressante, même si le HAC peut très bien la concurrencer. Toutefois, il est important de souligner la variabilité des résultats en fonction de la scène rencontrée. Ces résultats illustrent bien la principale difficulté liée au domaine de la vidéosurveillance en environnement non contrôlé : l'évaluation sur la grande variété des situations rencontrées en réalité.

5.2. Évaluation de la fusion des similarités

Les similarités comparées ici font intervenir l'aspect temporel, les positions et tailles dans l'image, la colorimétrie (RGB) et le flot optique afin d'avoir aussi une information de mouvement.

Voici les quatre similarités employées pour le comparatif :

1. Moyenne des similarités de Hellinger :

$$s_H(i,j) = \frac{1}{4}(P^g_{time}(t_j|t_i) + s_{pos}(i,j) + s_{rgb}(i,j) + s_{of}(i,j)) \tag{5.1}$$

2. Similarité s_H avec étape de normalisation du partitionnement spectral (avec 25 valeurs propres).
3. Estimation des similarités avec s_{pos}, s_{rgb} et s_{of}. Cette similarité sera notée s_e. L'aspect temporel est pris en compte en multipliant la similarité estimée par P^g_{time}.
4. Similarité issue de la distance de Hellinger entre les lois jointes : $s_{t\boldsymbol{xaof}}$.

Pour chacune des vidéos et pour les quatre similarités décrites, l'algorithme basé sur la modélisation MAP a été exécuté 10 fois avec différents paramètres p_e. L'*a priori* utilisé est le binomial (cf. définition de P_{start} équation 3.3) et le paramètre p_e a couvert 10 valeurs proches du ratio $\frac{\text{nombre de trajectoires}}{\text{nombre de détections}}$ issu de la vérité-terrain. Ces valeurs sont placées entre -1% et $+1\%$ du ratio de la vérité-terrain et espacées de 0.1%. Cela a permis de faire figurer moyennes et écarts-types des F-puretés pour les neuf vidéos. Ces résultats sont présentés à la figure 5.4.

La lecture des résultats montre que les performances entre les quatre stratégies de fusion des similarités sont très proches. Et, d'une vidéo à l'autre, les conclusions diffèrent quant à la performance des quatre approches.

La simple moyenne (s_H) entre les similarités correspondant aux différentes caractéristiques semble être légèrement au-dessus des autres en moyenne. Cela montre que les similarités sous-jacentes sont par nature bien représentatives des liens entre les détections.

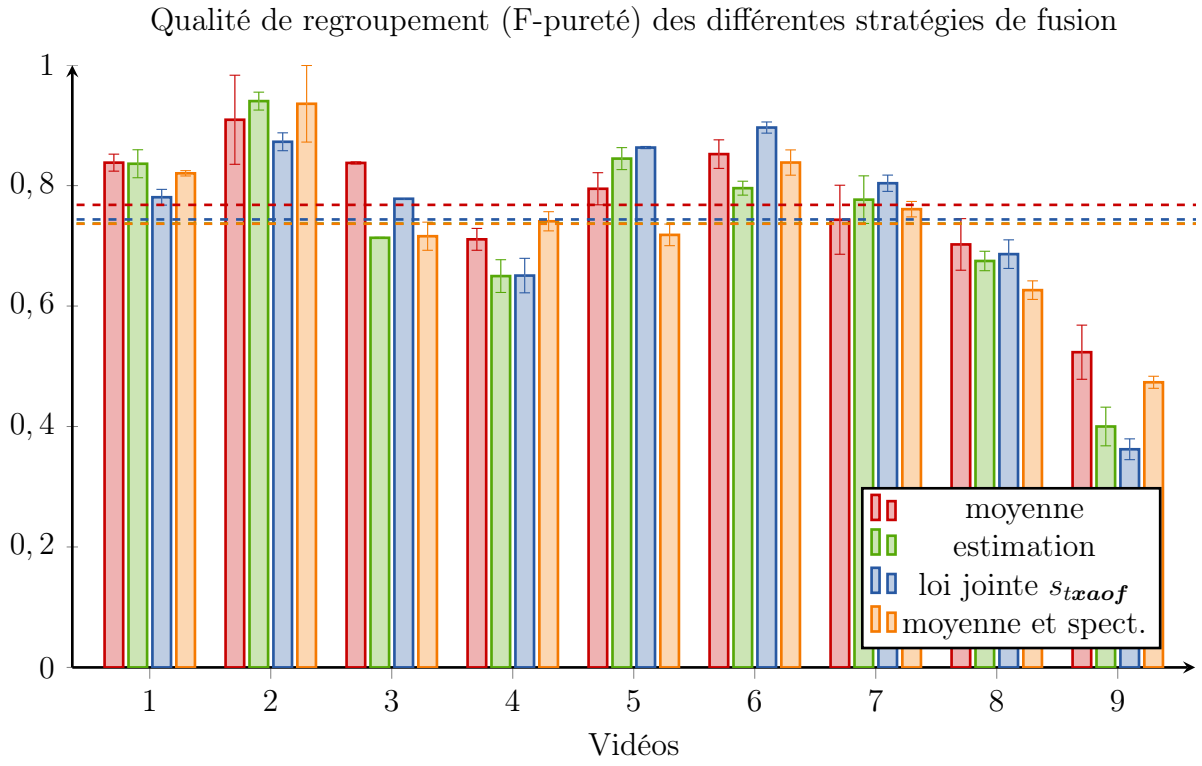

FIGURE 5.4. Comparatif des différents types de fusion. Les différentes similarités utilisées se basent sur les caractéristiques temporelles, spatiales, colorimétriques (moyennes RGB) et de mouvement (flot optique). *moyenne* : similarité s_H, *estimation* : similarité s_e, *loi jointe* : similarité s_{txaof} et *moyenne et spect.* : similarité s_H avec normalisation spectrale. Les moyennes sur les neuf vidéos des quatre cas figurent en pointillés sur le graphique (*estimation* et *moyenne et spect.* sont quasiment superposées).

5.2.1. Évaluation de l'estimation des similarités

Afin de tester la pertinence de l'estimation des similarités présentée à la section 4.2.4, l'algorithme d'estimation a été employé pour extraire des similarités fusionnant plusieurs distances. Dans le cadre de cette expérimentation, des dissimilarités relativement brutes ont été prises sur les détections. Voici les différentes distances utilisées :

position et taille : distance en position entre les détections i et j :

$$d_P(i,j) = \left|\left| \begin{pmatrix} \boldsymbol{x}_i \\ \boldsymbol{s}_i \end{pmatrix} - \begin{pmatrix} \boldsymbol{x}_j \\ \boldsymbol{s}_j \end{pmatrix} \right|\right| \tag{5.2}$$

apparence : la distance en apparence correspond à la distance de Bhattacharyya des histogrammes HS-V (notés ici $\boldsymbol{a}_i$ et $\boldsymbol{a}_j$) des détections i et j :

$$d_A(i,j) = D_B(\boldsymbol{a}_i, \boldsymbol{a}_j) \tag{5.3}$$

flot optique : cette distance ressemble à celle définie précédemment (section 4.1.4.2) mais utilise simplement une distance euclidienne en position :

$$d_F(i,j) = ||\boldsymbol{x}_i + \Delta t \boldsymbol{of}_i - (\boldsymbol{x}_j + \Delta t \boldsymbol{of}_j)|| \tag{5.4}$$

ZNCC : la distance ZNCC (notée ici d_Z) entre les pixels des détections i et j est calculée après redimensionnement des images des détections à une taille standard (50×50

pixels). Elle se définit à partir de la corrélation croisée centrée normalisée entre les pixels pris en niveaux de gris.

Ces différentes distances ont été utilisées par l'algorithme d'estimation des similarités (algorithme 5 section 4.2.4.1). Pour évaluer l'impact de ces distances sur la solution obtenue, quatre combinaisons de ces distances ont été testées :

— PA : la similarité entre les détections ne fait pas intervenir l'algorithme d'estimation et la position (d_P) et l'apparence (d_A) sont associées de la façon suivante :

$$\frac{1}{2}(e^{-d_P(i,j)} + e^{-d_A(i,j)}) \tag{5.5}$$

— EPA : les distances en position (d_P) et en apparence (d_A) sont employées pour l'estimation des similarités.

— EPAF : distance similaire à *EPA* avec la distance flot optique (d_F) en plus.

— EPAFZ : distance similaire à *EPAZ* avec la distance ZNCC (d_Z) en plus.

L'aspect temporel a été traité à part, il n'a pas été intégré dans la procédure d'estimation. En effet, l'aspect temporel donne une information particulière, et l'utilisation de l'algorithme uniquement sur cette information donne des résultats trop difficilement exploitables. Le temps a été modélisé par une loi géométrique (voir définition de P^g_{time} à l'équation 4.3 de la section 4.1.1) avec comme paramètre p_t fixé à 0.9. Le temps est traité de manière indépendante de l'estimation de la fonction de similarité. Il est intégré à la vraisemblance multipliant les similarités estimées par la probabilité P^g_{time} correspondante.

La figure 5.5 illustre l'apport de l'estimation des similarités sur les cas des vidéos 1 à 7. Ces résultats ont été obtenus par l'algorithme basé sur la modélisation MAP avec les versions simples de l'*a priori* pour le nombre de trajectoires et le taux de faux positifs (cf. équations 3.6 et 3.5 de la section 3.1.2). Le taux de faux positifs (β) a été fixé à 1% et le paramètre p_e de l'*a priori* sur le nombre de trajectoires a varié sur 100 valeurs de 0.02 à 0.2. Cette variation du paramètre p_e a permis de donner des résultats en moyenne et de faire figurer un écart-type des performances. Celles-ci sont indiquées en F-puretés.

Pour ce qui est du paramétrage de l'algorithme d'estimation des similarités, le seul paramètre restant à fixer est C_{max} correspondant au nombre maximal de trajectoires à envisager. Il a été fixé arbitrairement à 50 pour toutes les vidéos.

La table 5.1 montre un résumé des résultats sur les sept vidéos. La première ligne de résultats donne une moyenne des F-puretés. Elle confirme que l'estimation des similarité améliore la qualité des solutions et permet aussi l'ajout d'informations additionnelles (comme le flot optique ou la ZNCC). Cette tendance s'observe aussi sur les F-puretés maximales obtenues (2^e ligne). Dans le cas particulier de *PA*, le maximum de F-pureté ne se trouvait pas forcément dans la plage $[0.02, 0.2]$, ce qui explique que l'écart-type des p_e au maximum de F-pureté puisse être supérieur à 20%. L'écart-type des paramètres p_e correspondant aux maxima des F-puretés diminue avec l'ajout d'information additionnelles. L'estimation des similarités permet l'addition d'informations supplémentaires et un gain en stabilité face à des variations du paramètre d'*a priori* p_e.

Compte tenu des résultats présentés à la table 5.1 et de ceux détaillés à la figure 5.5, deux conclusions peuvent être tirées :

1. L'estimation des similarités permet d'améliorer nettement les résultats. Cette conclusion se tire de la comparaison des résultats *PA* et *EPA*. Ainsi, l'estimation des similarités permet de s'affranchir d'une normalisation précise des distances et évite des

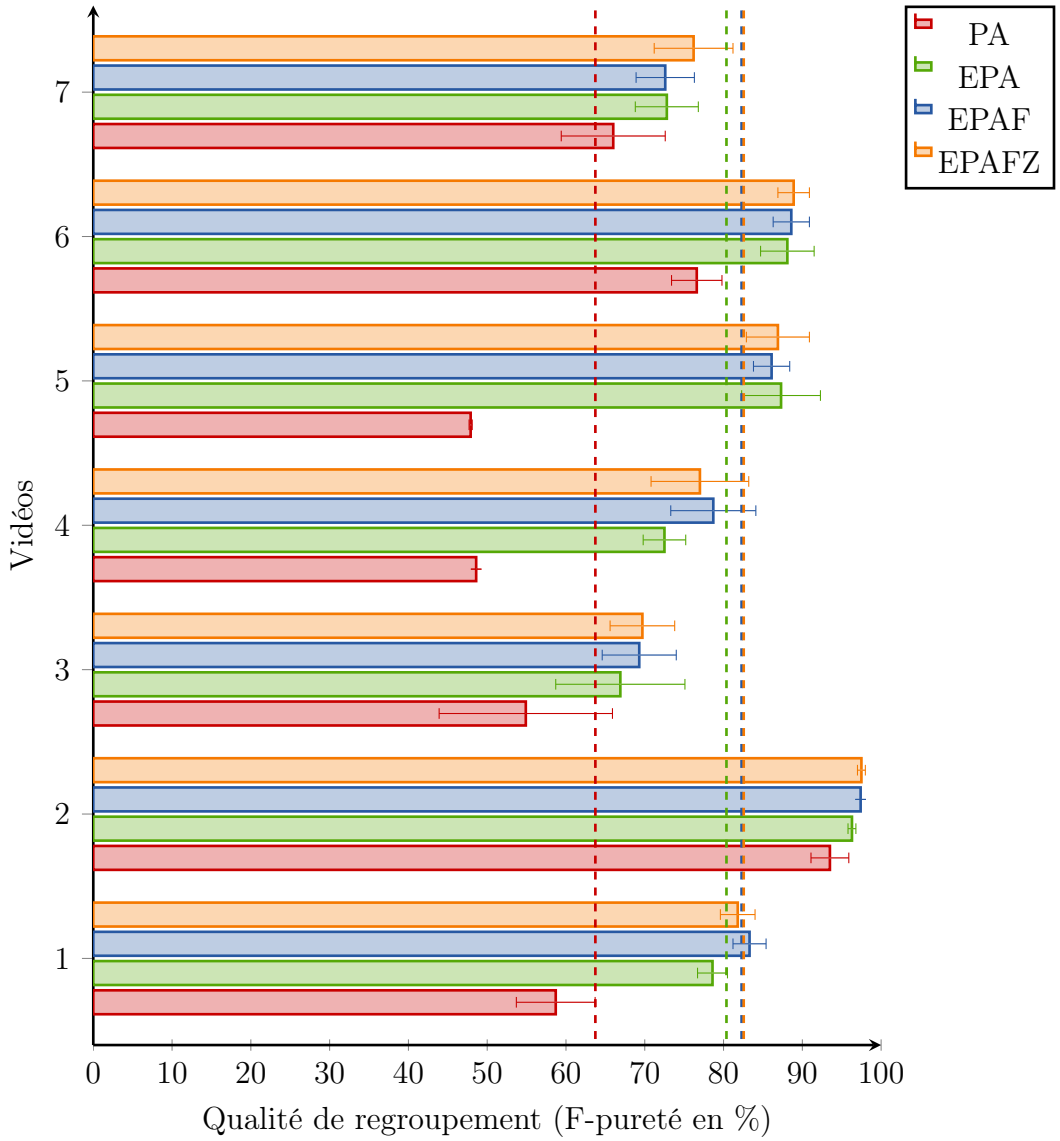

FIGURE 5.5. Résultats montrant l'apport des différentes similarité avec la procédure d'estimation issue de l'*ensemble clustering*. Les lignes pointillées représentent les moyennes des F-puretés sur les sept vidéos. *PA* : position et apparence sans estimation, *EPA* : position et apparence avec estimation, *EPAF* : position, apparence et flot optique avec estimation et *EPAFZ* : position, apparence, ZNCC et flot optique avec estimation.

définitions ambiguës ou difficilement ajustables des probabilités de transitions entre détections.

2. L'estimation des similarités permet une bonne fusion des différentes distances hétérogènes. En moyenne, on voit que le flot optique et la ZNCC ont bien été intégrés dans la similarité et les résultats se sont retrouvés améliorés. Dans certains cas, l'apport de la ZNCC n'a pas conduit à une amélioration. Cela peut s'expliquer par le fait qu'elle ne soit pas adaptée à certaines situations (particulièrement les vidéos 1 et 4 qui font intervenir des changements d'apparence relativement conséquents).

	PA	*EPA*	*EPAF*	*EPAFZ*
F-pureté	63.73	80.37	*82.29*	**82.56**
F-pureté max.	76.69	84.81	*85.66*	**86.47**
σp_e au max.	26.11	9.72	*5.44*	**4.47**

TABLE 5.1. Résumé de l'apport de l'estimation des similarités sur les sept vidéos. F-pureté : moyenne des F-puretés (en %) sur les sept vidéos, *max F* : moyenne des F-puretés (en %) maximales obtenues, σp_e *au max.* : écart-type des paramètres p_e (en %) permettant d'atteindre le maximum de F-pureté.

5.3. Qualité des albums photos construits

Après avoir présenté des résultats sur le regroupement de détections de visages, une étape est encore nécessaire pour répondre à la problématique de la construction d'albums photo. Elle consiste en la sélection des représentants de chaque groupe. Face à l'enjeu du regroupement de visages, la sélection des visages représentatifs pour l'album est une question pour laquelle on n'a pas pu accorder beaucoup de temps. Toutefois, quelques résultats sont présentés pour donner un ordre de grandeur des performances sur la base de tests précédemment décrite.

Pour sélectionner une détection par groupe établi, un score est calculé à partir de chaque détection et celle ayant le meilleur score est sélectionnée. Ce score fait intervenir trois éléments :

1. la résolution : plus la taille (en pixels) de l'image est élevée, plus elle peut donner de détails pouvant aider l'identification du visage
2. la qualité : l'album préférera sélectionner les visages les plus nets
3. la couleur : sélectionner une image comportant des couleurs proches de celles d'une peau permet de ne pas prendre comme représentant un faux positif qui aurait des couleurs trop éloignées des couleurs de peau.

Ces trois éléments sont intégrés dans un score par une simple multiplication :

$$score(z_i) = w_i \times s_{skin}(z_i) \times (1 - s_{blur}(z_i)) \tag{5.6}$$

où z_i décrit la détection i, w_i sa largeur en pixels, $s_{skin}(z_i)$ sa proportion de couleurs de peau et $s_{blur}(z_i)$ un score qualifiant sa quantité de flou.

Pour ce qui est de l'estimation de la quantité de flou, le critère présenté par F. CRETE et al. 2007 a été employé. Cette technique a été sélectionnée pour sa simplicité et la qualité de ses résultats. Elle a aussi été employée pour tester la qualité des images de visages dans le cadre de travaux (C. FICHE 2012) portant justement sur l'identification faciale dans les contextes de vidéosurveillance en environnement non-contraint.

La proportion de couleur de peau est calculée à partir d'une segmentation colorimétrique de la peau présentée par N. RAHMAN et al. 2006. Cette proportion correspond au nombre de pixels considérés comme couleur de peau sur le nombre total de pixels de l'image.

Un autre moyen employé pour diminuer le nombre de faux positifs dans l'album, consiste à simplement supprimer les groupes comportant un trop faible nombre de détections. Comme expliqué à la section 3.1.2, les faux positifs sont souvent plus isolés des autres détections. Quand cet aspect n'est pas directement intégré dans le modèle probabiliste (si P_{fp} n'est pas utilisée), des groupes comportant peu de détections (de 1 à 5 en pratique) sont constitués

par l'algorithme. Dans la grande majorité des cas, ils révèlent des faux positifs. Pour les résultats présentés, les groupes comportant moins de trois détections ont été mis de côté.

La méthode de regroupement utilisée ici est celle basée sur une modélisation par MAP avec comme probabilité de transition (P_{link}) une simple moyenne des différentes similarités (cf. s_H à l'équation 5.1). Cette similarité a été choisie pour ses performances qui sont illustrées à la section 5.2.

L'évaluation d'un album (construit par l'algorithme de regroupement et la méthode de sélection présentée) s'appuie sur la précision et le rappel définis à la section 2.4.4. La précision peut être interprétée comme le taux de personnes retrouvées (cf. C_r) et le rappel comme $1 - C_d$ où C_d est le taux de doublons de l'album.

La figure 5.6 présente un diagramme précision–rappel des albums obtenus en prenant l'algorithme basé sur la modélisation MAP. Trois cas ont été considérés.

Dans le premier cas, le paramètre d'*a priori* (p_e) a été arbitrairement fixé à 2% pour toutes les vidéos. Pour le deuxième cas, ce paramètre a été fixé par la proportion de trajectoires (ratio : nombre de personnes sur le nombre de détections) de la vérité-terrain. Le dernier cas montre les résultats obtenus en prenant l'algorithme k-CC, où le nombre de trajectoires est fixé au double du nombre de personnes de chaque vidéo.

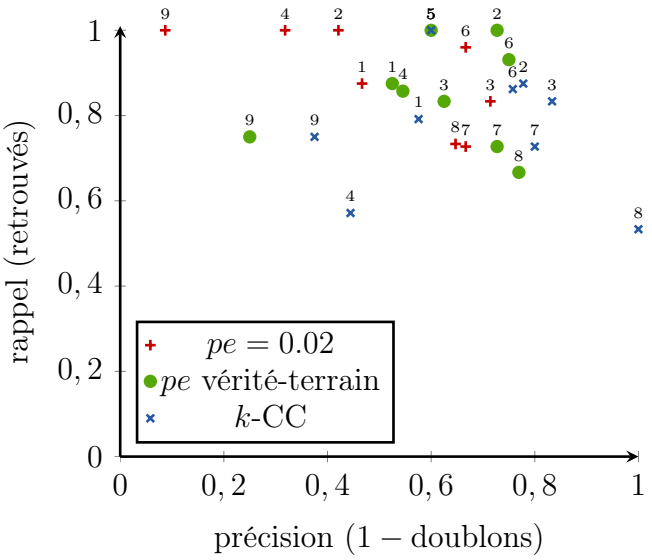

FIGURE 5.6. Qualité des albums obtenus. Les points présentent les solutions obtenus sur les différentes vidéos de test, les chiffres indiquent la vidéo en question. $p_e = 0.02$: le paramètre p_e est fixé à 2% pour toutes le vidéos, p_e *vérité-terrain* : le paramètre p_e est fixé pour chaque vidéo suivant la proportion de trajectoires issues de la vérité-terrain, k-*CC* : nombre de trajectoires fixé.

Premièrement, ces résultats montrent bien que le choix du paramètre d'*a priori* reste délicat. En surestimant le paramètre p_e (*ie* en favorisant un nombre élevé de regroupements), le rappel augmente, mais au détriment de la précision. Ces résultats illustrent aussi les deux stratégies possibles : utiliser la modélisation MAP et essayer de régler p_e au mieux ou utiliser l'algorithme k-CC avec un k surestimé. En favorisant le rappel, les résultats montrent que l'utilisation de l'*a priori* dans le MAP donne en général de meilleurs performances que le k-CC, même si p_e est surestimé.

Le deuxième point que l'on peut souligner est que les performances sont encore loin de celles attendues pour une application dans le domaine policier. Les points devraient avoir

vidéo	1	2	3	4	5	6	7	8	9
faux positifs (%)	3.7	0	0	9.8	10	0	15.5	0	19.4

TABLE 5.2. Moyenne sur les trois expérimentations des taux de faux positifs dans les albums.

une précision et un rappel supérieurs à 0.9. Comme dans les expérimentations le rappel a été privilégié (c'est ce qui est fait en pratique : les doublons sont moins graves que les omissions), dans plusieurs vidéos (2, 4, 5, 6 et 9) un rappel supérieur à 95% est atteint, mais la précision dépasse difficilement les 60%. Dans le cas où p_e est fixé à 2%, en moyenne, le pourcentage de personnes retrouvées dans l'album est de 90% et celui des doublons est de 50%.

Principalement grâce à la suppression des regroupements de petite taille et à la proportion de couleur de peau utilisée par le score de sélection, les taux de faux positifs dans l'album restent relativement faibles dans la plupart des cas (cf. table 5.2).

FIGURE 5.7. Aperçu des albums construits pour les neuf vidéos. Encadrés de rouge figurent les faux positifs et en bleu les doublons.

La figure 5.7 donne un aperçu des albums obtenus. Sur ces albums figurent aussi les doublons et les faux positifs. Les vidéos 1 et 6 font intervenir des figurants qui sortent

du champ de vue de la caméra et réapparaissent plus tard. C'est pour cela qu'ils figurent plusieurs fois dans l'album sans qu'ils ne soient comptés comme doublons.

5.4. Autres applications

5.4.1. Suivi de voitures

La méthode a aussi été appliquée à des détections de voitures. Elle se base sur un système de détection qui est en cours d'élaboration au laboratoire. La scène est constituée d'un rond-point filmé avec une caméra à focale courte. La résolution de la vidéo est faible (352×288) et la compression donne une qualité finale très médiocre. C'est pourquoi, même si les occultations interviennent moins souvent qu'avec les visages, le suivi des voitures n'est pas évident. La séquence dure 3 minutes et 60 secondes pour un total d'environ 5000 frames. Sur l'ensemble de cette vidéo, 14500 voitures ont été détectées.

Les tests menés ont permis d'obtenir les premiers résultats, qui comportent encore des défauts, mais donnent une première idée du potentiel de la méthode développée au cours de la thèse.

La similarité employée pour évaluer le lien entre les détections prend en compte le temps, la position et la colorimétrie. Ici un simple produit de P_{time}^{geo}, s_{pos} et s_{rgb} a été pris pour la similarité. Ensuite, l'algorithme d'estimation est employé sur cette similarité pour que l'on puisse avoir une probabilité P_{link} réaliste. Pour cette estimation, le nombre maximal de groupes a été fixé à $C_{max} = 200$. Comme cette étape peut prendre du temps, l'estimation a été effectuée sur des sous-séquences de 500 frames se chevauchant de 250 frames.

Finalement l'algorithme basé sur le MAP a été employé avec l'*a priori* binomial, mais sans prendre en compte les faux positifs dans l'*a priori*. Le paramètre de proportion de trajectoires a été fixé à $p_e = 0.005$. On peut noter que ce paramètre ($\frac{\text{nb. trajectoires}}{\text{nb. détections}}$) est proche de celui observé en réalité, car les trajectoires présentent en moyenne 200 détections.

La figure 5.8 donne un aperçu des trajectoires obtenues avec le procédé décrit.

Pour donner des ordres de grandeur des temps de calcul, l'étape d'extraction des caractéristiques et de calcul des similarités a pris 15 secondes. L'étape la plus longue est celle de l'estimation des similarités ; elle a pris 20 minutes de temps de calcul. Puis, l'algorithme de regroupement des détections a pris environ 30 secondes. Dans le cadre de ce test, peu d'efforts ont été faits pour construire une similarité adaptée à la situation, c'est l'algorithme d'estimation qui a permis de bien estimer les probabilités des liens entre les détections.

5.4.2. Suivi de bombes volcaniques

Un autre test de la méthode de regroupement de détections pour le suivi a été mené sur des vidéos d'éruptions volcaniques. Elles ont été tournées de nuit pour essayer d'obtenir des statistiques sur les trajectoires des bombes volcaniques. Sur chaque vidéo, environ 20 000 bombes ont été détectées. La détection est faite par recherche de maxima locaux sur le laplacien des images. Le laplacien a été estimé par différence de gaussiennes.

La particularité de ce type de séquences est qu'elles font intervenir un grand nombre de détections par image et plusieurs centaines de trajectoires au final. Comme l'apparence des différentes bombes volcaniques n'est pas discriminante pour le regroupement, les similarités employées n'ont fait intervenir que la position des bombes dans l'image et leur temps dans

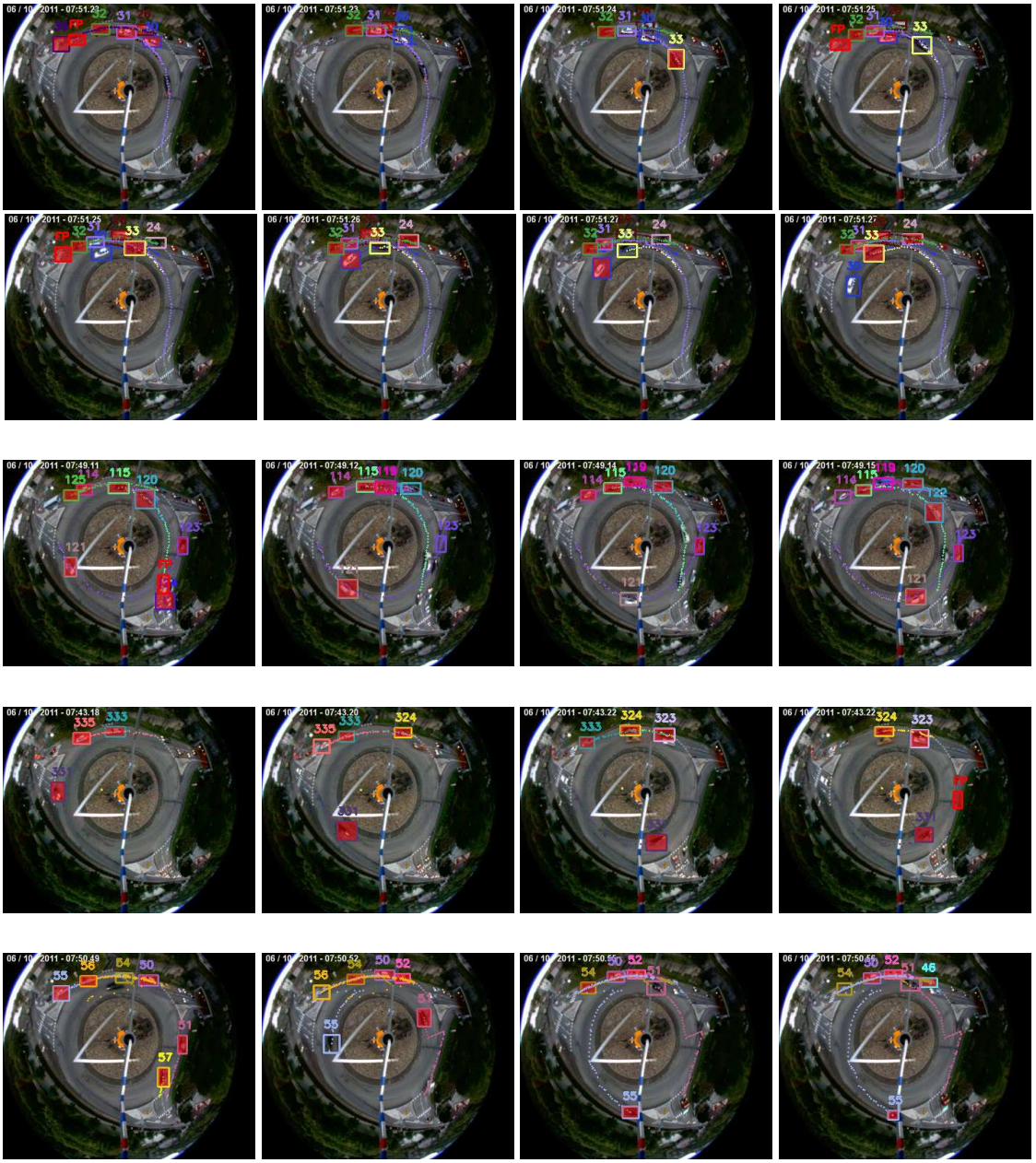

FIGURE 5.8. Aperçu des résultats obtenus en suivi de voitures par regroupement de détections. Les lignes d'images correspondent à une séquence de résultats. Chaque rectangle teinté de rouge correspond à une détection ayant servie pour le regroupement. Les rectangles qui ne sont pas teintés en rouge correspondent à une interpolation linéaire entre deux détections. Chaque couleur (avec son identifiant) correspond à chaque groupe construit par l'algorithme. Les détections rouges notées *FP* sont celles considérées comme un faux positif du détecteur. Les pointillés représentent l'historique de la trajectoire.

la vidéo. Un filtrage préalable des vitesses trop élevées (en pixels par frame) a aussi été effectué.

L'algorithme k-CC a ici été employé pour reconstituer les trajectoires, avec un nombre de groupes surestimé (4000 au lieu de plusieurs centaines). Les trajectoires faisant intervenir moins de 10 détections ont été enlevées. Cette stratégie est utilisée pour ne garder que les trajectoires les plus longues, étant donné qu'il est nécessaire d'avoir des trajectoires

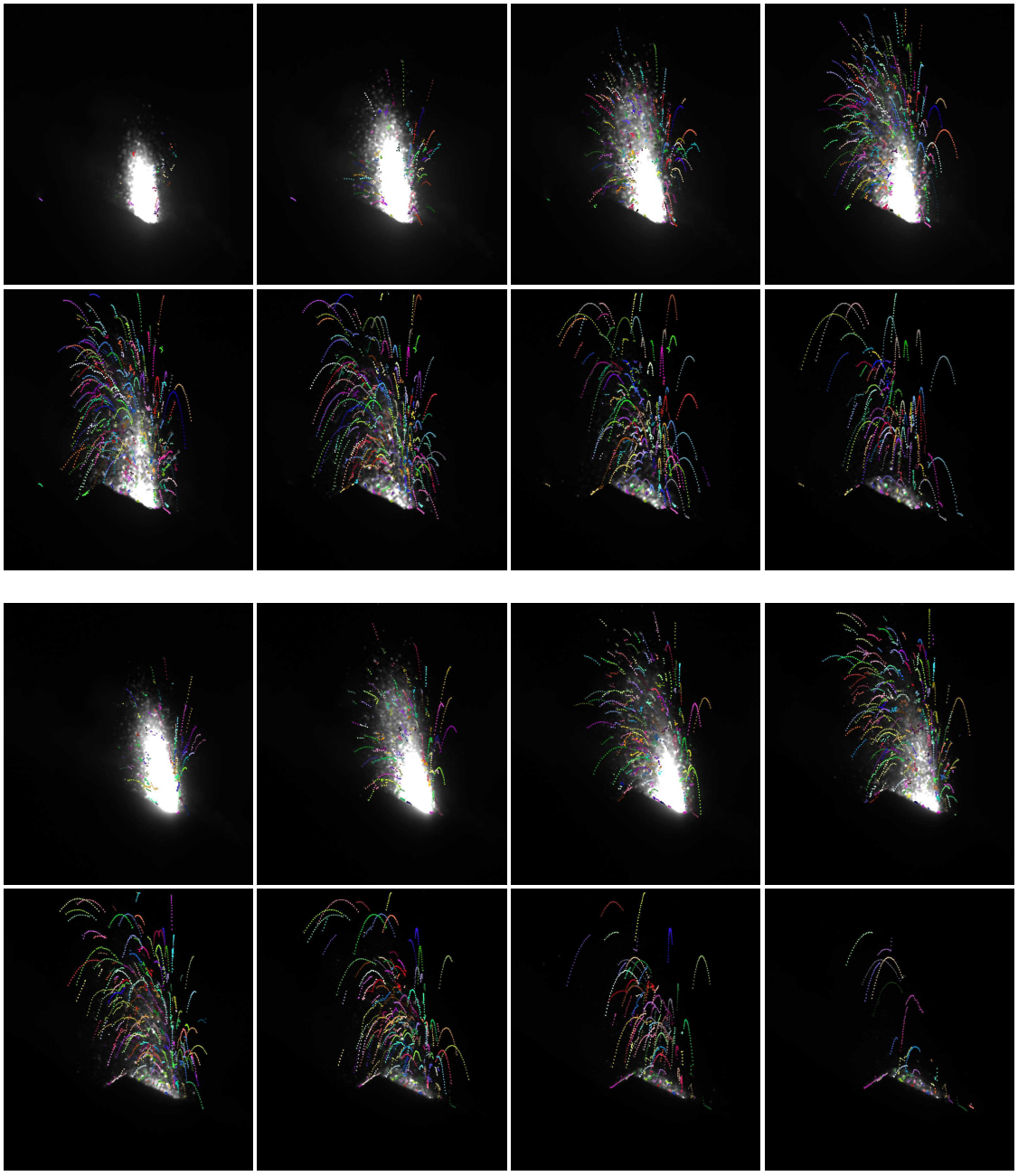

FIGURE 5.9. Aperçu du suivi de bombes volcaniques lors d'une éruption. Les images présentent deux séquences où des trajectoires ont été reconstruites. Pour représenter les séquences, une image à été prise toutes les dix frames. L'historique des différentes trajectoires construites est indiqué en pointillés colorés.

suffisamment grandes pour que les vulcanologues puissent en extraire des statistiques. Le calcul des matrices de similarités a pris environ 20 secondes et l'exécution du k-CC 2 minutes et 30 secondes.

La figure 5.9 donne un aperçu des trajectoires reconstruites. Aucune vérité-terrain n'est encore disponible, il est donc difficile de mesurer la qualité des résultats. Toutefois, on peut voir que des trajectoires paraboliques ont bien été extraites.

5.5. Répartition des temps de calcul

Les expérimentations menées ont nécessité un développement informatique qui a occupé une grande partie du temps de la thèse, mais ce rapport n'a pas pour vocation de rentrer dans des détails techniques quant à l'implémentation des algorithmes présentés. Les premiers tests ont été effectués avec le logiciel *Matlab* et, par soucis de performances, la quasi-totalité de la méthode et de ses nombreuses variantes ont été ré-implémentées en *C++* avec quelques bibliothèques tierces (*OpenCV*[1], *Eigen*[2] et *LEMON*[3]). Quelques chiffres donnant un ordre de grandeur des temps d'exécution sont présentés, ils ont pu être atteint grâce aux efforts faits au niveau de l'implémentation.

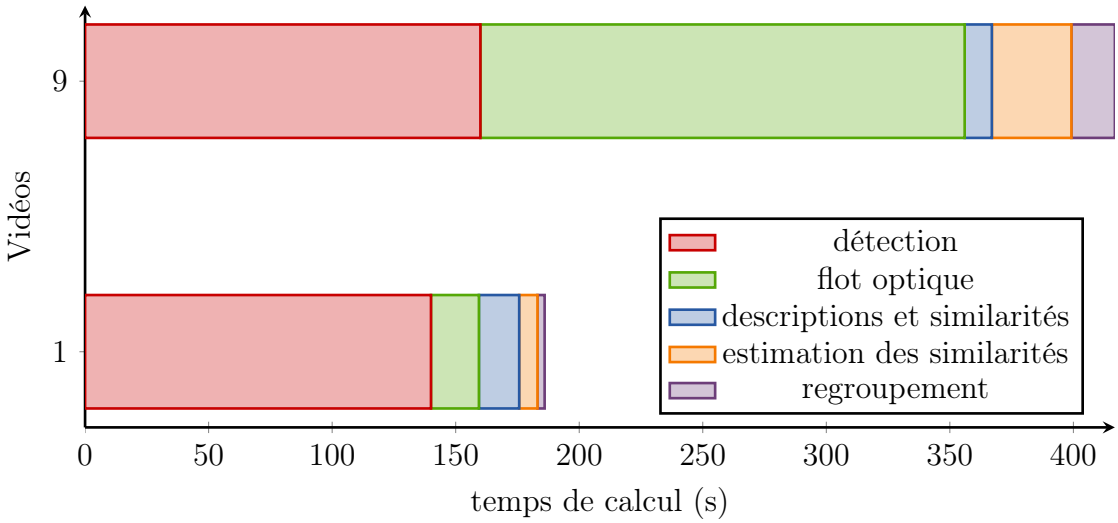

FIGURE 5.10. Comparatif du temps de calcul des différentes étapes de la méthode. Les vidéos 1 et 9 ont été utilisées.

La figure 5.10 donne un aperçu de la répartition des temps de calcul sur deux vidéos de la base de test. Lors de l'exécution de l'algorithme, la plus grande partie du temps a été utilisée dans la phase de détection et pour le calcul du flot optique. Le flot optique prend particulièrement beaucoup de temps quand les détections sont de grande taille (vidéo 9). Les autres étapes (calcul des descriptions et similarités, estimation et regroupement) sont plus rapides. Le calcul du flot optique n'étant pas toujours une étape nécessaire, il peut être omis. Il en résulte que l'étape de détection reste la plus consommatrice en temps de calcul (75% du temps total).

Toutes les expérimentations présentées dans ce rapport ont été effectuées sur un PC de bureau standard (processeur 2.4 GHz à 4 cœurs et 3 Go de RAM) et les temps de calcul indiqués ne font pas intervenir de parallélisation (exceptée l'étape de détection qui utilise plusieurs cœurs).

Ce qui ressort de ces tests est que la détection des visages prend un temps considérable (75% du temps de calcul pour la vidéo 1 et 40% pour la vidéo 2) et le calcul du flot optique reste aussi important pour la vidéo 9 (47% du temps de calcul) mais il n'est pas nécessaire dans tous les cas. Finalement, comme la phase de détection est incontournable, la réduction

1. `http://opencv.org`
2. `http://eigen.tuxfamily.org`
3. `http://lemon.cs.elte.hu`

du temps de traitement passe inévitablement par une amélioration du temps de calcul du détecteur.

5.6. Conclusion

Ce chapitre a présenté différentes expérimentations qui ont permis d'évaluer la méthode de regroupement et les stratégies mises en œuvre pour calculer les similarités. L'intérêt des méthodes développées a aussi été confirmé par des résultats de construction d'album photo et de suivi dans d'autres cadres applicatifs.

Tout d'abord les tests comparatifs ont permis de voir que l'algorithme de regroupement mis en œuvre (k-CC) est adapté au regroupement de détections issues d'une vidéo. Le partitionnement hiérarchique (HAC) reste toutefois compétitif. Avec certaines similarités, le partitionnement spectral peut aussi permettre d'améliorer légèrement les regroupements.

Les évaluations menées pour comparer les différentes mesures de similarités présentées ont révélé que l'algorithme d'estimation permet d'intégrer des similarités très hétérogènes. Toutefois, les distances de Hellinger entre lois normales se sont avérées plus pertinentes dans notre cas d'utilisation. La preuve en est qu'une simple moyenne sur les différentes similarités (position, apparence et mouvement) donne de bons résultats.

Ensuite, des résultats relatifs aux albums construits grâce à l'algorithme de regroupement ont été présentés. Les expérimentations ont montré qu'en moyenne une personne sur dix n'est pas représentée par l'album, et la moitié des photos de l'album est constituée de doublons. Ces résultats restent encore très variables suivant la qualité de la séquence et la situation filmée.

Finalement, l'intérêt en terme de suivi multi-cibles de la méthode a aussi été montré par quelques expérimentations. Une situation de suivi de véhicules a permis de mettre en avant l'algorithme d'estimation des similarités qui rend plus facile le réglage du terme d'*a priori*. Le cas du suivi de bombes volcaniques a montré que l'algorithme k-CC permet d'extraire facilement les trajectoires les plus cohérentes, ce qui est justement recherché par les vulcanologues.

Pour ce qui est de la répartition des temps de calcul, les tests ont indiqué que l'étape la plus limitante reste la détection des visages et que les autres étapes sont relativement rapides pour permettre une exploitation en situation réelle de la méthode développée.

Conclusions et perspectives

Conclusions

Cette thèse s'inscrit dans le contexte de la vidéosurveillance et plus précisément à celui de la fouille de vidéos archivées. Elle s'intéresse à la construction automatique d'albums photo pour faciliter la recherche de suspects dans le cadre d'enquêtes policières. La faible résolution des visages détectés et la qualité des images ne permettent pas une reconnaissance faciale directe. Afin de regrouper par identité les visages détectés sur une vidéo, les efforts se sont portés sur les problématiques de suivi multi-cibles. Le cadre applicatif de la fouille d'archives vidéo conduit vers les approches dites globales. Ainsi, les travaux développés au cours de cette thèse contribuent aux recherches correspondant au domaine du suivi multi-cibles global basé détections, qui reste un domaine encore très peu abordé et assez hétéroclite.

Plus précisément, voici les cinq principales contributions apportées :

1. Adaptation d'un algorithme de résolution pour trouver une solution optimale rapidement. En reformulant le problème et en adaptant un algorithme de résolution existant, la complexité calculatoire a été réduite et le terme d'*a priori* est mieux intégré dans le modèle.
2. Mise en œuvre d'une stratégie de traitement séquentiel s'appuyant sur l'algorithme global de résolution. Cela permet d'aborder de plus longues séquences et d'envisager le traitement d'un flux vidéo.
3. Création d'une méthode d'estimation de la mesure de similarité entre les détections. Cela permet de fusionner automatiquement les différentes informations liées aux détections (temps, position, apparence et mouvement local). On peut souligner la création d'un algorithme original (noté k-CC par la suite) de partitionnement minimisant les dissimilarités aux transitions intervenant au sein des trajectoires.
4. Comparaison avec des approches génériques de partitionnement de données. Cette thèse a pu montrer que les solutions données par le domaine du partitionnement de données permettent de généraliser facilement le suivi multi-cibles à différents cadres applicatifs.
5. Expérimentations avec des situations réalistes. Il existe peu de résultats d'analyse de vidéos présentant la réalité de la vidéosurveillance en environnement non-contrôlé. Cette thèse présente un ensemble de vidéos de test difficiles : basse qualité de l'image, situations très différentes et scènes denses.

Une des originalités de l'approche présentée, est de voir le problème du suivi multi-cibles comme un partitionnement de détections. Cela permet d'obtenir un algorithme général qui ne s'appuie que sur une mesure de similarités entre les détections. Plusieurs de ces mesures de similarités ont été présentées et comparées dans le cadre du suivi de visages en contexte de vidéosurveillance non-contrôlée. Les travaux ont aussi abouti à une méthode d'estimation des similarités entre les détections qui permet de joindre automatiquement différents indices

hétérogènes liés aux détections (temps, position, apparence et mouvement) et de faire le lien avec la modélisation probabiliste de maximum *a posteriori*.

Les techniques développées au cours de la thèse ont été testées sur des séquences complexes, représentant la réalité des scènes de vidéosurveillance. Bien que les résultats ne satisfassent pas encore complètement les besoins de la police judiciaire, la modélisation et l'algorithme présentés par cette thèse donnent un point de départ et un cadre suffisamment général pour permettre une extension à d'autres applications. Cette capacité de généralisation vient principalement du fait que l'approche adoptée fait intervenir uniquement une mesure des similarités entre détections. Cela s'est également illustré par des tests menés sur des situations très différentes (suivi de voitures vues de haut et suivi de bombes volcaniques lors d'éruptions nocturnes).

Perspectives

Les performances des méthodes présentées pourraient être améliorées en considérant les points suivants :

Évaluer sur un plus grand jeu de vidéos. Cette thèse présente, à différents niveaux, de nombreuses options possibles :

- Algorithme : le MAP résolu par une recherche de flot de coût minimal donne de bons résultats et le partitionnement hiérarchique reste compétitif.
- Similarités inter-détections : différentes mesures de similarités et manières de les combiner ont été présentées, avec possibilité de pré-conditionnement spectral ou d'estimation par *ensemble clustering*.
- Traitement séquentiel : par sous-séquences disjointes ou hiérarchique.

Pour compléter la pertinence de ces différentes options, il faudrait pouvoir effectuer des tests sur un plus grand nombre de séquences de vidéosurveillance ainsi que sur des vidéos plus longues. La principale difficulté réside probablement dans la mise en place d'une campagne d'acquisition grandeur nature qui reste coûteuse. De plus, pour d'autres types de suivis (faisant intervenir des situations différentes de celles de la vidéosurveillance) les conclusions sur la pertinence de ces options peuvent différer de celles présentées.

Intégration de données supplémentaires. En fonction des applications recherchées, le système mis en œuvre (notamment l'estimation des similarités) rend possible l'ajout d'indices supplémentaires. Plusieurs expérimentations dans des cadres différents de celui du suivi de visages pourraient être menées pour compléter l'approche proposée. La reconnaissance faciale, les sous-titres ou encore le découpage par scènes pourraient, par exemple, être ajoutés si l'on veut traiter des vidéos de films ou séries TV par exemple. Pour une application en surveillance de trafic routier, il peut être intéressant d'utiliser des informations sur les sens de circulation. Avec l'estimation des similarités proposée, tous ces indices peuvent être facilement intégrables.

Amélioration de l'approche hiérarchique. La direction qui semble la plus pertinente pour améliorer la méthode de suivi multi-cibles présentée, consiste à traiter le problème de manière progressif en effectuant successivement une construction des trajectoires et une ré-estimation des similarités. Ce point a commencé à être abordé par le traitement en

sous-séquences, mais les similarités inter-trajectoires peuvent être améliorées. Ces similarités pourraient être perfectionnées en considérant la dynamique des cibles ou encore en utilisant un système capable de bien comparer deux ensembles d'apparences (cf. A. FITZGIBBON et A. ZISSERMAN 2003). Une telle stratégie de traitement progressif serait une manière de répondre aux problèmes liés aux dépendances du 1er ordre utilisé dans les représentations des trajectoires par des chaines de Markov.

Pistes pour améliorer le temps de traitement des vidéos. Une utilisation de la méthode de suivi proposée pour des vidéos de longues durées (plusieurs heures) demande encore des efforts pour réduire le temps de calcul. L'implémentation développée au cours de la thèse montre que, si des descriptions simples sont utilisées, l'étape limitante reste la détection. L'accélération de cette étape permettrait de rendre la méthode beaucoup plus rapide. Plusieurs stratégies pourraient être envisagées. Si la vidéo provient d'une caméra fixe, l'extraction d'arrière-plan peut être utilisée pour guider le détecteur pour ne pas scanner toute l'image, ce qui permet d'améliorer les temps de calcul. Une autre direction (cf. travaux de H. WANG et al. 1999) serait d'utiliser directement les informations issues de la compression vidéo. L'objectif de la compression étant de réduire la taille des vidéos, une analyse est faite pour décrire au mieux les apparences et estimer les mouvements locaux. Il est possible d'utiliser directement ces descripteurs de la compression (par exemple le DCT qui est souvent utilisé) et les trames clés pour détecter les visages, mais aussi l'information de mouvement déjà extraite par l'encodage pour suivre localement les détections.

Annexes

Annexe A.

A.1. Modélisation par MDL

Dans une volonté de décrire plus justement l'*a priori* et d'équilibrer plus efficacement l'*a priori* et la vraisemblance, le principe de *longueur de description minimale* a attiré notre attention.

L'objectif de notre approche est de trouver le "meilleur" modèle T (*ie* ensemble de trajectoires, ou partition de détections) qui expliquerait au mieux l'ensemble des observations issues des détections de visages. Exprimé en ces termes, le lien entre notre objectif et les problématiques dites de sélection de modèle est plus explicite. Cette classe de problèmes occupe une place importante dans l'inférence statistique, où l'on recherche (de manière implicite ou explicite) un compromis entre la complexité du modèle et la qualité de l'ajustement aux données observées.

Dans le domaine de la théorie pour la sélection de modèle, il existe un principe relativement récent donnant un cadre assez générique : la longueur de description minimale (*Minimum Description Length* noté MDL).

Cette section présentera premièrement le principe de longueur minimale de description (MDL : Minimum Description Length) et décrira une application possible à notre problématique. Ce principe a déjà été employé dans le domaine du suivi multi-objets (B. Benfold et I. Reid 2011 et B. Leibe et al. 2007).

Le principe provient principalement des travaux de J. Rissanen (1978) qui s'appuie notamment sur la théorie de la complexité de Kolmogoroff. Pour la compréhension de ce principe, nous avons principalement utilisé un tutoriel de P. Grunwald 2004 et une publication M. Hansen et B. Yu 2001.

Le MDL est basé sur l'idée suivante : une certaine régularité dans les données peut être utilisée pour les compresser. Compresser veut dire ici décrire les données avec moins de symboles qu'avec le nombre de symboles nécessaires à la définition littérale de celles-ci. Ainsi, un modèle qui compressera au mieux les données sera un modèle qui les représentera au mieux.

Les deux principaux aspects du MDL ayant attiré notre attention sont les suivants :

1. Le MDL tient compte des problèmes de sur-ajustement. Il gère conjointement la structure du modèle (par exemple le nombre de paramètres) et l'ajustement de ce modèle aux données, et cela de manière inhérente.
2. Le MDL conserve des liens étroits avec l'inférence Bayésienne.

Le premier intérêt est particulièrement pertinent dans le cadre du suivi multi-cibles traité globalement. Le suivi recherche d'un côté des trajectoires proches des observations et de l'autre à limiter le nombre de trajectoires, cela pour ne pas avoir un trop grand nombre de trajectoires sur-ajustées qui ne représente plus la réalité. On peut dire que le suivi recherche

aussi un représentation minimaliste et sans injecter d'*a priori* dans le modèle afin de rester général.

Dans la plupart des méthodes de suivi multi-cibles globaux ou de partitionnement, les modélisations probabilistes utilisées font intervenir un *a priori* sur le nombre de trajectoires (ou groupes) qui est fixé empiriquement ou à l'aide d'apprentissage sur des bases annotées. Le MDL semble intéressant pour ces types d'applications parce qu'il incite à ne pas ajouter trop d'*a priori* dans le modèle par sa gestion intrinsèque de la complexité du modèle.

Comme de nombreux travaux se placent dans un cadre bayésien, le deuxième point a aussi tout son intérêt : il permet de garder un lien avec les modélisations plus classiques qui sont plus couramment utilisées.

Pour aller plus en détails, on peut dire que le MDL se présente sous deux versions : l'une simpliste (*crude MDL*) et l'autre plus raffinée (*refined MDL*) en se basant notamment sur la notion de code universel. La version simpliste est ici abordée.

Plus formellement l'objectif est de trouver T minimisant $L(T) + L(Z|T)$, où T est une hypothèse de l'ensemble $\mathcal{T}$ de tous les modèles et Z l'ensemble des données observées.

Le MDL sera maintenant illustré par notre problématique de regroupement de détections. En supposant avoir un ensemble de D détections sur une séquence vidéo, un partitionnement T optimal est recherché. Les observations liées à ces D détections $Z \doteq (z_1, ..., z_D)$ où z_i caractérise une détection (par exemple position, taille, vitesse, apparence...) représentent les données du problème.

Pour formuler le MDL il nous faut décrire T ce qui permettra de définir $L(T)$, et ensuite décrire la manière dont le lien avec les observations $L(Z|T)$ est modélisé.

Une idée simple est de voir les éléments de T comme des trajectoires modélisées par une chaine de Markov dont on connait les transitions par les observations. Ainsi, une hypothèse T peut être représentée par le nombre de trajectoires et un identifiant de trajectoire par détection : $T = (K, c_1, ...c_D)$ où $K \in [1, D]$ et $c_i \in [1, K]$ représente l'index de la trajectoire contenant la i-ème détection. Si les z_i sont rangées par date d'apparition dans la séquence et que l'on connait les probabilités de transition θ_{ij}, cela suffit à définir les K chaines de Markov.

La longueur de code de T peut alors se définir de la façon suivante :

$$L(T) = \log(\frac{K^D}{K!}) + \log(K) = (D+1)\log(K) - \log(K!) \tag{A.1}$$

K et les D index $c_i \in [1, K]$ sont codés, et cela sans prendre en compte les permutations des index (d'où $\frac{1}{K!}$).

Pour ce qui est du terme de vraisemblance, on peut utiliser simplement l'opposé de la log-vraisemblance :

$$L(Z|T) = -\log(P(Z|T)) \tag{A.2}$$

en se basant sur la définition de la vraisemblance donnée à la section précédente. Ainsi, on peut chercher à minimiser la longueur de code définie de la manière suivante :

$$L(T) + L(Z|T) = (D+1)\log(K) - \log(K!) - \log(P(Z|T)) \tag{A.3}$$

Le problème avec cette modélisation est la manière dont est représenté le terme d'*a priori* ($L(T)$). Comme l'illustre la figure A.1, le coût lié à l'*a priori* est décroissant à partir d'un

certain nombre de trajectoires, et ainsi, la minimisation de $L(T) + L(Z|T)$ peut conduire à une solution où il y aurait autant de trajectoires que de détections car $L(Z|T)$ serait aussi minimale dans ce cas-ci. Les tests menés s'appuyant sur cette approche ne permettaient pas de donner de meilleurs résultats que ceux obtenus par la méthode présentée au chapitre 3.

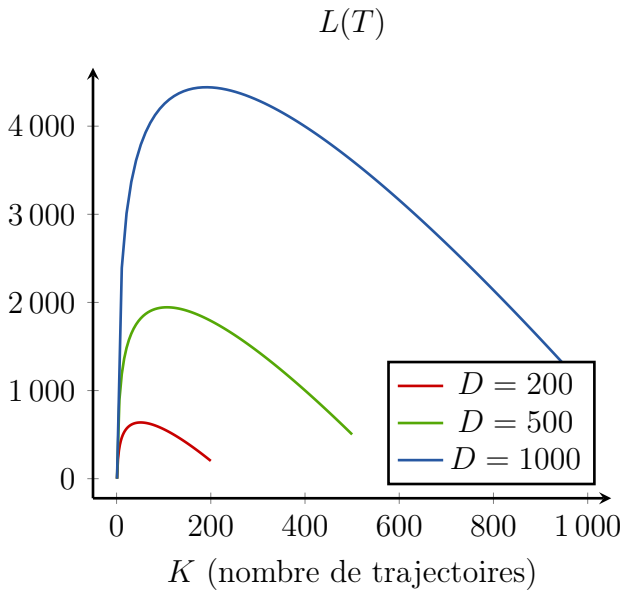

FIGURE A.1. Aperçu du coût représentant l'*a priori* du modèle en utilisant le principe du MDL.

Afin d'utiliser le potentiel du MDL il faudrait pousser plus loin la modélisation et ne pas se limiter à la forme simpliste et notamment se pencher sur l'ensemble des représentations possibles de T pour s'orienter vers la forme raffinée.

Dans le cadre des travaux présentés dans ce rapport, la modélisation n'a pas été poussée plus loin. Ce principe peut toutefois apporter un angle de vue intéressant sur le problème du choix du nombre de trajectoires et sur la définition de la vraisemblance des trajectoires.

A.2. Modélisation par un réseau bayésien

Étant donné la volonté de définir des probabilités conditionnelles pour estimer les probabilités de transition et ayant déjà exploité une représentation graphique du problème, les réseaux bayésiens pourraient être une modélisation plus adaptée à notre problématique.

Le principe fondamental des modèles graphiques (dont les réseaux bayésiens font partie) est de représenter par un graphe les dépendances conditionnelles entre les différentes variables aléatoires. Dans notre cas, il serait possible de représenter la vraisemblance $P(Z|T)$ comme la probabilité jointe d'un réseau bayésien pour représenter la dépendance entre les détections et utiliser les similarités entre les détections. Cette probabilité jointe sera notée $P_{B_z}(T)$.

Le premier point important, dans le cadre des réseaux bayésiens, est de construire le graphe de dépendance des variables aléatoires. Avec l'objectif du regroupement de détections en trajectoires, dans l'idéal, il faudrait envisager l'ensemble des dépendances temporellement admissibles entre détections. Plus précisément, il faudrait prendre en compte toutes les dépendances de chaque détection avec celles qui sont apparues plus tard dans la séquence

vidéo. Cela reviendrait à la construction d'un graphe de dépendances sensiblement identique à celui construit pour la résolution par flots de coût minimal (cf. 3.2.1).

La probabilité jointe peu s'exprimer de la façon suivante :

$$P_{B_z}(T) = \prod_{i=1}^{D} P(C_i|\text{parents de } C_i) \tag{A.4}$$

où C_i correspond à une variable aléatoire discrète permettant de déterminer à quelle trajectoire le détection i appartient.

Dans notre cadre applicatif, comme les liens entre les variables sont supposés connus (par le graphe causal), l'intérêt serait d'utiliser des méthodes d'inférence pour estimer les probabilités conditionnelles. Toutefois, l'inférence présente vite des complexité calculatoires non négligeables. Cela vient du fait que, d'une part le graphe est fortement connecté et, d'autre part, le nombre d'états de chaque variable aléatoire peut être assez élevé (il correspond au nombre maximal de trajectoires).

Le problème rencontré avec les modèles graphiques (réseau bayésien ou champs de Markov) est qu'une trajectoire n'est pas représentée comme une simple chaine de Markov. Avec une modélisation par modèle graphique, on pourrait alors retrouver les mêmes problèmes qu'avec les algorithmes de partitionnement qui ne font pas intervenir une représentation chainée des groupes (un élément en début de cluster peut être très différent de l'élément final alors qu'ils sont dans le même cluster).

Ces deux aspects ont découragé l'usage des modèles graphiques dans le cadre de cette thèse, même si l'étude du lien entre la modélisation présentée et un modèle graphique pourrait s'avérer pertinente. Une étude plus approfondie conduirait peut-être à inférer plus efficacement les probabilités de transitions.

Annexe B.

B.1. Introduction du coefficient binomial dans le problème de flot de coût minimal

Le but de cette annexe est de décrire comment il est possible de modifier le graphe pour introduire le terme $b_k \doteq -\log\binom{D}{k}$ dans la fonction de coût globale. Le problème de ce terme est qu'il n'est pas linéaire en fonction de K et ne peut donc pas être intégré dans le coût E_{start}. Il est toutefois possible de le définir comme une somme de K termes d'une suite croissante, ce qui nous permet de le représenter par un coût d'un flot.

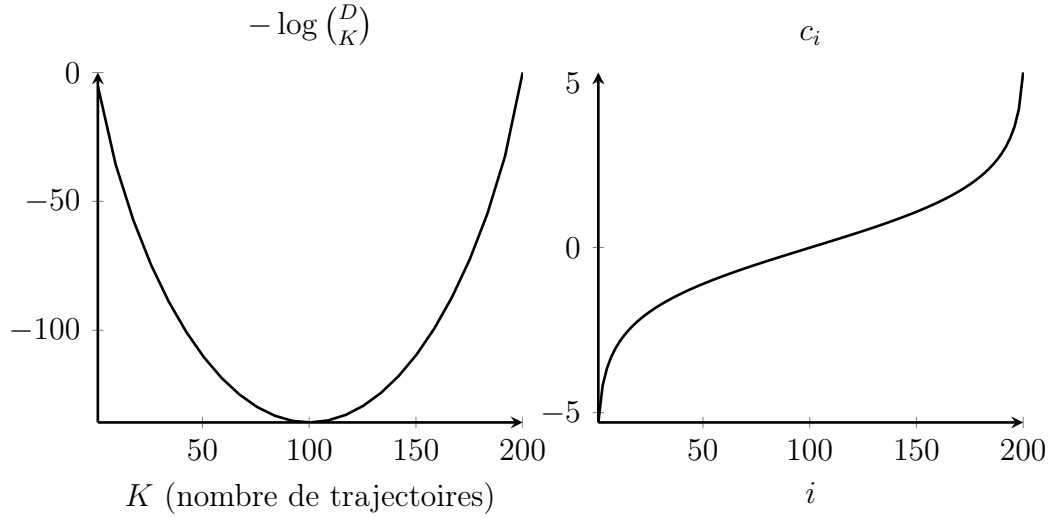

FIGURE B.1. *Gauche :* coût du coefficient binomial $(-\log\binom{D}{K})$ à ajouter pour passer de la version simplifiée à la version binomiale de l'*a priori* P_{start}. *Droite :* suite des coûts à additionner pour représenter $-\log\binom{D}{K}$. Comme la suite est croissante, le coût du coefficient binomial s'intègre dans un flot de coût minimal.

En effectuant quelques calculs avec les coefficients binomiaux, on voit que :

$$\begin{aligned} b_k &= \sum_{i=0}^{k-1} b_{i+1} - b_i = \sum_{i=0}^{k-1} -\log\frac{\binom{D}{i+1}}{\binom{D}{i}} \\ &= \sum_{i=0}^{k-1} -\log\frac{D-i}{i+1} = \sum_{i=1}^{k} -\log\frac{D-i+1}{i} = \sum_{i=1}^{k} c_i \end{aligned} \tag{B.1}$$

avec $c_i = -\log\frac{n-i+1}{i}$ et cela pour tout $k \in \{1, ..., D-1\}$ (notons que $b_0 = b_D = 0$). Comme les c_i définissent une suite croissante, il est possible d'écrire :

$$b_k = \min_{f,\ \text{s.t. } |f|=k} \sum_{i=1}^{D-1} f_i c_i \tag{B.2}$$

où $f_i \in \{0, 1\}$ correspond au flot sur un graphe particulier décrit à la figure B.2.

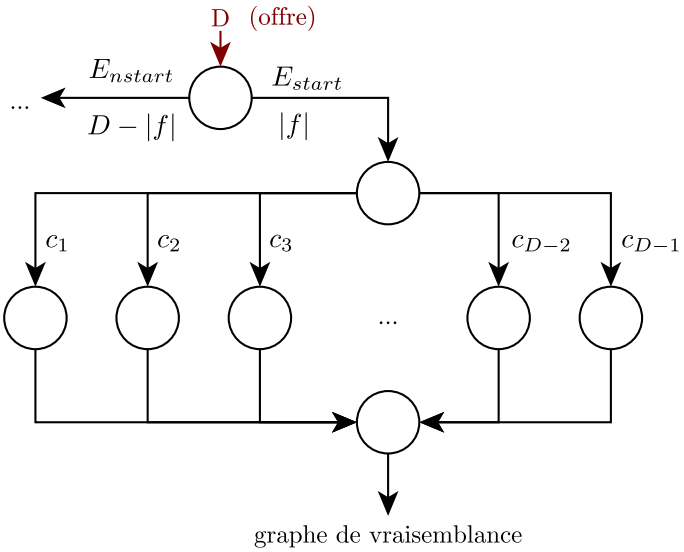

FIGURE B.2. Graphe utilisé pour représenter $-\log \binom{D}{k}$ dans le problème de flot de coût minimal. Ce graphe est placé dans le contexte de la recherche du MAP avec P_{start} dans sa version non simplifiée (cf. section 3.1.1).

Ainsi, ce graphe peut être inséré dans un graphe global pour représenter $-\log \binom{D}{k}$ et donc permettre de représenter pleinement l'*a priori* binomial P_{start} dans le coût du flot.

La même approche peut s'utiliser pour représenter $P_{struct} \propto \binom{D}{k}\binom{A}{D-k}$. Il suffit d'utiliser le graphe présenté à la figure B.2 pour insérer $-\log \binom{D}{k}$ dans le coût. Le terme $-\log \binom{A}{D-k}$ peut être représenté de la même façon en insérant le graphe au niveau de l'arc où circule le flot $D - |f|$.

B.2. Introduction du nombre de Stirling de seconde espèce dans le problème de flot de coût minimal

Comme pour le cas du coefficient binomial, l'objectif est d'intégrer, dans un coût de flot, le logarithme du nombre de Stirling de seconde espèce. En introduisant : $s_k \doteq \left\{ {D \atop k} \right\}$, le but est de le définir comme une somme croissante de 1 à k pour pouvoir le représenter dans le coût final du flot (cf. annexe B.1). Pour cela, il suffit de prendre simplement :

$$s_k = \sum_{i=1}^{k} s_i - s_{i-1} \tag{B.3}$$

et de définir les coûts du graphe par : $c_i^{str} = -\log \left\{ {D \atop i} \right\} + \log \left\{ {D \atop i-1} \right\}$. Contrairement aux coefficients binomiaux, ce terme ne se simplifie pas facilement et cette formulation a donc été

utilisée directement pour définir le coût du graphe. Cette suite semble bien croissante, elle s'intègre dans le graphe de la même façon que le coefficient binomial a été intégré (cf. figure B.2). Pour calculer la suite s_k des nombres de Stirling, la relation de récurrence suivante a été utilisée :

$$\left\{ {D \atop k} \right\} = k \left\{ {D-1 \atop k} \right\} + \left\{ {D-1 \atop k-1} \right\} \tag{B.4}$$

Ce calcul peut poser problème pour un D grand. Avec l'implémentation utilisée, les calculs n'ont pu s'effectuer qu'avec $D \leq 2000$.

B.3. Distances de Hellinger entre lois normales multivariées

Soit deux lois normales multivariées de dimension $d : \mathcal{N}(\mu_1, \Sigma_1)$ et $\mathcal{N}(\mu_2, \Sigma_2)$. Le coefficient de Bhattacharyya C_B entre ces deux distributions vaut :

$$C_B(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)) = \int_{\boldsymbol{x}} \sqrt{\mathcal{N}_{\boldsymbol{x}}(\mu_1, \Sigma_1) \mathcal{N}_{\boldsymbol{x}}(\mu_2, \Sigma_2)} d\boldsymbol{x} \tag{B.5}$$

or :

$$\begin{aligned} \sqrt{\mathcal{N}_x(\mu, \Sigma)} &= |2\pi\Sigma|^{-\frac{1}{4}} |2\pi 2\Sigma|^{\frac{1}{2}} \frac{e^{(\boldsymbol{x}-\mu)^\top (2\Sigma)^{-1}(x-\mu)}}{|2\pi 2\Sigma|^{\frac{1}{2}}} \\ &= 2^{\frac{d}{2}} |2\pi\Sigma|^{\frac{1}{4}} \mathcal{N}_{\boldsymbol{x}}(\mu, 2\Sigma) \end{aligned} \tag{B.6}$$

ainsi :

$$\begin{aligned} C_B(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2)) &= 2^{\frac{d}{2}} |2\pi\Sigma_1|^{\frac{1}{4}} 2^{\frac{d}{2}} |2\pi\Sigma_2|^{\frac{1}{4}} \int_{\boldsymbol{x}} \mathcal{N}_{\boldsymbol{x}}(\mu_1, 2\Sigma_1) \mathcal{N}_{\boldsymbol{x}}(\mu_2, 2\Sigma_2) d\boldsymbol{x} \\ &= 2^d |4\pi^2 \Sigma_1 \Sigma_2|^{\frac{1}{4}} \frac{e^{-\frac{1}{2}(\mu_1-\mu_2)^\top (2\Sigma_1+2\Sigma_2)^{-1}(\mu_1-\mu_2)}}{|2\pi(2\Sigma_1+2\Sigma_2)|^{\frac{1}{2}}} \end{aligned} \tag{B.7}$$

d'après un tutoriel (P. Ahrendt 2005) détaillant des calculs sur les lois normales multivariées. Ce résultat peut aussi se retrouver dans un livre (K. Fukunaga 1990).

Finalement, en utilisant le lien entre le coefficient de Bhattacharyya et la distance de Hellinger, on obtient le résultat suivant :

$$D_H(\mathcal{N}(\mu_1, \Sigma_1), \mathcal{N}(\mu_2, \Sigma_2))^2 = 1 - \sqrt{\frac{\sqrt{|\Sigma_1 \Sigma_2|}}{|\frac{\Sigma_1+\Sigma_2}{2}|}} e^{-\frac{1}{4}(\mu_1-\mu_2)^\top (\Sigma_1+\Sigma_2)^{-1}(\mu_1-\mu_2)} \tag{B.8}$$

Cette formulation a été utilisée pour calculer la distance de Hellinger entre lois normales. Dans le cas 1D ou s'il y a indépendance entre les composantes du vecteur aléatoire, l'expression se simplifie. N'utilisant pas de dimensions très élevées, le temps de calcul des distances reste abordable même si des systèmes linéaires sont à résoudre et si des déterminants sont à calculer.

Publications dans le cadre de cette thèse

S. Schwab, T. Chateau, C. Blanc et L. Trassoudaine (2011). « Clustering de visages : vers la construction automatique d'un album photo à partir d'une séquence vidéo ». *ORASIS – Congrès des jeunes chercheurs en vision par ordinateur*. Praz-sur-Arly, France

S. Schwab, T. Chateau, C. Blanc et L. Trassoudaine (2012). « Suivi de visages par regroupement de détections : traitement séquentiel par blocs ». *Actes de la conférence Reconnaissance des Formes et l'Intelligence Artificielle (RFIA)*. Lyon, France

S. Schwab, T. Chateau, C. Blanc et L. Trassoudaine (2013). « A multi-cue spatio-temporal framework for automatic frontal face clustering in video sequences. » *EURASIP Journal on Image and Video Processing*

N.-S. Vu, S. Schwab, P. Bouges, X. Naturel, C. Blanc, T. Chateau et L. Trassoudaine (2013). « Face Recognition for Video Security Applications ». *Workshop Interdisciplinaire sur la Sécurité Globale*. Troyes, France

Bibliographie

P. Ahrendt (2005). « The multivariate gaussian probability distribution » (cité p. 117).

R. Ahuja, T. Magnanti et J. Orlin (1993). *Network flows : theory, algorithms, and applications*. Englewood Cliffs : Prentice Hall (cité p. 53).

E. Amigó, J. Gonzalo, J. Artiles et F. Verdejo (2009). « A comparison of extrinsic clustering evaluation metrics based on formal constraints ». *Information retrieval* 12.4 (cité p. 26).

A. Andriyenko et K. Schindler (2010). « Globally Optimal Multi-target Tracking on a Hexagonal Lattice ». *Proc. European Conference on Computer Vision*. T. 6311. Heidelberg : Springer (cité p. 15).

O. Arandjelovic et A. Zisserman (2005). « Automatic face recognition for film character retrieval in feature-length films ». *Proc. Computer Vision and Pattern Recognition*. T. 1. IEEE (cité pp. 8, 9).

J. Barr, K. Bowyer, P. Flynn et S. Biswas (2012). « Face Recognition From Video : A Review ». *International Journal of Pattern Recognition and Articial Intelligence* (cité p. 10).

J. Barron, D. Fleet et S. Beauchemin (1994). « Performance of optical flow techniques ». *International Journal of Computer Vision* 12.1 (cité p. 71).

M. Bäuml, M. Tapaswi et R. Stiefelhagen (2013). « Semi-supervised Learning with Constraints for Person Identification in Multimedia Data ». *Proc. Computer Vision and Pattern Recognition*. IEEE (cité p. 10).

B. Benfold et I. Reid (2011). « Stable Multi-Target Tracking in Real-Time Surveillance Video (in press) ». *Proc. Computer Vision and Pattern Recognition*. IEEE (cité pp. 17, 45, 47, 111).

J. Berclaz, F. Fleuret et P. Fua (2009). « Multiple object tracking using flow linear programming ». *Performance Evaluation of Tracking and Surveillance (PETS-Winter)*. IEEE (cité pp. 15, 16, 47).

K. Bernardin et R. Stiefelhagen (2008). « Evaluating multiple object tracking performance : the CLEAR MOT metrics ». *EURASIP Journal on Image and Video Processing* 2008 (cité p. 24).

M. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier et L. Van Gool (2010). « Online multi-person tracking-by-detection from a single, uncalibrated camera ». *IEEE Transactions on Pattern Analysis and Machine Intelligence* (cité pp. 13, 15).

T. Brox, C. Bregler et J. Malik (2009). « Large displacement optical flow ». *Proc. Computer Vision and Pattern Recognition*. IEEE (cité p. 71).

G. Celeux et G. Soromenho (1996). « An entropy criterion for assessing the number of clusters in a mixture model ». *Journal of classification* 13.2 (cité p. 38).

A. Cherian, V. Morellas, N. Papanikolopoulos et S. Bedros (2011). « Dirichlet process mixture models on symmetric positive definite matrices for appearance clustering in video surveillance applications ». *Proc. Computer Vision and Pattern Recognition*. IEEE (cité p. 65).

J. Choi, W. De Neve et Y. Ro (2010). « Towards an automatic face indexing system for actor-based video services in an IPTV environment ». *Consumer Electronics* 56.1 (cité pp. 8, 9).

F. Crete, T. Dolmiere, P. Ladret et M. Nicolas (2007). « The blur effect : perception and estimation with a new no-reference perceptual blur metric ». *Human Vision and Electronic Imaging XII* 6492 (cité p. 96).

D. Dueck et B. Frey (2007). « Non-metric affinity propagation for unsupervised image categorization ». *Proc. International Conference on Computer Vision*. IEEE (cité p. 88).

J. Edmonds et R. Karp (1972). « Theoretical improvements in algorithmic efficiency for network flow problems ». *Journal of the ACM (JACM)* 19.2 (cité p. 53).

M. Everingham, J. Sivic et A. Zisserman (2009). « Taking the bite out of automated naming of characters in TV video ». *Image and Vision Computing* 27.5 (cité p. 10).

C. Fiche (2012). « Repousser les limites de l'identification faciale en contexte de vidéo-surveillance ». Thèse de doct. Université de Grenoble (cité pp. 11, 96).

A. Fitzgibbon et A. Zisserman (2002). « On affine invariant clustering and automatic cast listing in movies ». *Computer Vision—ECCV 2002* (cité p. 9).

A. Fitzgibbon et A. Zisserman (2003). « Joint manifold distance : a new approach to appearance based clustering ». *Proc. Computer Vision and Pattern Recognition*. IEEE Computer Society (cité pp. 9, 107).

S. Foucher et L. Gagnon (2007). « Automatic detection and clustering of actor faces based on spectral clustering techniques ». *Proc. Canadian Conference on Computer and Robot Vision*. IEEE (cité pp. 9, 22).

K. Fukunaga (1990). *Introduction to statistical pattern recognition*. Academic Press Professional, Inc. (cité p. 117).

W. Ge et R. Collins (2008). « Multi-target data association by tracklets with unsupervised parameter estimation ». *Proc. British Machine Vision Conference* (cité pp. 16, 17, 47).

A. Goder et V. Filkov (2008). « Consensus clustering algorithms : Comparison and refinement » (cité p. 22).

M. Grgic, K. Delac et S. Grgic (2011). « SCface–surveillance cameras face database ». *Multimedia tools and applications* 51.3 (cité p. 11).

R. Gross et J. Shi (2001). « The CMU Motion of Body (MoBo) Database » (cité p. 11).

P. Grunwald (2004). « A tutorial introduction to the minimum description length principle ». *Arxiv preprint math* (cité p. 111).

M. Hansen et B. Yu (2001). « Model selection and the principle of minimum description length ». *Journal of the American Statistical Association* 96.454 (cité p. 111).

J. Henriques, R. Caseiro et J. Batista (2011). « Globally optimal solution to multi-object tracking with merged measurements ». *International Conference on Computer Vision*. IEEE (cité pp. 15, 56).

C. Huang, B. Wu et R. Nevatia (2008). « Robust object tracking by hierarchical association of detection responses ». Springer (cité pp. 15, 16, 38, 45, 47, 56).

H.-C. Huang, Y.-Y. Chuang et C.-S. Chen (2012). « Multi-affinity spectral clustering ». *Proc. Conference on Acoustics, Speech and Signal Processing*. IEEE (cité p. 22).

A. K. Jain (2010). « Data clustering : 50 years beyond K-means ». *Pattern Recognition Letters* 31.8 (cité p. 19).

M. Kim, S. Kumar, V. Pavlovic et H. Rowley (2008). « Face tracking and recognition with visual constraints in real-world videos ». *Proc. Computer Vision and Pattern Recognition*. IEEE (cité p. 11).

K. Lee, J. Ho, M. Yang et D. Kriegman (2005). « Visual tracking and recognition using probabilistic appearance manifolds ». *Computer Vision and Image Understanding* 99.3 (cité pp. 10, 11).

K.-C. Lee, J. Ho, M.-H. Yang et D. Kriegman (2003). « Video-based face recognition using probabilistic appearance manifolds ». *Proc. Computer Vision and Pattern Recognition*. T. 1. IEEE (cité p. 11).

B. Leibe, K. Schindler et L. Van Gool (2007). « Coupled detection and trajectory estimation for multi-object tracking ». *Proc. International Conference on Computer Vision*. IEEE (cité pp. 16, 111).

Y. Li, C. Huang et R. Nevatia (2009). « Learning to associate : HybridBoosted multi-target tracker for crowded scene ». *Proc. Computer Vision and Pattern Recognition*. IEEE (cité pp. 16, 17).

B. Lucas, T. Kanade et al. (1981). « An iterative image registration technique with an application to stereo vision » (cité p. 71).

E. Maggio, E. Piccardo, C. Regazzoni et C. A. (2007). « Particle PHD filter for multi-target visual tracking ». IEEE (cité p. 31).

J. Marzat, Y. Dumortier et A. Ducrot (2009). « Real-time dense and accurate parallel optical flow using cuda ». *Proc. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision* (cité pp. 71, 72).

M. C. Nechyba, L. Brandy et H. Schneiderman (2008). « PittPatt Face Detection and Tracking for the CLEAR 2007 Evaluation ». *Multimodal Technologies for Perception of Humans : International Evaluation Workshops CLEAR 2007 and RT 2007* (cité p. 8).

C. Needham et R. Boyle (2003). « Performance Evaluation Metrics and Statistics for Positional Tracker Evaluation » (cité p. 23).

A. Ng, M. Jordan et Y. Weiss (2002). « On spectral clustering : Analysis and an algorithm ». *Advances in neural information processing systems* 2 (cité pp. 21, 22, 78, 79).

T. Oncel, P. Fatih et M. Peter (2006). « Region Covariance : A Fast Descriptor for Detection And Classification » (cité p. 64).

N. Pande, M. Jain, D. Kapil et P. Guha (2012). « The Video Face Book ». *Advances in Multimedia Modeling* (cité pp. 8, 9).

D. Pelleg, A. Moore et al. (2000). « X-means : Extending k-means with efficient estimation of the number of clusters ». 1 (cité p. 38).

X. Pennec, P. Fillard et N. Ayache (2006). « A Riemannian Framework for Tensor Computing ». *Int. J. Comput. Vision* 66.1 (cité p. 65).

A. A. Perera, C. Srinivas, A. Hoogs, G. Brooksby et W. Hu (2006). « Multi-object tracking through simultaneous long occlusions and split-merge conditions ». *Proc. Computer Vision and Pattern Recognition*. T. 1. IEEE (cité p. 56).

P. Perez, C. Hue, J. Vermaak et M. Gangnet (2002). « Color-based probabilistic tracking ». *Computer Vision—ECCV 2002* (cité p. 63).

P. Perona et L. Zelnik-Manor (2004). « Self-tuning spectral clustering ». *Advances in neural information processing systems* 17 (cité p. 22).

P. J. Phillips et al. (2009). « Overview of the multiple biometrics grand challenge » (cité p. 11).

F. Porikli, O. Tuzel et P. Meer (2005). « Covariance Tracking using Model Update Based on Lie Algebra ». *Proc. Computer Vision and Pattern Recognition*. IEEE (cité pp. 65, 70).

J. Prokaj, M. Duchaineau et G. Medioni (2011). « Inferring tracklets for multi-object tracking ». *Proc. Computer Vision and Pattern Recognition*. IEEE (cité p. 15).

N. Rahman, K. Wei et J. See (2006). « RGB-H-CbCr Skin Colour Model for Human Face Detection ». *Proceedings of The MMU International Symposium on Information and Communications Technologies*. M2USIC (cité pp. 45, 96).

D. Ramanan, S. Baker et S. Kakade (2007). « Leveraging archival video for building face datasets ». *Proc. International Conference on Computer Vision*. IEEE (cité p. 9).

S. Schwab, T. Chateau, C. Blanc et L. Trassoudaine (2011). « Clustering de visages : vers la construction automatique d'un album photo à partir d'une séquence vidéo ». *ORASIS – Congrès des jeunes chercheurs en vision par ordinateur*. Praz-sur-Arly, France (cité p. 119).

– (2012). « Suivi de visages par regroupement de détections : traitement séquentiel par blocs ». *Actes de la conférence Reconnaissance des Formes et l'Intelligence Artificielle (RFIA)*. Lyon, France (cité p. 119).

– (2013). « A multi-cue spatio-temporal framework for automatic frontal face clustering in video sequences. » *EURASIP Journal on Image and Video Processing* (cité p. 119).

F. Septier, A. Carmi et S. J. Godsill (2009). « Multiple Object Tracking Using Evolutionary MCMC-Based Particle Algorithms » (cité p. 39).

J. Sivic, M. Everingham et A. Zisserman (2005). « Person spotting : video shot retrieval for face sets ». *Proc. International Conference on Image and Video Retrieval* (cité pp. 8, 9).

– (2009). « "Who are you ?"-Learning person specific classifiers from video ». IEEE (cité p. 10).

K. Smith, D. Gatica-Perez, J.-M. Odobez et S. Ba (2005). « Evaluating Multi-Object Tracking ». *Proc. Computer Vision and Pattern Recognition*. Washington, DC, USA : IEEE (cité pp. 25, 26).

R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofolo, D. Mostefa et P. Soundararajan (2006). « The CLEAR 2006 evaluation ». *Multimodal Technologies for Perception of Humans* (cité p. 24).

A. Strehl et J. Ghosh (2003). « Cluster ensembles—a knowledge reuse framework for combining multiple partitions ». *The Journal of Machine Learning Research* 3 (cité pp. 22, 80).

D. Sun, E. Sudderth et M. Black (2010). « Layered image motion with explicit occlusions, temporal consistency, and depth ordering ». *Advances in Neural Information Processing Systems* 23 (cité p. 71).

Z. Tao, N. Ram et W. Bo (2008). « Segmentation and Tracking of Multiple Humans in Crowded Environments ». *IEEE Trans. Pattern Anal. Mach. Intell.* 30.7 (cité p. 16).

M. Tapaswi, M. Bauml et R. Stiefelhagen (2012). « "Knock ! Knock ! Who is it ?" probabilistic person identification in TV-series ». *Proc. Computer Vision and Pattern Recognition*. IEEE (cité p. 9).

A. Tyagi et J. Davis (2008). « A recursive filter for linear systems on riemannian manifolds ». *Proc. Computer Vision and Pattern Recognition*. IEEE (cité p. 65).

R. C. Verma, C. Schmid et K. Mikolajczyk (2003). « Face detection and tracking in a video by propagating detection probabilities ». *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 25.10 (cité p. 8).

P. Viola et M. Jones (2001). « Rapid Object Detection using a Boosted Cascade of Simple Features ». T. 1. Los Alamitos, CA, USA : IEEE (cité pp. 13, 32).

U. Von Luxburg (2007). « A tutorial on spectral clustering ». *Statistics and Computing* 17.4 (cité pp. 20, 22, 78).

N. Vretos, V. Solachidis et I. Pitas (2011). « A mutual information based face clustering algorithm for movie content analysis ». *Image and Vision Computing* 29.10 (cité pp. 9, 22).

N.-S. Vu (2010). « Contributions à la reconnaissance de visages à partir d'une seule image et dans un contexte non-contrôlé ». Thèse de doct. Institut National Polytechnique de Grenoble-INPG (cité p. 66).

N.-S. Vu, H. M. Dee et A. Caplier (2012). « Face recognition using the POEM descriptor ». *Pattern Recognition* 45.7 (cité p. 66).

N.-S. Vu, S. Schwab, P. Bouges, X. Naturel, C. Blanc, T. Chateau et L. Trassoudaine (2013). « Face Recognition for Video Security Applications ». *Workshop Interdisciplinaire sur la Sécurité Globale*. Troyes, France (cité p. 119).

H. Wang, H. Stone et S. Chang (1999). « FaceTrack : Tracking and summarizing faces from compressed video ». *SPIE Multimedia Storage and Archiving System IV* (cité p. 107).

B. Wu et R. Nevatia (2007). « Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors ». *International Journal of Computer Vision* 75.2 (cité p. 68).

J. Yao et J.-M. Odobez (2008). « Fast Human Detection from Videos Using Covariance Features ». Springer (cité p. 65).

Q. Yu et G. Medioni (2009). « Multiple-Target Tracking by Spatiotemporal Monte Carlo Markov Chain Data Association ». *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.12 (cité pp. 16, 17, 39, 45, 47).

L. Zhang, Y. Li et R. Nevatia (2008). « Global data association for multi-object tracking using network flows ». *Proc. Computer Vision and Pattern Recognition*. IEEE (cité pp. 16, 17, 38–40, 45, 47, 49, 51, 54, 68).