Suivi de visages par regroupement de détections : traitement séquentiel par blocs.

S. Schwab[1,2] T. Chateau[1] C. Blanc[1,2] L. Trassoudaine[1]
[1] LASMEA, Université Blaise Pascal
24, avenue des Landais
63177 Aubière cedex – France
{simeon.schwab, thierry.chateau, christophe.blanc, laurent.trassoudaine}@univ-bpclermont.fr

[2] VESALIS, Parc Technologique de la Pardieu
8, allée Evariste Galois
63000 Clermont-Ferrand – France
christophe.blanc@vesalis.fr

Résumé

Cet article décrit une méthode de partitionnement des visages dans une séquence vidéo ; elle se base sur une méthode de type tracking-by-detections *[19] et utilise une modélisation probabiliste de type Maximum* A Posteriori *résolu par un algorithme s'appuyant sur une recherche de flot de coût minimal dans un graphe. Face aux contraintes de densité, de mouvement et de taille des détections de visage issues de la vidéosurveillance, les travaux présentés apportent deux contributions : (1) la définition de différentes dissimilarités (spatiale, temporelle, apparence et mouvement) combinées de façon simple et (2) la mise en œuvre d'une version séquentielle par blocs d'images qui permet de traiter des flux vidéos. La méthode proposée est évaluée sur plusieurs séquences réelles annotées.*

Mots clés

partitionnement, détections de visages, vidéo, flot optique, Maximum *A Posteriori*

Abstract

This paper presents a clustering method of faces found in a video, it is a tracking-by-detection *based solution inspired by [19] and relies on a Maximum* A Posteriori *probabilistic framework solved by min cost flow algorithms in a graph. Faced to high density, small size and fast movements of faces in video surveillance, this article presents two contributions : (1) the definition of dissimilarities (spatial, temporal, appearance and movement) combined in a simple and effecient way, and (2) the implementation of a sequential version with blocks of frames which can handle video streams.*

Keywords

clustering, face detection, video, optical flow, Maximum *A Posteriori*

1 Introduction

L'efficacité croissante des détecteurs de visages sur une image a permis leur utilisation dans le domaine de la vidéosurveillance. Toutefois, face au grand nombre de détections extraites d'une vidéo, le traitement automatique de ces visages reste encore un problème, notamment lorsque intervient un grand volume de données. Avec le nombre grandissant de vidéos (archives de vidéosurveillance, vidéos personnelles ou vidéos publiques de sites web d'hébergement), un partitionnement automatique des visages, sans nécessairement aborder la reconnaissance faciale, présente déjà un grand intérêt. En effet, il est plus facile et efficace d'avoir accès à un album photo des personnes présentes dans une vidéo (que ce soit dans le cadre d'enquêtes policières ou simplement pour la recherche dans une base personnelle de vidéos).

Les travaux présentés se concentrent sur les situations rencontrées en vidéosurveillance : scènes denses (en nombre de personnes) avec des visages relativement petits (en termes de résolution) et des déplacements erratiques de piétons. Pour la plupart des situations de vidéosurveillance, l'exploitation de techniques de reconnaissance faciale automatiques est difficile, principalement à cause de la faible résolution et du manque de netteté des détections. C'est pourquoi la méthode développée s'applique à utiliser les informations spatiales et temporelles qu'apporte la vidéo, et ne traite pas directement le problème de la reconnaissance faciale.

La première méthode présentée, provenant de travaux

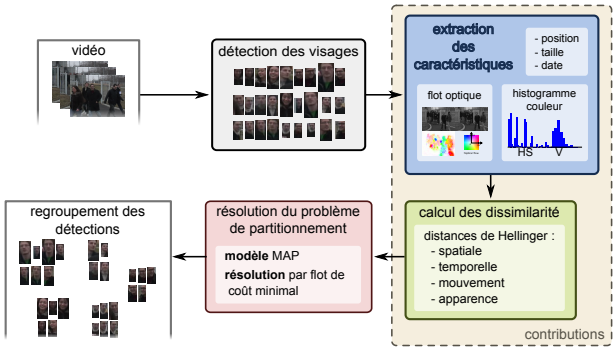

FIGURE 1 – Vue d'ensemble de la méthode de regroupement des visages d'une vidéo. Après une première étape de détection des visages, leurs caractéristiques sont extraites de la vidéo. Ces caractéristiques sont ensuite utilisées pour construire une dissimilarité entre chaque détections. L'étape finale consiste à résoudre le partitionnement de manière probabiliste, en se basant sur les dissimilarités précédemment calculées.

de Nevatia et al. [19], se base sur trois étapes principales : extraction des caractéristiques des détections, calcul des dissimilarités et partitionnement des détections (cf figure 1).

Toutefois, dans le cas de longues vidéos, ou d'une vidéo en cours d'acquisition, cette méthode n'est pas directement applicable, ce qui nous a amené à l'étendre. En respectant les principes de la première méthode, la seconde s'intéresse à la résolution du problème de partitionnement en traitant la vidéo de manière séquentielle, par blocs d'images.

Les deux sections suivantes décrivent le contexte et la méthode sur laquelle se fonde les travaux présentés, ensuite la section 4 décrit les contributions apportées au modèle puis la section 5 présente l'adaptation au traitement par blocs d'images. L'article se termine par la méthodologie expérimentale et les résultats obtenus.

2 État de l'art

Dans le domaine des vidéos publiques telles que les émissions et séries TV, des résultats intéressants concernant l'étiquetage de visages ont été publiés ces dernières années. La plupart des travaux [4, 7, 8, 12, 15, 16] utilisent un détecteur de visage puis un système de suivi et de reconnaissance. Certains ne sont pas complètement automatiques [4, 15], d'autres utilisent des fichiers de sous-titres [7]. La majeure partie de ces systèmes utilisent des vidéos où les techniques de reconnaissance faciale sont exploitables parce que les visages sont bien résolus et les acteurs apparaissent souvent de face. L'utilisation de ces méthodes dans des scènes de vidéosurveillance reste encore difficile.

Dans le cadre de séquences de vidéosurveillance, le partitionnement des visages présente des similarités avec les problématiques du suivi multiple. En effet, si l'on parvient à suivre les têtes des piétons dans la vidéo, il est possible d'obtenir un partitionnement des visages. Certaines méthodes font intervenir un système d'association de détections pour suivre plusieurs personnes dans une vidéo [6, 10, 11, 18, 19].

La plupart des méthodes effectuant une association de données à des fins de partitionnement [2, 18], ou de suivi multiple [7, 9, 15], font intervenir des paramètres pour combiner les différentes dissimilarités ou probabilités de transition. Ces paramètres sont souvent estimés par apprentissage, soit en ligne [2, 9], soit hors-ligne [15, 18]. Nous proposons une approche qui minimise le nombre de paramètres à estimer.

Après avoir modélisé le problème d'association des détections d'une vidéo entière, il est difficile de lui trouver une solution étant donné le grand nombre d'associations potentielles. Plusieurs méthodes sont employées pour résoudre le problème : des échantillonnages basés sur des méthodes de Monte Carlo [2, 9, 18], des méthodes génériques d'optimisation telles que les programmes linéaires [3], l'algorithme Hongrois [10] ou encore les flots de coût minimal [19].

Les méthodes citées sont plutôt utilisées sur la totalité des vidéos, mais pour des raisons de complexité, elles utilisent parfois un système de fenêtres glissantes [2, 18]. Nous proposons dans cet article une méthode qui s'oriente plus vers l'idée d'analyse séquentielle de blocs d'images, tout en maintenant la continuité entre les blocs par une étape d'association inter-blocs. Ce principe est détaillé à la section 5.

3 Modélisation du problème

Cette partie présente la modélisation mise en place par [19] pour répondre au problème du suivi multiple de piétons par association de données.

La méthode s'appuie sur une modélisation probabiliste du problème. Cette modélisation se base sur les observations (constituées des détections) pour obtenir un regroupement des détections par individus. La partition recherchée est un ensemble d'associations maximisant une probabilité *a posteriori*. Pour modéliser cette probabilité, on définit premièrement l'ensemble des observations $Z = \{\boldsymbol{z}_i\}$ dont chaque élément $\boldsymbol{z}_i$ est issu d'une détection. Ces observations sont définies par un ensemble de caractéristiques : position, taille, apparence et date dans la vidéo.

Le partitionnement recherché (parmi l'ensemble $\mathcal{T}$ de tous les partitionnements possibles) est un ensemble de trajectoires $T = \{T_k\}$, où chaque trajectoire est un ensemble de détections : $T_k = \{\boldsymbol{z}_{k_1}, ..., \boldsymbol{z}_{k_{n_k}}\}$ où $k_i = j$ signifie que le détection z_j est le i-ème élément de la trajectoire T_k. Étant donné qu'une détection observée ne peut appartenir qu'à une seule trajectoire (contrainte de non-recouvrement), l'ensemble

des trajectoires constitue bien une partition des détections. Les détections considérées comme des faux positifs sont représentées par une trajectoire particulière de ce partitionnement.
Après avoir défini les observations ainsi que l'état recherché, le problème du Maximum *A Posteriori* s'écrit de la manière suivante :

$$\hat{T} = \arg\max_{T \in \mathcal{T}} P(T|Z) = \arg\max_{T \in \mathcal{T}} P(Z|T)P(T) \quad (1)$$

sous hypothèses d'indépendance et sous la contrainte de non-recouvrement $\forall T_k \neq T_l \in T$, $T_k \cap T_l = \emptyset$ qui permet de bien définir une partition des détections. Nous avons légèrement modifié la formulation de [19] pour clarifier le formalisme. Des termes dépendants des observations étaient initialement dans l'*a priori*, nous les avons simplement déplacés dans le terme de vraisemblance.

3.1 Vraisemblance

La vraisemblance se décompose en deux aspects, la vraisemblance des détections prises indépendamment (P_{FP}) et celles des différentes trajectoires ($P(Z|T_k)$) :

$$P(Z|T) = \prod_{z_i \in Z} P_{FP}(\boldsymbol{z}_i|T) \prod_{T_k \in T} P(Z|T_k) \quad (2)$$

La vraisemblance d'une détection $P(\boldsymbol{z}_i|T)$ est modélisée par une loi de Bernoulli de paramètre β représentant le fait que l'observation soit une vraie détection ou un faux positif (où β représente le taux estimé de faux positifs du détecteur).
Dans notre cas, cette vraisemblance prend aussi en compte une information colorimétrique. La plupart des détecteurs de visages de face n'utilisent pas d'information colorimétrique, ainsi l'ajout de la proportion de pixels dont la couleur est proche de celle de la peau permet d'éliminer certains faux positifs du détecteur.
La vraisemblance d'une détection est définie de la manière suivante :

$$P_{FP}(\boldsymbol{z}_i|T) = \begin{cases} (1-\beta)P_f(a_i) & \text{si } \exists\ T_k \in T \mid z_i \in T_k \\ \beta(1-P_f(a_i)) & \text{sinon} \end{cases} \quad (3)$$

où a_i représente l'apparence de la détection et P_f son taux de pixels couleur de peau. P_f est seuillé pour être au minimum à 1% (au lieu de 0), cela afin de limiter l'exclusion de vrais positifs (ayant peu de pixels considérés comme pixels de peau). La segmentation des couleurs de peau est fixe, elle utilise les frontières colorimétriques proposées dans [14]. Cette segmentation a ses limites, notamment pour les problèmes liés aux couleurs de peau dues aux diversités ethniques et aux variations d'illumination ; elle permet toutefois un apport d'information qui améliore la qualité dans le cas de vidéos couleur.

La vraisemblance d'une trajectoire $P(Z|T_k)$ caractérise le fait que cette trajectoire T_k soit cohérente. Cette cohérence est représentée par les similarités entre les détections formant la trajectoire, elles s'appuient sur l'apparence mais aussi sur les aspects spatiaux et temporels. Elle est modélisée par une chaine de Markov du 1er ordre où chaque état provient d'une détection et où les probabilités de transition sont issues des similarités entre détections.
Pour chaque trajectoire T_k d'une partition T on définit la vraisemblance de la manière suivante :

$$P(Z|T_k) = \prod_{i=1}^{|T_k|-1} P_{link}(\boldsymbol{z}_{k_{i+1}}|\boldsymbol{z}_{k_i}) \quad (4)$$

où P_{link} est la probabilité de transition entre deux détections.

3.2 Probabilité *a priori*

La probabilité *a priori* fait intervenir uniquement la proportion de regroupements. Elle se définit de la façon suivante : $P(T) = P_e^{2|T|}$ où P_e représente la probabilité de démarrer et d'arrêter une trajectoire, elle est estimée par l'inverse du taux de détections par personnes.

3.3 Résolution du MAP

Énumérer les partitionnements d'un ensemble donné est très vite un problème trop complexe pour être envisageable. Il n'est pas concevable de trouver la solution optimale par simple énumération.
Cependant, grâce essentiellement à la modélisation de la vraisemblance des trajectoires par une chaine de Markov, il est possible de trouver une solution optimale au problème (1). Cela peut se faire par résolutions successives de problèmes de flots de coût minimal sur un graphe [19]. Les nœuds du graphe représentent les détections et un chemin sur le graphe représente un groupe de la partition. Les coûts des arcs sont fixés aux dissimilarités entre deux détections. Ainsi, pour une valeur de flot donnée, le flot de coût minimal détermine les associations à faire pour construire une partition. La partition optimale du MAP est ensuite obtenue en faisant varier le flot et en calculant itérativement le flot de coût minimal.

4 Contributions

Cette section présente les contributions apportées à la méthode pour s'adapter au cas de partitionnement des visages avec une paramétrisation simplifiée. Les observations seront définies de la façon suivante $\boldsymbol{z}_i = (\boldsymbol{x}_i, \boldsymbol{s}_i, a_i, t_i, of_i)$, avec :
- $\boldsymbol{x}_i = (x_i, y_i)$: position (en pixels) de la détection
- $\boldsymbol{s}_i = (w_i, h_i)$: hauteur et largeur (en pixels) de la détection
- a_i : descripteur de l'apparence de la détection

– t_i : numéro de l'image dans la vidéo
– $of_i = (\boldsymbol{f}_i, \Sigma_f^i)$: moyenne et covariances du flot optique estimé dans la fenêtre considérée.

Les probabilités de transitions et les caractéristiques des observations seront détaillées par la suite.

4.1 Définition des dissimilarités

Les différentes dissimilarités utilisées pour la vraisemblance des trajectoires font intervenir des grandeurs hétérogènes : date (index de l'image), distance image (en pixels), vitesse (en pixels par frame) et apparence (histogramme couleur). Afin d'associer les distances entre ces grandeurs au sein d'une dissimilarité comparant deux détections, et cela sans paramètres réglant l'apport de chaque distance, nous proposons d'utiliser une unique distance se calculant sur des densités de probabilité.

De nombreuses distances [5] (ou similarités) ont été définies pour comparer des densités ; notre choix s'est porté sur la distance de Hellinger. Ce choix a été motivé par le fait que cette distance soit proche de la divergence de Bhattacharyya (qui est relativement performante pour la comparaison d'histogrammes couleurs) et du fait qu'elle définisse en plus une métrique satisfaisant l'inégalité triangulaire. Voici la définition de la distance de Hellinger :

$$D_H(f,g) = \sqrt{\frac{1}{2}\int \left(\sqrt{f(x)} - \sqrt{g(x)}\right)^2 dx} = \sqrt{1 - BC(f,g)} \quad (5)$$

où f et g sont les densités de probabilité à comparer et où $BC(f,g) = \int \sqrt{f(x)g(x)}dx$ est le coefficient de Bhattacharyya.

Pour deux détections données, nous proposons une probabilité de transition qui prend en compte, non seulement leurs dissimilarités en terme de date, position, taille, et apparence ; mais aussi leur mouvement. Elle se définit de la façon suivante :

$$P_{link}(\boldsymbol{z}_i|\boldsymbol{z}_j) = P_t(t_i|t_j)P_{pos}(\boldsymbol{z_i}|\boldsymbol{z_j})P_m(\boldsymbol{z_i}|\boldsymbol{z_j})P_a(a_i|a_j) \quad (6)$$

Étant donné que la distance a ses valeurs dans $[0,1]$, on peut définir la probabilité de transition :

$$P_{link}(\boldsymbol{z_i}|\boldsymbol{z_j}) = (1 - d_t(t_i,t_j))(1 - d_{pos}(\boldsymbol{z_i},\boldsymbol{z_j}))... (1 - d_m(\boldsymbol{z_i},\boldsymbol{z_j}))(1 - d_a(a_i,a_j)) \quad (7)$$

où les distances d_x sont des distances de Hellinger et sont détaillées dans les sections suivantes.

Dissimilarité temporelle. Pour utiliser la distance entre densités de probabilités, il nous faut définir des densités en rapport avec les dates t_i des observations $\boldsymbol{z_i}$. L'idée retenue est de représenter les temps des observations par des gaussiennes d'écart-type fixé et centrées sur le temps en question :

$$d_t(t_i,t_j) = D_H(\mathcal{N}(t_i,\sigma_t^2),\mathcal{N}(t_j,\sigma_t^2)) \quad (8)$$

où $\mathcal{N}$ représente la distribution normale et σ_t l'écart-type fixé. Cet écart-type est le seul paramètre intervenant dans la probabilité de transition, il permet de régler l'étendue temporelle maximale pour une association. Dans le cas de la méthode par blocs d'images (cf section 5), cet écart-type est fixé par la taille des blocs d'images.

Dissimilarité de position. La même démarche est utilisée pour ce qui est de la distance spatiale. La position et la taille d'une détection sont utilisées pour définir une probabilité de présence sur chaque pixel. Elle est construite comme une loi normale 2D centrée sur la position du centre de la détection et d'écart-type d'une demi-longueur de la taille de la zone détectée. Ainsi :

$$d_{pos}(\boldsymbol{z}_i,\boldsymbol{z}_j) = D_H(\mathcal{N}(\boldsymbol{x}_i,\Sigma_{xy}^i),\mathcal{N}(\boldsymbol{x}_j,\Sigma_{xy}^j)) \quad (9)$$

où $\boldsymbol{x}_i$ est la position en pixels de la détection $\boldsymbol{z}_i$ et où $\Sigma_{xy}^i = \begin{pmatrix} \frac{h_i}{2}^2 & 0 \\ 0 & \frac{w_i}{2}^2 \end{pmatrix}$ définit la matrice de covariance se basant sur la largeur w_i et la hauteur h_i de la zone détectée.

Dissimilarité de mouvement. Comme pour la position, la dissimilarité de mouvement est définie avec une distance de Hellinger entre deux distributions normales de la position estimée à une date donnée.

Nous proposons d'effectuer l'estimation d'une position par celle de la vitesse de déplacement obtenue par calcul d'un flot optique sur les images de la vidéo.

Pour deux détections données, leurs flots optiques respectifs sont utilisés pour prédire les positions respectives à la date moyenne. La précision est prise en compte par la covariance en position (issue des tailles des détections) et la covariance du flot optique.

La dissimilarité du mouvement devient donc :

$$d_m(\boldsymbol{z}_i,\boldsymbol{z}_j) = D_H(\mathcal{N}(\boldsymbol{x}_i + \Delta t\boldsymbol{f}_i, \Sigma_{xy}^i + \Delta t^2\Sigma_f^i), \mathcal{N}(\boldsymbol{x}_j - \Delta t\boldsymbol{f}_j, \Sigma_{xy}^j + \Delta t^2\Sigma_f^j)) \quad (10)$$

où $\Delta t = \frac{t_i - t_j}{2}$ et Σ_f^i représente la matrice de covariance du flot optique estimée à partir des flots optiques de la zone de détection.

Dissimilarité d'apparence. Nous avons choisi de représenter l'apparence par un histogramme HS-V [13]. Cet histogramme est la concaténation d'un histogramme 2D HS et d'un histogramme V des pixels de l'image, où H, S et V sont les trois canaux du système colorimétrique HSV. Afin de filtrer les valeurs et saturations extrêmes, si S et V sont suffisamment grandes pour un pixel donné, ce pixel contribuera à la partie HS de l'histogramme ; sinon il sera compté dans l'histogramme V.

Si l'on ne considère que la zone de détection fournie par le détecteur de visages, l'information colorimétrique

n'est pas suffisante pour distinguer deux visages différents de la même vidéo. Cela nous a amené à étendre la zone de détection en la doublant vers le bas, ainsi la couleur du vêtement contribue à la construction de l'histogramme.
La dissimilarité d'apparence est simplement la distance de Hellinger entre les histogrammes HS-V calculés à partir des détections :

$$d_{app}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \sqrt{1 - \sum_{k=1}^{n} \sqrt{a_i^k a_j^k}} \qquad (11)$$

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

5 Méthode séquentielle par blocs

Dans cette section, nous proposons une extension de la méthode précédente dans le but de traiter des vidéos de manière séquentielle. L'intérêt d'un traitement séquentiel est de pouvoir fonctionner dans un cas temps réel où la vidéo est en cours d'acquisition, ou de pouvoir appréhender de longues vidéos dont le traitement global est trop complexe.

5.1 Algorithme

L'idée principale est de découper la vidéo en blocs de taille fixée. Pour chaque bloc, la méthode précédemment décrite sera utilisée pour faire un premier regroupement des visages. On procède ensuite à un partitionnement prenant en compte les groupes créés sur ce bloc et ceux intervenant au bloc précédent (cf figure 2). L'algorithme suit les étapes suivantes :

Pour tous les blocs d'images :
- détection et extraction des caractéristiques du bloc courant
- **partitionnement intra-bloc** : recherche du MAP sur le bloc courant
- mise à jour des dissimilarités en intégrant les nouvelles trajectoires
- mise à jour des paramètres (P_{eI} et β_I)
- **partitionnement inter-blocs** : recherche du MAP avec les trajectoires des blocs précédents.

Le *partitionnement intra-bloc* correspond à la méthode présentée, non plus appliquée à toute la vidéo mais en se concentrant sur un seul paquet d'images. Pour ce qui est du *partitionnement inter-blocs*, 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-bloc* sur le bloc courant et des groupes ayant au moins un élément dans le bloc précédent (voir figure 2). Cela permet de ne pas perdre la continuité entre deux blocs successifs. Dans le cas du partitionnement intra-bloc, on fixe les paramètres β et P_e empiriquement. Pour ce qui est du partitionnement inter-blocs, ces paramètres (notés P_{eI} et β_I) sont ré-estimés à chaque itération :

$$P_{eI} = \frac{\text{nb personnes}}{\text{nb groupes actifs}} = P_e \frac{\text{nb détections}}{\text{nb groupes actifs}} \qquad (12)$$

car P_e est estimé par $\frac{\text{nb personnes}}{\text{nb détections}}$. Le terme *groupes actifs* désigne les regroupements issus du partitionnement intra-bloc ou issus des partitionnements précédents dont au moins une détection figure dans le bloc de l'itération précédente (cf figure 2). Le paramètre du taux de faux positifs est estimé comme suit :

$$\beta_I = \frac{\text{nb groupes FP}}{\text{nb groupes actifs}} = P_e \frac{\text{nb détections FP}}{\text{nb détections}} = \beta P_{eI} \qquad (13)$$

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.

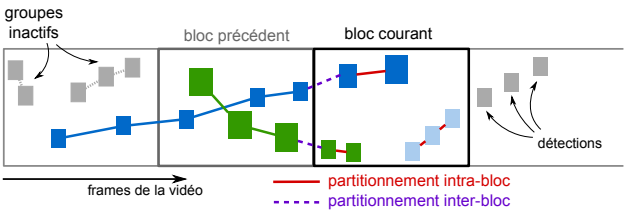

FIGURE 2 – Schéma illustrant les blocs d'images utilisés dans le traitement par blocs.

5.2 Dissimilarités inter-blocs

Ayant décrit la méthode, il nous reste à définir les dissimilarités entre deux groupes de détections. En utilisant toujours les distances de Hellinger définies précédemment, nous avons choisi d'estimer les dissimilarités entre deux groupes par une moyenne des dissimilarités 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).

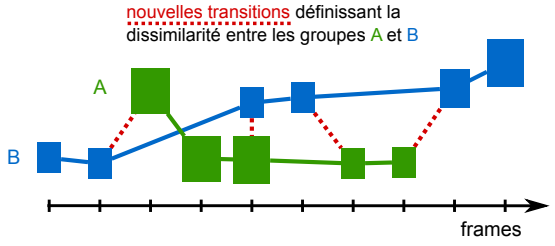

FIGURE 3 – Schéma illustrant les partitionnements intra-bloc et inter-blocs.

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

6 Expérimentations et résultats

6.1 Base d'expérimentation et méthodologie

Les expérimentations s'appuient sur 5 vidéos différentes. Quelques chiffres clés de ces vidéos son donnés dans le tableau 1.

vidéo	passages	images	nb détect.	FP	taille détect.
1	24	1934	1725	2.78	35
2	6	307	200	11.5	35
3	7	384	920	2.61	36
4	6	485	463	3.46	58
5	29	1966	1794	1.56	63

TABLE 1 – Vidéos de l'expérimentation. *passages* : nombre de personnes traversant le champs de vue de la caméra, *images* : nombre total d'images de la vidéo, *nb détect.* : nombre de détections obtenues, *FP* : taux observé de faux positifs du détecteur de visages, *taille détect.* : moyenne des largeurs des détections.

Ces vidéos sont particulièrement difficiles à traiter de par les croisements rapides des piétons, leurs forts changements de direction et la petite taille des détections.
Les expérimentations effectuées se basent sur le détecteur de visages de face issu des travaux de Viola et Jones [17] en utilisant l'implémentation de la bibliothèque C++ OpenCV. Dans les vidéos utilisées, les tailles des détections sont plutôt petites comparées à celles demandées par les problématiques de reconnaissance faciale. Les tailles utilisées ici sont parmi les plus petites détectables avec l'implémentation OpenCV du détecteur. La figure 4 donne un aperçu des détections obtenues.
Le flot optique est calculé par une méthode Lucas-Kanade pyramidale avec 5 niveaux de pyramide. Pour réduire le bruit, à chaque frame, deux flots optiques sont calculés sur l'ensemble de l'image, le premier en utilisant l'image précédente et le deuxième avec l'image suivante. Ces deux flots sont ensuite moyennés.

FIGURE 4 – Aperçu des détections de visages de face, obtenues avec l'implémentation OpenCV du détecteur Viola-Jones. La zone de détection est doublée vers le bas.

La résolution des deux dernières vidéos est de 704×576 pixels, et celle des autres est de 800×600. La vérité terrain du partitionnement a été faite manuellement en regroupant toutes les détections, avec un groupe par passage de personne et un groupe des faux positifs. La figure 5 donne un aperçu des vidéos.

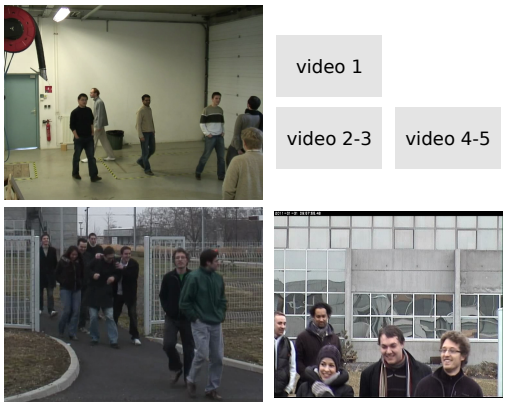

FIGURE 5 – Aperçu des différentes vidéos.

6.2 Critère d'évaluation

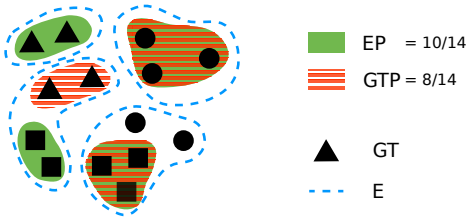

FIGURE 6 – Illustration du calcul de la pureté (EP) et de la pureté inverse (GTP). Elles représentent respectivement la qualité des regroupements et celle des séparations.

La qualité d'un partitionnement, par rapport au partitionnement vérité terrain, est définie comme un compromis entre la qualité des regroupements et celle des séparations. Ce compromis s'exprime comme une F-mesure entre la pureté et la pureté inverse du partitionnement (cf [1]). La figure 6 donne une illustration de ce critère. En supposant un partitionnement vérité terrain $GT = \{GT_j\}$ et un partitionnement estimé $E = \{E_k\}$, on définit la pureté EP et la pureté inverse GTP comme suit :

$$EP = \frac{1}{D} \sum_k \max_j |E_k \cap GT_j| \tag{14}$$

$$GTP = \frac{1}{D} \sum_k \max_j |GT_k \cap E_j| \tag{15}$$

où D est le nombre de détections à partitionner. Ensuite la F-mesure est définie par :

$$F\text{-mesure} = 2\frac{EP \times GTP}{EP + GTP} \tag{16}$$

où EP et GTP sont au préalable normalisées afin d'être à valeurs dans $[0, 1]$.

6.3 Résultats

Le tableau 2 présente les résultats obtenus sur les 5 vidéos de test. Il permet de comparer la qualité des différentes versions de la méthode présentée. Les résultats montrent bien l'apport du flot optique au sein des dissimilarités, tant en moyenne qu'en sensibilité. Le traitement séquentiel par blocs n'atteint pas encore les performances de la première méthode, mais rend dans certains cas le système moins sensible au paramètre P_e.

vidéos	avec flot optique	sans flot optique	algo. par blocs
1	**86.1** (**2.5**)	84.8 (2.8)	80.7 (3.3)
2	**86.9** (**3.7**)	69.9 (4.7)	80.2 (6.0)
3	**88.4** (**0.8**)	85.2 (3.5)	79.2 (0.9)
4	**86.2** (7.9)	70.6 (3.7)	80.1 (**0.9**)
5	**89.8** (4.5)	78.6 (12)	89.3 (**0.6**)

TABLE 2 – F-mesure (en %) des différentes variantes de la méthode : MAP sur la vidéo entière (avec et sans flot optique) et MAP séquentiel par blocs (avec 100 images par blocs). Les valeurs représentent la moyenne et l'écart-type de la F-mesure, obtenus pour 100 valeurs du paramètre P_e.

Le tableau 3 montre le potentiel des méthodes présentées dans l'article par rapport à une méthode générique de partitionnement. Les résultats de la table 3 indiquent que la solution basée sur le MAP permet d'accéder à de meilleures solutions que celles qu'on pourrait obtenir avec un simple clustering ascendant hiérarchique. Le potentiel de la méthode par blocs n'atteint pas encore celui de la méthode globale présentée dans cet article mais dépasse dans plusieurs cas celui de la méthode hiérarchique sur la globalité de la vidéo.

vidéos	MAP	MAP par blocs	hiérarchique
1	**88.8**	86.2	*84.9*
2	**91.6**	89.2	*74.8*
3	**90.5**	85.7	*76.4*
4	**91.4**	*83.0*	87.8
5	**95.6**	*91.0*	92.3

TABLE 3 – Meilleures F-mesures observées pour la méthode sur la vidéo entière (colonne 2), pour la méthode séquentielle avec blocs d'images (colonne 3) et pour un simple clustering hiérarchique (colonne 4). Les expérimentations colonnes 2 et 4 sont faites avec les mêmes matrices de dissimilarité.

L'impact de la taille des blocs d'images sur la qualité du partitionnement n'est pas négligeable. Comme le montre la figure 7, la qualité de la solution tend à s'améliorer avec la taille des blocs, ce qui semblait prévisible. Cela montre aussi l'intérêt d'un traitement par blocs de grandes tailles, et montre que l'information des 200 images passées et futures d'une détection demeure pertinente pour le partitionnement.

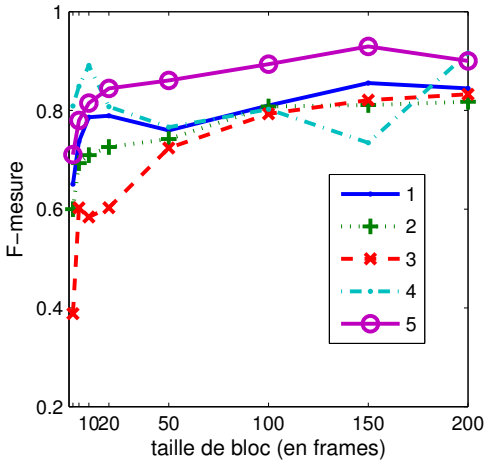

FIGURE 7 – Qualité du partitionnement en fonction de la taille des blocs pour la méthode par blocs, pour les 5 vidéos.

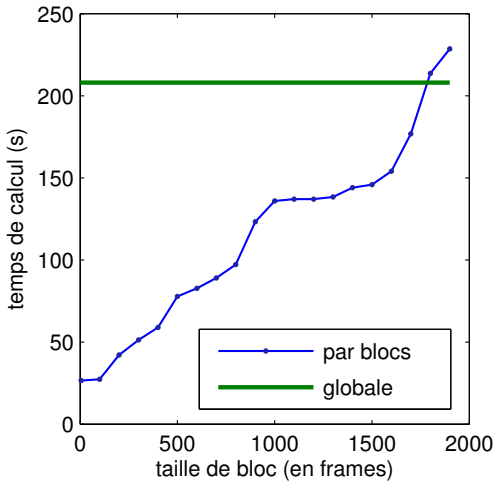

FIGURE 8 – Temps de calcul en fonction de la taille des blocs pour la vidéo 1 (avec 1934 frames). Le temps de calcul ne fait intervenir ici que les calculs des dissimilarités et la résolution du MAP. Le temps pris par les détections et l'extraction des caractéristiques n'est pas comptabilisé, il reste identique suivant la méthode et le nombre de blocs.

Toutefois l'avantage majeur de la méthode séquentielle par blocs est qu'elle diminue considérablement la complexité calculatoire. La figure 8 montre l'évolution du temps de calcul en fonction de la taille des blocs. L'implémentation actuelle de la méthode n'est pas optimisée, elle fait intervenir des scripts *MATLAB* mais les parties les plus critiques en temps de calcul sont implémentées *via* des bibliothèques C++.

Conclusion

Cet article propose une méthode de partitionnement de visages dans des cas de vidéos particulièrement contraignantes en terme de mouvement et de taille des visages. En se basant sur un modèle probabiliste de type MAP, nous avons pu mettre en place des dissimilarités simples en terme de paramétrisation et tout de même efficaces, en utilisant notamment le flot optique pour estimer localement le mouvement des visages. Une version séquentielle de la méthode a été élaborée et expérimentée, elle n'atteint pas encore la qualité de la méthode plus globale mais concurrence une méthode globale plus générique. Elle a aussi l'avantage d'être bien plus rapide.

Remerciements

Nous tenons à remercier la société *Vesalis* pour son financement, l'équipe du projet *Biorafale* pour les acquisitions vidéos ainsi que les figurants du *LASMEA*.

Références

[1] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4) :461–486, 2009.

[2] B. Benfold and I. Reid. Stable multi-target tracking in real-time surveillance video (in press). In *Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society Conference, 2011.

[3] J. Berclaz, F. Fleuret, and P. Fua. Multiple object tracking using flow linear programming. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter)*, pages 1–8. IEEE, June 2009.

[4] T.L. Berg, A.C. Berg, J. Edwards, M. Maire, R. White, Y.W. Teh, E. Learned-Miller, and D.A. Forsyth. Names and faces in the news. In *Computer Vision and Pattern Recognition*, volume 2, pages 848–854. IEEE Computer Society, 2004.

[5] S.H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. In *International Journal of Mathematical Models and Methods in Applied Sciences*, volume 1, pages 300–307, 2007.

[6] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. A mobile vision system for robust multi-person tracking. In *Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[7] M. Everingham, J. Sivic, and A. Zisserman. Taking the bite out of automated naming of characters in tv video. *Image and Vision Computing*, 27(5) :545–559, 2009.

[8] S. Foucher and L. Gagnon. Automatic detection and clustering of actor faces based on spectral clustering techniques. In *Fourth Canadian Conference on Computer and Robot Vision (CRV'07)*, pages 113–122. IEEE, 2007.

[9] W. Ge and R. Collins. Multi-target data association by tracklets with unsupervised parameter estimation. In *British Machine Vision Conference*, 2008.

[10] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. *Proceedings of the 10th European Conference on Computer Vision : Part II*, pages 788–801, 2008.

[11] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *Proceedings 11th International Conference on Computer Vision*. IEEE, 2007.

[12] Michael C. Nechyba, Louis Brandy, and Henry Schneiderman. Pittpatt face detection and tracking for the clear 2007 evaluation. *Multimodal Technologies for Perception of Humans : International Evaluation Workshops CLEAR 2007 and RT 2007*, pages 126–137, 2008.

[13] P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. *Computer Vision—ECCV 2002*, pages 661–675, 2002.

[14] N. Rahman, K. Wei, and J. See. Rgb-h-cbcr skin colour model for human face detection. In *Proceedings of The MMU International Symposium on Information and Communications Technologies*. M2USIC, 2006.

[15] D. Ramanan, S. Baker, and S. Kakade. Leveraging archival video for building face datasets. In *11th International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007.

[16] Josef Sivic, Mark Everingham, and Andrew Zisserman. Person spotting : video shot retrieval for face sets. In *International Conference on Image and Video Retrieval (CIVR)*, pages 226–236, 2005.

[17] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition*, 1 :511–I–518 vol.1, 2001.

[18] Qian Yu and Gérard Medioni. Multiple-target tracking by spatiotemporal monte carlo markov chain data association. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12) :2196–2210, 2009.

[19] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.