Clustering de visages : vers la construction automatique d'un album photo à partir d'une s equence vidé o

Siméon Schwab, Thierry Chateau, Christophe Blanc, Laurent Trassoudaine

Clustering de visages : vers la construction automatique d'un album photo à partir d'une séquence vidéo

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
{thierry.chateau,laurent.trassoudaine,simeon.schwab}@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 présente une méthode de regroupement de détections d'un même objet vu sur une séquence vidéo, en se plaçant dans le cadre applicatif plus précis de la construction automatique d'un album photo. Nous utilisons une méthode d'analyse globale, basée sur une formalisation probabiliste du problème d'association de données. La solution du problème est alors donnée par une estimation du Maximum A Posteriori (MAP). La principale contribution concerne l'utilisation d'une méthode de suivi locale avant-arrière appliquée à chaque détection. Cela afin d'enrichir l'information d'apparence issue de la détection, par une information spatiale provenant de la construction de pistes locales. Nous introduisons une nouvelle mesure de vraisemblance basée sur la dissimilarité spatio-temporelle entre les pistes. L'algorithme obtenu est alors capable d'adresser des situations où les détections de visages sont éparses. Nous proposons d'utiliser des critères dérivés de la pureté et la pureté inverse d'un clustering pour évaluer les performances de la méthode proposée. La méthode est ensuite comparée à un clustering ascendant hiérarchique, sur deux séquences test réelles.

Mots Clef

clustering, détection visage, maximum *a posteriori*, multi-suivi visuel

Abstract

This paper presents a clustering method of detections of the same object seen on a video. We apply it to the context of the automatic construction of photo album. We use a global analysis, based on a probabilistic framework of data association problems. The solution is given by Maximum A Posteriori estimation. Our main contribution concerns the use of a local front-back tracking, applied to each detection ; to increase appearance information of detections with a spatial information, through local tracks construction. We introduce a new likelihood measure based on the spatio-temporal dissimilarity between tracks. The algorithm is then able to deal with situations in which the face detections are scattered. We propose to use criteria derived from purity and inverse purity of a clustering to assess performances of the proposed method. This method is compared to hierarchical clustering on two real test sequences.

Keywords

clustering, face detection, maxium *a posteriori*, tracking, multiple visual tracking

1 Introduction

Les détecteurs de visages sur images statiques sont de plus en plus courant et performants, cependant, pour leur application aux séquences de vidéosurveillance, il est nécessaire d'ajouter une phase de labellisation. En effet, regrouper les détections de visages présente un grand intérêt pour l'analyse en vidéosurveillance, notamment lors de fouilles d'archives vidéos. Par exemple, il serait intéressant d'avoir une méthode qui extrait automatiquement un album photo des passants d'une séquence de vidéosurveillance.
Nous proposons une méthode de regroupement de détections d'un même objet vu sur une séquence vidéo, en se plaçant dans le cadre applicatif plus précis de la construction automatique d'un album photo. L'état de

l'art associé à ces travaux se classe en deux principales catégories. D'une part, il est possible de modéliser le problème d'étiquetage des détections par une approche séquentielle, où pour une image donnée, on cherche à labelliser les objets détectés par rapport à l'historique des observations. On se ramène alors à une problématique d'association de données. La gestion de l'apparition et de la disparition de nouvelles pistes est également une problématique. Dans [2], les auteurs proposent une méthode de suivi séquentiel d'un nombre variable d'objets, basée sur un filtre séquentiel probabiliste. D'autre part, il est possible d'adresser le problème de manière globale, en considérant que la totalité des observations est disponible lors de l'analyse. Les travaux les plus visibles dans ce domaine proposent de modéliser le problème par une estimation d'un Maximum *A Posteriori* (MAP), dont l'état recherché est l'ensemble des trajectoires et les observations sont des détections de la vidéo. Après modélisation du problème, la recherche d'un MAP fait appel à différentes méthodes : modélisation stochastique de la probabilité *a posteriori* [13] ou différentes méthodes d'optimisation : programmation linéaire [3], algorithme hongrois [4] [9] ou flot de coût minimal [14]. La méthode envisagée dans cet article est basée sur une estimation du MAP résolu par recherches successives de flot de coût minimal sur un graphe [14].
La première partie de cet article est consacrée à la description de la méthode proposée.
La seconde partie expose les critères mis en place pour comparer les performances des méthodes de clustering d'objets sur des séquences d'images, ensuite des expérimentations à partir de séquences vidéo réelles sont présentées.
L'article se termine par des conclusions et perspectives.

2 Méthode

Cette section présente la méthode basée sur [14] et les extensions apportées pour traiter le problème des détections de visages. Ensuite nous définissons plus en détails les dissimilarités mises en œuvre pour traiter le problème.

2.1 Association de données par MAP

En reprenant la formalisation probabiliste de [14] où l'état et les observations sont des variables aléatoires, les détections de visages constituent les observations et l'ensemble des associations de ces détections constitue l'état recherché.
Les observations tiennent compte de la position dans l'image, dans la séquence et de l'apparence des détections. On observe un ensemble de détections $\mathcal{Z} = \{\boldsymbol{z}_i\}$, dont chaque élément sera noté : $\boldsymbol{z}_i = (\boldsymbol{x}_i, \boldsymbol{s}_i, \boldsymbol{a}_i, t_i)$ où $\boldsymbol{x}_i$ représente la position (abscisse et ordonnée en pixel) de la détection, $\boldsymbol{s}_i$ sa taille (hauteur et largeur en pixel), $\boldsymbol{a}_i$ l'apparence qui dépend du descripteur choisi, et t_i la date (numéro de frame) dans la vidéo.
L'état recherché est un ensemble de trajectoires $\mathcal{T} = \{T_k\}$, où les trajectoires sont des groupes de détections : $T_k = \{\boldsymbol{z}_{k_1}, ..., \boldsymbol{z}_{k_{n_k}}\}$, cela signifie que le i-ième élément de la trajectoire k correspond à la détection $\boldsymbol{z}_{k_i}$. En faisant l'hypothèse de non recouvrement (*ie* une détection ne peut se retrouver sur plusieurs trajectoires) et en ajoutant un groupe des mauvaises détections, l'ensemble des trajectoire recherchées est en fait un partitionnement des détections.
Le problème de l'estimation du MAP s'écrit :

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

en ajoutant les contraintes de non-recouvrement et sous hypothèses d'indépendances :

$$\hat{T} = \arg\max_{T \in \mathcal{T} \text{ et } \forall k \neq l \ , \ T_k \cap T_l = \emptyset} \prod_i P(\boldsymbol{z}_i|T) \prod_k P(T_k)$$

La vraisemblance explique le fait qu'une observation $\boldsymbol{z}_i$ soit vraie ou soit une fausse alarme, elle est modélisée par une loi de Bernoulli :

$$P(\boldsymbol{z}_i|T) = \begin{cases} 1-\beta & \text{si } \boldsymbol{z}_i \text{ est dans une traj. de } T \\ \beta & \text{sinon} \end{cases}$$

où β est le taux de faux positifs du détecteur.
$P(T_k)$ représente le fait que les détections de T_k soient cohérentes en position, temps et apparence. Elle est modélisé par un processus markovien d'ordre 1, dont les états sont les détections de T_k :

$$\begin{aligned} P(T_k) &= P_e P_{link}(\boldsymbol{z}_{k_1}|\boldsymbol{z}_{k_0}) P_{link}(\boldsymbol{z}_{k_2}|\boldsymbol{z}_{k_1}) ... \\ &\quad P_{link}(\boldsymbol{z}_{k_{l_k}}|\boldsymbol{z}_{k_{l_k-1}}) P_e \end{aligned}$$

où P_e représente la probabilité de démarrer et d'arrêter une trajectoire. La probabilité de transition P_{link} va être modélisée par un produit de probabilités prenant en compte l'apparence P_a, le mouvement P_m et le temps P_t :

$$P_{link}(\boldsymbol{z}_i|\boldsymbol{z}_j) = P_a(\boldsymbol{z}_i|\boldsymbol{z}_j) P_m(\boldsymbol{z}_i|\boldsymbol{z}_j) P_t(\boldsymbol{z}_i|\boldsymbol{z}_j)$$

Cette formulation va être conservée par la suite, et la dissimilarité entre deux détections se définit à partir de la log-vraisemblance par :

$$\begin{aligned} d(\boldsymbol{z}_i, \boldsymbol{z}_j) &= -log(P_{link}(\boldsymbol{z}_i|\boldsymbol{z}_j)) \\ &= -log(P_a(\boldsymbol{z}_i|\boldsymbol{z}_j)) - log(P_m(\boldsymbol{z}_i|\boldsymbol{z}_j)) \\ &\quad -log(P_t(\boldsymbol{z}_i|\boldsymbol{z}_j)) \\ &= \tilde{d_a}(\boldsymbol{z}_i, \boldsymbol{z}_j)^2 + \tilde{d_m}(\boldsymbol{z}_i, \boldsymbol{z}_j)^2 + \tilde{d_t}(\boldsymbol{z}_i, \boldsymbol{z}_j)^2 \end{aligned}$$

avec $\tilde{d_a}$ la dissimilarité associée à la log-vraisemblance de l'apparence, $\tilde{d_m}$ celle associée au mouvement et $\tilde{d_t}$ celle associée au temps vidéo.
Pour donner aux différentes dissimilarités le même poids dans la dissimilarité totale, on procède simplement à une réduction des distances :

$$\tilde{d_x}(\boldsymbol{z}_i, \boldsymbol{z}_j) = \frac{d_x(\boldsymbol{z}_i, \boldsymbol{z}_j)}{\sigma_x}$$

où σ_x est l'écart type estimé statistiquement sur l'ensemble des dissimilarités x calculées – x étant : apparence, mouvement ou temps. Les différentes dissimilarités sont détaillées à la section 2.4.
Pour chaque situation deux paramètres sont à estimer : le taux de faux positifs β du détecteur et la probabilité P_e de démarrer une trajectoire. Ces paramètres sont estimés statistiquement à partir de vidéos de test :

$$\begin{aligned} \beta &= \frac{\text{faux positifs}}{\text{nombre de détections}} \\ P_e &= \frac{\text{nombre de personnes}}{\text{nombre de détections}} \end{aligned}$$

Dans le cadre des détections de visages, les pertes de détections ne peuvent pas être gérées, comme pour [14], avec une méthode de gestion explicite des occlusions. Les pertes de détections ne se trouvent pas forcément lors d'occlusions, le détecteur de visages est en défaut dès que la personne n'est plus de face.

2.2 Extension des détections

Pour palier au problème des longs espaces entre les détections nous proposons une extension : travailler avec de larges écarts temporels et augmenter les détections par des suivis vers l'arrière et l'avant, pour enrichir l'information spatio-temporelle des détections. Dans la pratique, on n'envisage de lier qu'uniquement les détections dont l'écart temporel ne dépasse pas un certain seuil – cf. 2.4. Si ce seuil est trop important des erreurs de regroupement apparaissent et la complexité calculatoire augmente. L'idée est d'enrichir les détections pour augmenter le seuil sans ajouter trop d'erreurs.
Les observations ne sont plus de simples détections mais un ensemble de patchs localisés spatialement et temporellement. Chaque patch est représenté de la même manière que les détections – cf. section 2.1. Cela revient à étendre les observations $\boldsymbol{z}_i$:

$$\tilde{\boldsymbol{z}_i} = \{\boldsymbol{z}_i^1, \boldsymbol{z}_i^2, ..., \boldsymbol{z}_i^{N_i}\}$$

où $\boldsymbol{z}_i^1$ est identique au z_i défini précédemment, où les $\boldsymbol{z}_i^k = (\boldsymbol{x}_i^k, \boldsymbol{s}_i^k, t_i^k)$ sont issus du suivis et où N_i est le nombre d'éléments de la piste. Les dissimilarités seront modifiées en conséquence – cf. section 2.4.
Le suivi est effectué par la recherche de la position et la taille du patch à la frame suivante. Celles-ci minimisent la dissimilarité en apparence entre la détection et le patch courant. La fenêtre de recherche est bornée à l'aide d'une vitesse maximale de déplacement (de l'ordre de 50 pixel/frame en pratique) et d'un pourcentage d'agrandissement pour la taille – de l'ordre de 25%. La recherche est effectuée par une minimisation de la dissimilarité d'apparence sur la position et la taille des patchs. On utilise une méthode de Nelder-Mead basée sur les simplexes avec comme point de départ la position et taille précédente. Le suivi est lancé pour chaque détection, vers son passé et son futur par rapport au temps de la vidéo.

2.3 Résolution du MAP

L'énumération de l'ensemble des solutions pour trouver le MAP n'est pas envisageable. Dans l'absolu cela reviendrait à rechercher le nombre de partitionnements possibles d'un ensemble, ce qui devient vite inaccessible. Comme le montre [14], il est possible de trouver le MAP en un temps convenable, en utilisant un algorithme de recherche d'un flot de coût minimal sur un graphe. Les détections sont représentées par les nœuds du graphe et les clusters sont assimilés à des chemins. Les coûts des arcs sont fixés par l'opposé des log-vraisemblances (*ie* les dissimilarités) entre deux détections et ainsi le flot de coût minimal (pour une valeur de flot donnée) sélectionnera les arcs construisant le clustering du maximum *a posteriori* pour un nombre donné de partitions. Ainsi la recherche du MAP global se fait itérativement par calculs successifs de flot de coût minimal.

2.4 Définition des dissimilarités

Les définitions des dissimilarités, telles que décrites par la suite, sont applicables aux détections étendues par leurs pistes. Ces définitions restent valable dans le cas où les détections ne sont pas étendues — *ie* où $\tilde{\boldsymbol{z}_i} = \{\boldsymbol{z}_i\}$.

Apparence. Pour mesurer la différence d'apparence entre deux pistes, on compare uniquement les détections ayant engendré les deux pistes.
Nous avons choisi de représenter les détections par un histogramme HS-V [6]. Ces histogrammes sont la concaténation d'un histogramme HS et d'un histogramme V des pixels de l'images – où H, S et V représentent teinte, saturation et valeur. Si les valeurs S et V d'un pixel sont suffisamment grandes, elles seront comptées dans l'histogramme HS, sinon dans l'histogramme V. Pour mesurer la dissimilarité entre deux histogrammes HS-V nous utilisons une distance de Bhattacharyya.
Pour améliorer les résultats, nous étendons la détection de visage à une zone sous la tête pour récupérer une information colorimétrique du piéton, cela est fait en doublant la taille de la zone de détection vers le bas.
La dissimilarité d'apparence ne prend pas en compte

l'extension par suivi : elle ne compare que l'apparence des détections de départ. Ce choix est motivé par le fait que les pistes semblent mieux se comparer au niveau des détections, et aussi parce qu'il est plus difficile de comparer efficacement deux ensembles d'apparence plutôt que deux apparences.
Des matrices de covariance ont aussi été testées pour décrire l'apparence. Dans la littérature, ces descripteurs sont utilisés tant pour le suivi [7] que pour la détection [10] [12], c'est ce qui nous a initialement poussé à les utiliser. L'idée est d'avoir un descripteur qui utilise tant la position des pixels que leur couleur en observant leurs corrélations et pas uniquement des statistiques pour chaque paramètres pris indépendamment, ce qui est, par exemple, le cas des histogrammes.
Pour s'approcher des résultats obtenus avec les comparaisons d'histogrammes couleur, nous devions construire la matrice de covariance comportant : la position, les canaux R, G et B ainsi que les gradients en X et Y de ces trois canaux, soit une matrice de taille 11 – cf. figure 4 pour le comparatif. Le principal problème des matrice de covariance reste la "distance" entre elles. Se plaçant dans le cadre des variétés riemanniennes [5] pour mesurer une distance entre matrices de covariances, il devient difficile de comparer deux distances faisant intervenir quatre matrices de covariances différentes, et donc la distance n'est pas théoriquement applicable au clustering.

Mouvement. La dissimilarité de mouvement va se baser pleinement sur le fait que l'on possède des suivis des détections, cependant elle peut aussi s'utiliser dans le cas d'une simple détection.
Pour quantifier la présence de continuité entre deux suivis, on procède à une interpolation de la fin d'une trajectoire jusqu'au début de l'autre, et vice-versa. Ensuite on mesure la distance entre la position finale de l'extrapolation et le début de l'autre trajectoire – cf figure 1. L'extrapolation est faite de manière très simple : en estimant la vitesse par une moyenne de différences finies à partir des derniers points d'une extrémité de piste. Afin d'utiliser cette mesure avec des observations singleton, on fixe la vitesse à zéro. Si les suivis sont recouvrant (*ie* les deux suivis ont des frames en commun), alors on moyenne les distances spatiales sur les frames communes.

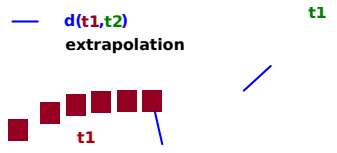


FIGURE 1 – Dissimilarité de mouvements entre deux trajectoires non-recouvrantes.

Pour obtenir la dissimilarité de mouvement entre deux trajectoires, on moyenne les écarts entre les extrapolations d'une trajectoire et la position de l'autre trajectoire. Cette moyenne des deux écarts est obtenue par moyenne de différences finies, et est pondérée par le nombre d'éléments qu'on a pu utiliser pour estimer la vitesse. En pratique on borne le nombre de différences finies pour l'estimation de la vitesse, cela afin de rendre compte d'éventuelles fortes accélérations.
En notant l'intersection des frame $T^{ij}_{inter} = \{t^1_i, ..., t^{N_i}_i\} \cap \{t^1_j, ..., t^{N_j}_j\}$ et en supposant que $\boldsymbol{z}^1_i$ est avant $\boldsymbol{z}^1_j$, on définit la dissimilarité de mouvement :
– si non-recouvrement (*ie* $T^{ij}_{inter} = \emptyset$) :

$$\tilde{d_m}(\tilde{\boldsymbol{z}_i}, \tilde{\boldsymbol{z}_j}) = \frac{K_i d_{pos}(\hat{\boldsymbol{z}_i}, \boldsymbol{z}_j) + K_j d_{pos}(\hat{\boldsymbol{z}_j}, \boldsymbol{z}^{N_i}_i)}{K_i + K_j}$$

où $\hat{\boldsymbol{z}_i}$ représente la position de l'extrapolation en avant de $\tilde{\boldsymbol{z}_i}$ et $\hat{\boldsymbol{z}_j}$ celle en arrière de $\tilde{\boldsymbol{z}_i}$, et où K_i (resp. K_j) est le nombre d'éléments utilisés pour estimer la vitesse issue de $\tilde{\boldsymbol{z}_i}$ (resp. $\tilde{\boldsymbol{z}_j}$)
– si recouvrement :

$$\tilde{d_m}(\tilde{\boldsymbol{z}_i}, \tilde{\boldsymbol{z}_j}) = \frac{\sum_{t \in T^{ij}_{inter}} d_{pos}(\boldsymbol{z}_i^{k^i(t)}, \boldsymbol{z}_j^{k^j(t)})}{|T^{ij}_{inter}|}$$

où $k^i(t) = n$ si $t^n_i = t$ et où d_{pos} mesure les distance entre les positions et les tailles des deux observations.

Temps. La dissimilarité temporelle est définie comme le plus petit écart entre les débuts et les fins des deux suivis à comparer :

$$d_t(\tilde{\boldsymbol{z}_i}, \tilde{\boldsymbol{z}_j}) = \begin{cases} \infty & \text{si } t_{min}(t_i, t_j) < \Delta t \\ t_{min}(t_i, t_j) & \text{sinon} \end{cases}$$

où Δt est fixé empiriquement et :

$$t_{min}(t_i, t_j) = \min(|t_i - t_j|, |t^{N_i}_i - t^{N_j}_j|, |t_i - t^{N_j}_j|, |t^{N_i}_i - t_j|)$$

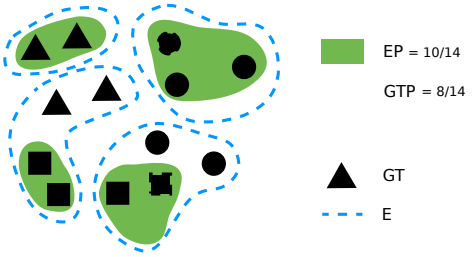
3 Évaluation

Cette section présente la méthode d'évaluation mise en œuvre pour comparer les résultats, puis les vidéos utilisées pour les expérimentations. Ensuite, nous montrons des résultats comparant les différentes méthodes et dissimilarités utilisées.

3.1 Méthode d'évaluation

Il existe plusieurs manières de mesurer la qualité d'un clustering : des méthodes intrinsèques (en mesurant la proximité des éléments d'un cluster et éloignement des clusters) et des méthodes extrinsèques qui utilisent des données expert. Dans notre cas, on s'intéresse aux méthodes extrinsèques étant donné que l'on a la possibilité de classer les détections par un humain pour avoir une vérité terrain.

L'évaluation extrinsèque d'un clustering est un sujet de discussions [1] et il existe de nombreuses mesures de qualité, comme des comptages de bonnes et mauvaises paires, des mesures de puretés ou d'entropie et leurs variantes.
Pour des raisons de simplicité et d'interprétation, nous avons choisi d'utiliser une mesure se basant sur la pureté et la pureté inverse, que nous appellerons pureté de l'estimation (EP) et pureté vérité-terrain (GTP). Par la suite on désignera un clustering obtenu par un algorithme par "clustering estimé" en opposition au clustering de la vérité-terrain.



on voit que les minima des puretés ne sont pas nuls, mais des constantes dépendant uniquement du nombre de détections et du plus grand cluster de la vérité terrain. Afin de mieux pouvoir comparer différents algorithmes de regroupement des détections, nous avons apporté des modifications pour ramener les EP et GTP minimales à 0.

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

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

La partition est meilleure quand EP et GTP sont toutes les deux proches de 1, la F-mesure entre EP_n et GTP_n peut être employée pour mesurer la qualité d'un clustering. Elle est définie par :

$$F = 2\frac{EP_n \times GTP_n}{EP_n + GTP_n}$$

Dans notre cas, il semble intéressant de ne pas cou-

FIGURE 3 – Aperçu des vidéos utilisées pour l'expérimentation. *Gauche :* defile, *droite :* lasmea

la proportion de pixels de peau ne dépasse pas un seuil fixé ont été enlevées. Étant donné que notre étude ne porte pas sur la détection, nous nous sommes permis de choisir empiriquement un seuil qui paraissait convenable pour notre expérimentation.

Procédé expérimental. Nous comparons l'approche MAP avec un clustering hiérarchique *single-link* basée sur la même mesure de dissimilarité que la méthode MAP, sur les bases *lasmea* et *defile*, en utilisant le critère EP-GTP pour les cas suivants :

- différentes mesures d'apparence : corrélation croisée normalisée (*zncc*), matrice de covariance (position, RGB et gradients RGB) avec distances riemanniennes (*CovMatRGBgg*), distance de Bhattacharyya entre histogrammes RGB (*BhattacharyyaRBG*) et entre histogrammes HS-V (*BhattacharyyaHS-V*)
- avec et sans prise en compte de l'information spatio-temporelle
- avec et sans construction de pistes.

3.3 Résultats

La figure 4 montre l'évolution du couple EP-GTP au cours des itérations des algorithmes. On constate que les différentes mesures d'apparence sont assez proches. Pour la suite, l'histogramme HS-V a été choisi parce qu'il est légèrement plus pertinent que les autres pour le MAP et sur la vidéo *defile*.
Des résultats comparant le clustering hiérarchique et la méthode basée sur le MAP sont donnés à la figure 5.

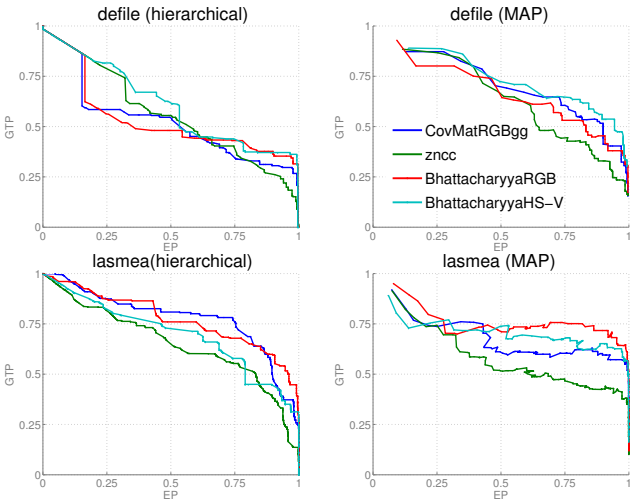


FIGURE 4 – Diagramme EP-GTP comparant différentes dissimilarités d'apparence.

On voit que la méthode MAP est meilleure tant avec les dissimilarités issues de P_m et P_t que sans. Cela explique que la méthode MAP est plus adaptée au problème que le clustering hiérarchique. En ajoutant les pistes aux détections (figure 6), on peut voir que la simple méthode de clustering hiérarchique approche les résultats obtenus par le MAP. Toutefois il est important de noter que les graphiques représentent les différentes solutions par lequel l'algorithme est passé, et que le clustering hiérarchique ne fournit pas directement une solution contrairement à la méthode MAP.

3.4 Synthèse

Sur la plupart des expérimentations, la méthode MAP est plus efficace que le clustering hiérarchique, mais surtout MAP permet de sélectionner une solution. Pour ce qui est de l'ajout du suivi aux détections, le clustering hiérarchique nous montre qu'une information est bien ajoutée et que les résultats sont meilleurs, dans le cas du MAP l'apport est moins flagrant.

4 Conclusion et perspectives

Nous avons proposé une méthode permettant de regrouper par personnes les détections de visages obtenues sur une séquence vidéo. Cette dernière est une extension des travaux de [14] au cas où les détections sont enrichies par des pistes.
Une méthodologie de comparaison des performances pour différents algorithmes et dissimilarités a été présentée. Des expérimentations sur des données réelles ont permis de montrer l'apport de la méthode par rapport à l'état de l'art. Néanmoins des expérimentations complémentaires, sur un plus grand nombre de vidéos, doivent être réalisées pour confirmer cette conclusion.
Un intérêt de la méthode présentée est qu'elle pourrait être utilisée de manière itérative, afin de traiter encore

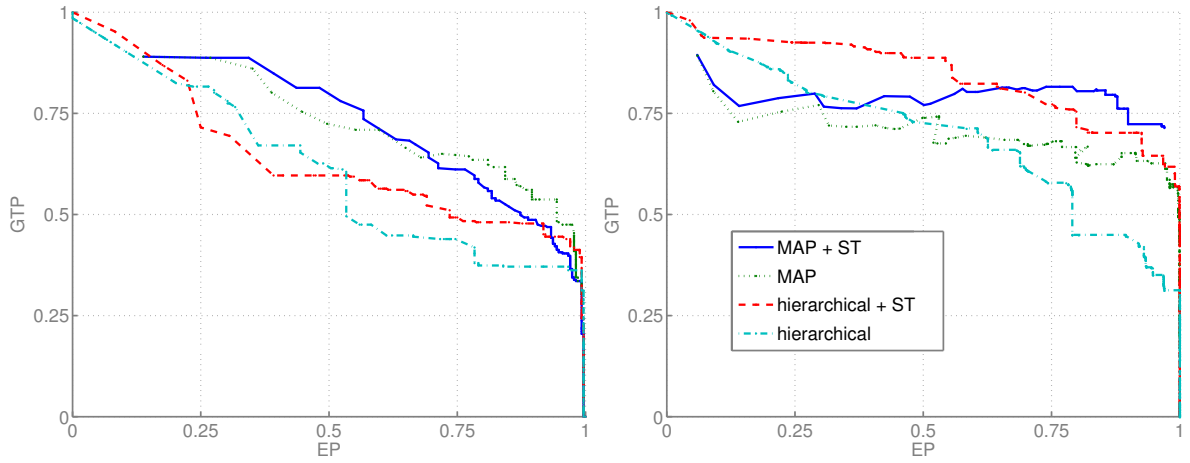


FIGURE 5 – Diagramme EP-GTP comparant les deux algorithmes avec et sans dissimilarités spatio-temporelles (ST). *Gauche :* defile, *droite :* lasmea

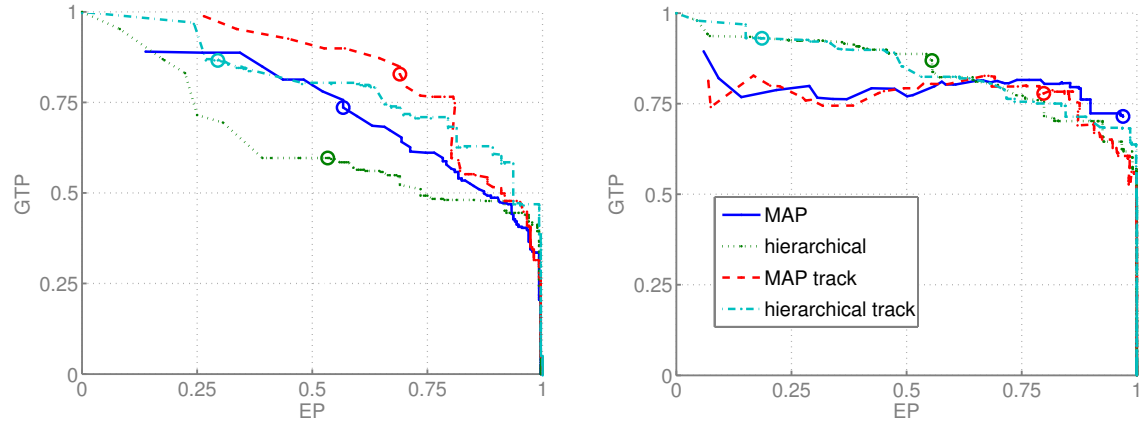


FIGURE 6 – Exemple de diagramme EP-GTP comparant les deux algorithmes avec ou sans suivis (en avant et en arrière) au préalable. Les ronds représentent l'optimum trouvé. *Gauche :* defile, *droite :* lasmea

plus efficacement de longues pertes de détections par notamment un apport en précision des extrapolations des pistes.

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] F. Bardet, T. Chateau, and D. Ramadasan. "illumination aware mcmc particle filter for long-term outdoor multiobject tracking and classification". ICCV, 2009.

[3] J. Berclaz, F. Fleuret, and P. Fua. "multiple object tracking using flow linear programming". (10), 2009.

[4] C. Huang, B. Wu, and R. Nevatia. "robust object tracking by hierarchical association of detection responses". *Computer Vision–ECCV 2008*, pages 788–801, 2008.

[5] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. "a riemannian framework for tensor computing". *Int. J. Comput. Vision*, 66(1) :41–66, 2006.

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

[7] Fatih Porikli, Oncel Tuzel, and Peter Meer. "covariance tracking using model update based on lie algebra". 2005.

[8] N. Rahman, K. Wei, and J. See. "rgb-h-cbcr skin colour model for human face detection". 2006.

[9] Andreas Stergiou, Ghassan Karame, Aristodemos Pnevmatikakis, and Lazaros Polymenakos. "the ait 2d face detection and tracking system for clear 2007". pages 113–125, 2008.

[10] Oncel Tuzel, Fatih Porikli, and Peter Meer. "region covariance : A fast descriptor for detection and classification". pages 589–600, 2006.

[11] Paul Viola and Michael Jones. "rapid object detection using a boosted cascade of simple features". *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1 :511–I–518 vol.1, 2001.

[12] Jian Yao and Jean-Marc Odobez. "fast human detection from videos using covariance features". 2008.

[13] 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.

[14] L. Zhang, Y. Li, and R. Nevatia. "global data association for multi-object tracking using network flows". 2008.