Un système de suivi multi-objets utilisant une stratégie d'association en trois passes adapté à la vidéosurveillance

M.Rogez[1,2,3] L. Robinault[1,3] et L.Tougne[1,2]

[1]Université de Lyon, CNRS
[2]Université Lyon 2, LIRIS, UMR5205, F-69676, France
[3]Foxstream, Vaulx-en-Velin, France

Résumé

*Le suivi multi-objets est une des thématiques centrales de l'analyse vidéo du fait de son large champ d'application. Nous nous intéressons ici plus particulièrement aux applications en vidéo-surveillance. Ainsi, nous décrivons un ensemble d'améliorations destinées à l'algorithme de suivi multi-objets proposé par [LFP*13]. En particulier, nous généralisons le suivi en retirant la spécialisation faite pour les piétons ; nous intégrons le modèle de scène et de visualisation développé dans [RTR13] afin de permettre un raisonnement tridimensionnel permettant de mieux gérer les occultations ; et enfin nous améliorons le mécanisme de formation et destruction des groupes d'objets grâce à l'introduction d'une passe d'association supplémentaire ainsi que d'un critère de similarité de recouvrement. Enfin, nous évaluons le système proposé sur des vidéos synthétiques et réelles afin de montrer l'apport de nos modifications. L'algorithme proposé améliore sensiblement les performances générales par rapport à la version originale, notamment pour la création et destruction des groupes, et ouvre la possibilité d'un raisonnement tridimensionnel.*

Mots clé : Suivi multi-objets, vidéo-surveillance, groupes, automate fini

1. Introduction

Ce document traite du suivi multi-objets en ligne à partir d'une caméra unique, fixe et calibrée. Plus précisément, nous destinons ces travaux aux applications en vidéosurveillance, où s'appliquent des contraintes de traitement en temps réel (et potentiellement de plusieurs flux sur un même ordinateur), ainsi que la nécessité de pouvoir suivre simultanément différents types d'objets (piétons, voitures ou camions par exemple). Ce sujet est très important en analyse vidéo, car il permet d'obtenir les trajectoires des objets suivis et ainsi autorise une prise de décision basée sur des informations de plus haut niveau que la simple image brute (par exemple pour la détection de trajectoire anormale, ou celle des points d'entrée et de sortie de la scène). En pratique, cependant, un certain nombre de difficultés rendent l'obtention de ces trajectoires difficile : par exemple, comment détecter les objets à suivre ? Comment être robuste face aux mauvaises détections ? Comment gérer les occultations inter-objets ? Comment garantir le maintien de l'identité de chaque objet même après occultation ?

La plupart des algorithmes récents de suivi multi-objets se spécialisent dans le suivi de piétons [BRL*09, FJ14] et ainsi utilisent des détecteurs spécialisés à cet effet, en général en utilisant des HOG et une classification à base de SVM. Cette spécialisation permet d'être robuste face aux fausses détections, aux changements de luminosité et aux occultations partielles. En revanche, ces approches sont souvent coûteuses en temps de calcul, et nécessitent un apprentissage souvent délicat en pratique étant donné la variabilité des points de vue et des poses à considérer. En particulier, cette spécificité de détection peut être vue comme une faiblesse dans certains contextes, comme la détection d'intrusion, où les individus peuvent pénétrer en rampant par exemple, ce qui peut mettre en défaut les détecteurs classiques, car entrainés sur des exemples de piétons debouts. Ou bien, lorsque justement on souhaite réaliser un suivi générique, multi-classes (piétons, voitures et camions par exemple).

Une alternative aux détecteurs spécialisés, mentionnés ci-dessus, est d'utiliser un modèle de fond afin de détecter les objets d'intérêts. Cette approche est largement utilisée en pratique [PA10, CFPV10, RKDL12, GMG12, LFSV12, LFP*13], car elle est souvent plus rapide et plus générique que les détecteurs spécialisés. Cependant, l'utilisation d'un modèle de fond génère souvent plus d'erreurs liées aux changements des conditions d'éclairage, ou aux mouvements de la caméra ou du fond. De plus, raisonner sur un masque de détection pose un certain nombre de contraintes nouvelles par rapport à l'utilisation d'un détecteur spécialisé (voir figure 1) : les composantes connexes extraites du masque de détection (que l'on désignera par "Blob" dans ce qui

suit) peuvent ne pas représenter un seul objet réel, mais un groupe d'objets (blob 1 dans la figure 1). Les causes peuvent être multiples : occultation totale, occultation partielle, forte proximité. De plus, il est possible qu'un objet réel ne soit pas représenté par un seul blob, mais morcelé en plusieurs petits blobs (blobs 3, 4 et 5 dans la figure 1). C'est pourquoi, les approches utilisant un masque de détection doivent résoudre explicitement ces problèmes.

Les travaux présentés dans [LFP*13], sont ceux sur lesquels nous nous appuyons, car ils proposent une formulation adaptée au suivi d'objets à partir de détections sur un masque de mouvement. En particulier, ils proposent un algorithme d'associations objets suivis/blobs détectés capable de gérer les associations multiples qui peuvent se manifester lors d'occultations ou à cause de défauts de segmentation. De plus, ils utilisent la notion de groupe afin de suivre collectivement les objets occultants et occultés pendant la durée de l'occultation ; les identités des individus formant le groupe sont cependant conservées afin de pouvoir les réassigner correctement à la fin de l'occultation. Enfin, ils modélisent l'évolution des objets suivis à l'aide d'un automate fini. Cette modélisation a plusieurs intérêts : d'une part, c'est une représentation compacte et directement intelligible de l'état d'un objet (à l'inverse d'une densité de probabilité par exemple). D'autre part, cette modélisation permet d'adapter les traitements et les valeurs de paramètres spécifiquement pour chaque objet en fonction de son état.

Bien que largement inspirée par les travaux de [LFP*13], dont la formulation est particulièrement adapté à la vidéosurveillance, notre méthode se distingue par les changements majeurs qui suivent :

- **Généralisation du suivi :** nous retirons la spécialisation du suivi aux seuls piétons. Ceci inclut notamment la suppression de la classification des objets et blobs, et donc l'apprentissage associé.
- **Prise en compte de la 3d :** nous tirons parti de la calibration de la caméra pour opérer en 3d, notamment en estimant la position au sol, ainsi que les dimensions réelles des objets suivis.
- **Prise en compte des bâtiments environnants :** en utilisant les travaux de [RTR13], nous avons intégré la connaissance des bâtiments environnants afin de délimiter la zone effective de suivi.
- **Amélioration de la formation/destruction des groupes :** d'une part, un nouvel algorithme d'association incluant une première passe permettant de réaliser les associations très probables a été proposé. Cette passe supplémentaire permet d'évacuer les associations simples très probables qui pourraient autrement interférer avec les formations/destructions de groupe. D'autre part, nous avons introduit, un critère de similarité supplémentaire, basé sur le recouvrement entre la silhouette de l'objet et le blob considéré. Ce critère additionnel est utilisé spécifiquement pour détecter les associations multiples, à la place du critère de forme original peu adapté dans ce cas.
- **Possibilité de re-rentrer dans la scène :** dans la version originale, l'automate fini n'autorise pas un objet sortant à re-rentrer dans la scène, alors qu'en pratique cette transition s'avère utile.

Le reste du document s'organise de la manière suivante : un aperçu global de l'algorithme est présenté dans la partie 2.1. Nous présentons ensuite les différents composants de cet algorithme dans les sections suivantes : à savoir, l'évaluateur de similarité en section 2.2, l'algorithme d'association en section 2.3, la procédure de mise à jour des modèles d'objets en section 2.4, l'automate fini en section 2.5 et l'intégration du contexte 3d de la scène en section 2.6. Nous présentons une évaluation qualitative et quantitative de notre algorithme dans la section 3, et finalement concluons sur une synthèse des principaux résultats ainsi que les perspectives d'évolution dans la partie 4.

2. Méthode

2.1. Aperçu de l'algorithme

Avant de présenter l'algorithme, nous souhaitons préciser la définition des termes suivants :

Un objet : est une modélisation d'une entité réelle dont on souhaite réaliser le suivi ; par exemple, un piéton ou une voiture. A chaque objet nous associons, un identifiant unique (ID), un modèle (voir section 2.4) ainsi qu'un état (voir table 1 et figure 5) qui permet de synthétiser de manière compacte et intelligible l'évolution de l'objet.

Un blob : est une composante connexe issue du masque de mouvement.

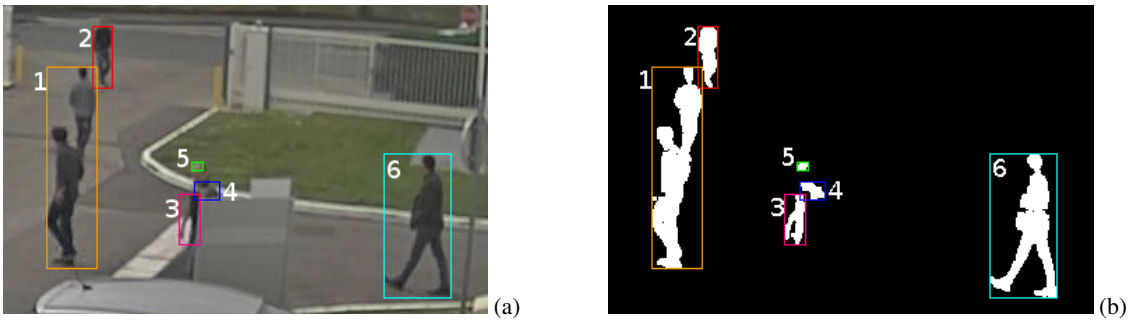
Un groupe : est une modélisation d'un ensemble d'entités réelles qui ne sont plus suivies individuellement, mais collectivement. Comme pour les objets, le groupe dispose d'un ID, d'un modèle et d'un état, mais aussi des identifiants des objets qu'il contient. Ceci permet de réassigner correctement, les identités des objets lorsque le groupe se sépare.

L'algorithme de suivi (figure 2) comporte quatre étapes : La première consiste à extraire les blobs du masque de mouvement. Pour chaque composante connexe extraite du masque de mouvement, nous calculons également un ensemble de caractéristiques de position (centroïde, position réelle au sol), de forme (boite englobante, aire, taille réelle) et d'apparence (histogramme de couleur RGB), qui serviront à évaluer la similarité entre un modèle d'objet et un blob.

La deuxième permet d'associer les objets précédemment suivis et les blobs détectés à l'image courante. Le processus d'association est réalisé en trois passes consécutives (détaillées en section 2.3), et s'appuie sur une mesure de similarité explicité en section 2.2. À l'issue de ces trois passes, les objets non associés sont considérés "perdus" et chaque blob non associé donne naissance à un nouvel objet. Cette étape gère, en outre, les cas d'occultation inter-objets, ainsi que les formations et séparations de groupes.

La troisième étape réalise la mise à jour des modèles des objets. Cette mise à jour dépend des associations réalisées à l'étape précédente et de l'état courant de l'objet. La logique de mise à jour est détaillée en section 2.4.

Enfin, la dernière étape met à jour l'état des objets en réalisant les transitions éventuelles conformément à l'automate fini présenté en section 2.5.



(a) (b)

Figure 1: *Exemples de problèmes liés à l'emploi d'un masque de mouvement : Le blob 1 correspond à un groupe de 2 objets, l'un occultant l'autre. Les blobs (3,4,5) correspondent à des morceaux d'un même objet. Les blobs 2 et 6 correspondent bien à des piétons correctement segmentés.*

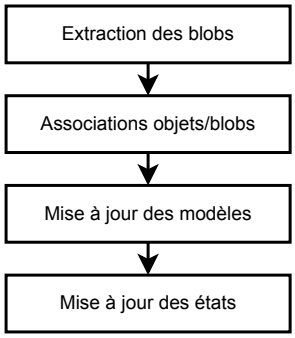


Figure 2: *Grandes étapes de l'algorithme de suivi*

2.2. Calcul des similarités objets/blobs

Pour évaluer la similarité entre l'objet i et le blob j, nous utilisons quatre critères de similarité, chacun donnant une valeur comprise entre 0 (pas similaires) et 1 (identiques). À noter également, que nous compensons le retard temporel des objets sur les blobs (les objets suivis relèvent de l'image à $t-1$, alors que les blobs sont des détections à l'instant t), en prédisant les modèles des objets à l'instant t à partir du modèle à l'instant $t-1$ grâce à un filtre de Kalman (plus de détail en section 2.4).

Le premier critère porte sur la distance entre centroïdes de l'objet i et du blob j :

$$s_{ij}^p = \begin{cases} 1 - d_{ij}/d_{\max} & \text{si } d_{ij} \leq d_{\max} \\ 0 & \text{sinon.} \end{cases} \quad (1)$$

où $d_{\max}$ est la distance maximale admissible entre objets et blobs, et d_{ij} est la distance entre le centroïde prédit de l'objet i et celui du blob j. Ainsi si l'objet i et le blob j sont proches, s_{ij}^p tendra vers 1, et si ils sont lointains, s_{ij}^p tendra vers 0.

Le deuxième critère porte sur la forme (aire et hauteur réelle) de l'objet i et du blob j :

$$s_{ij}^s = 1 - \sqrt{\frac{(\Delta A_{ij})^2 + (\Delta h_{ij})^2}{2}} \quad (2)$$

où ΔA_{ij} et Δh_{ij} représentent la différence relative entre l'aire (resp. la hauteur réelle) de l'objet i et du blob j. Ainsi si les aires et hauteurs réelles de l'objet i et du blob j sont proches, leur différence relative sera nulle et donc s_{ij}^s sera proche de 1.

Le troisième critère porte sur l'apparence. Plus précisément, il mesure le recouvrement entre les histogrammes de couleur de l'objet i et du blob j de la manière suivante :

$$s_{ij}^a = \mathrm{BC}_{i,j} = \frac{1}{B}\sum_{b=1}^{B}\sqrt{h_i(b)h_j(b)} \quad (3)$$

Ici, B est le nombre de classes d'un histogramme de couleur, et $\mathrm{BC}_{i,j}$ est le coefficient de Bhattacharyya entre l'histogramme de couleur h_i du modèle d'objet i, et l'histogramme de couleur h_j du blob j. Pour des histogrammes présentant un fort recouvrement, s_{ij}^a tendra vers 1. En revanche, pour des histogrammes n'ayant pas un bon recouvrement s_{ij}^a tendra vers 0.

Le dernier critère porte sur le recouvrement de la silhouette prédite du cuboïde représentant l'objet i, et le blob j :

$$s_{ij}^o = \begin{cases} 1 & \text{si la silhouette prédite du cuboïde de l'objet } i \\ & \text{intersecte le blob } j \\ 0 & \text{sinon.} \end{cases} \quad (4)$$

A l'inverse des autres critères, qui varient continument entre 0 et 1, le critère de recouvrement est binaire, et ne prend donc que les valeurs 0 ou 1. Il permet de favoriser la détection de la création et de la destruction des groupes.

Finalement, nous définissons la similarité entre un objet i et un blob j par

$$s_{ij} = \sqrt{\frac{\alpha_p(s_{ij}^p)^2 + \alpha_s(s_{ij}^s)^2 + \alpha_a(s_i^a j)^2 + \alpha_o(s_{ij}^o)^2}{\alpha_p + \alpha_s + \alpha_a + \alpha_o}} \quad (5)$$

Les coefficients α définissent les poids de chacun des critères de similarité. Nous détaillerons les valeurs choisies pour ces poids dans la section 2.3. Par construction, s_{ij} est compris entre 0 et 1 et mesure la similarité entre l'objet i et le blob j.

2.3. Association objets/blobs

L'algorithme d'association prend en compte l'état des objets à associer afin de spécialiser les traitements. Aussi nous

invitons le lecteur à se familiariser avec les différents états, en lisant la description de ceux-ci en section 2.5, avant de poursuivre la présentation de l'algorithme d'association.

Par ailleurs, l'algorithme d'association objets/blobs utilise la notion de matrice de similarité S, dont les éléments s_{ij} sont les indices de similarité entre l'objet o_i et le blob b_j, tels que définis dans la section 2.2. Nous illustrons différents cas d'une telle matrice à la figure 3.

L'algorithme d'association se déroule en trois passes successives, chacune tentant d'associer les objets et blobs n'ayant pas encore été associés. Un aperçu de ces trois passes est donné à la figure 4, et sont décrites dans ce qui suit.

La première passe (notée ϕ_1), consiste à associer les objets et blobs ayant une forte similarité. Durant cette phase, tous les objets hormis ceux dans l'état EN GROUPE ou SUPPRIMÉ sont considérés. Les poids des différents critères de similarité sont choisis ainsi : $(\alpha_p = 1, \alpha_s = 0, \alpha_a = 0, \alpha_o = 0)$ pour les objets NOUVEAU, car pour ceux-ci seul la position est jugée fiable ; $(\alpha_p = 0, \alpha_s = 1, \alpha_a = 1, \alpha_o = 0)$ pour les objets PERDU, car pour eux ni la position ni le recouvrement ne peuvent être considérés fiables ; $(\alpha_p = 1, \alpha_s = 1, \alpha_a = 1, \alpha_o = 0)$ pour les objets SUIVI, STABLE et SORTANT. La stratégie d'association est de type glouton (voir algorithme (b) figure 4), et le seuil minimal d'association T_1 est choisi élevé (ici 0.9).

La deuxième passe (notée ϕ_2), tente d'associer les blobs et objets SUIVI, STABLE ou SORTANT restants en prenant en compte les possibilités d'associations multiples (voir algorithme (c) figure 4). Les poids des différents critères sont les suivants $(\alpha_p = 1, \alpha_s = 0, \alpha_a = 1, \alpha_o = 1)$. Le seuil minimal d'association T_2 est fixé à 0.66. Le raisonnement permettant de prendre en compte les associations multiples est le suivant : une fois la meilleure association sélectionnée (o_i, b_j), on recherche dans la ligne i, d'autres blobs notés $\mathcal{B}$, distinct de b_j dont la similarité avec l'objet o_i est supérieure au seuil T_2. On procède de même pour la colonne j, où l'on recherche d'autres objets $\mathcal{O}$, distinct de o_i dont la similarité avec le blob b_j est supérieure au seuil T_2. On a alors plusieurs cas : Si $\mathcal{B} = \mathcal{O} = \emptyset$, on est dans le cas d'association simple 1 objet, 1 blob : l'association (o_i, b_j) est validée. Si $\mathcal{B} \neq \emptyset$ et $\mathcal{O} = \emptyset$, on est dans le cas où l'objet o_i peut être associé à plusieurs blobs. Ce cas peut correspondre soit à un défaut de segmentation (*ie* où l'objet o_i est morcelé en plusieurs blobs) et dans ce cas il convient de fusionner les blobs correspondants en un seul blob et associer ce dernier avec l'objet o_i ; soit à la fin d'une occultation (*ie* si o_i est un groupe), dans ce cas, le groupe o_i est détruit, et on procède à une association gloutonne simple (algorithme (b) figure 4, avec un seuil $T_g = 0.66$ et des poids $(\alpha_p = 0, \alpha_s = 1, \alpha_a = 1, \alpha_o = 0)$) entre les membres du groupe et les blobs de $\mathcal{B} \cup b_j$. Si $\mathcal{O} \neq \emptyset$ et $\mathcal{B} = \emptyset$, on est dans le cas où le blob b_j peut être associé à plusieurs objets. Nous considérons que ceci marque le début d'une occultation, et dans ce cas il convient de créer un groupe comportant les objets $\mathcal{O} \cup o_i$, et d'associer celui-ci au blob b_j. Si $\mathcal{O} \neq \emptyset$ et $\mathcal{B} \neq \emptyset$, par souci de simplicité, nous négligeons le fait qu'il y ait d'autres blobs que b_j, et nous nous ramenons au cas de création d'un groupe.

Pour la troisième et dernière passe (notée ϕ_3), tous les objets non encore associés, hormis ceux dans l'état EN GROUPE ou SUPPRIMÉ, sont considérés afin d'être associés aux blobs restants. Les poids des différents critères de similarité et la stratégie d'association (algorithme (b) figure 4) sont les mêmes que pour la première passe (ϕ_1), seul le seuil minimal d'association change : il est choisi plus bas ($T_3 = 0.75$) afin de permettre d'effectuer les associations qui n'auraient pas été faites durant les deux passes précédentes.

Enfin, à l'issue de ces trois phases d'associations, pour chaque blob restant (non associé), un nouvel objet est créé. Les objets restants sont marqués comme non associés.

L'intérêt de ce déroulement en trois passes est le suivant : la première passe (ϕ_1) permet de propager les associations relativement sûres, et permet de diminuer le nombre d'objets et blobs en entrée de la seconde passe (ϕ_2), plus complexe et donc plus enclin à faire de mauvaises associations. La deuxième passe (ϕ_2) gère les association multiples, et utilise le critère de recouvrement au lieu du critère de forme afin de mieux détecter les formations et destruction de groupes. La troisième passe (ϕ_3) tente d'associer les objets restants, quitte à utiliser un seuil d'association moindre, afin d'éviter la création erronée de nouveaux objets ou le marquage erroné d'objets comme non associé.

2.4. Modèle d'objet : initialisation et mise à jour

Un modèle d'objet comporte les caractéristiques suivantes :

- **position et taille 2d :** un centroïde (x, y), des dimensions de boite englobante 2d (w, h), et une aire A.
- **position, orientation et taille 3d :** un cuboïde 3d posé sur le sol à la position $(x_c, y_c, 0)$, de dimensions (w_c, d_c, h_c) dirigé dans la direction (θ_z).
- **apparence :** histogramme de couleur RGB quantifié. Nous utilisons un histogramme 3d, et quantifions les composantes R, G et B sur 3, 3 et 2 bits respectivement.

L'initialisation d'un modèle d'objet est faite à partir des caractéristiques du blob ayant conduit à la création de l'objet. Le cas délicat d'initialisation du cuboïde 3d est non trivial et mérite quelques explications.

Le but est d'estimer les paramètres du cuboïde conduisant à un masque de détection donné. Nous formulons ce problème d'estimation des paramètres du cuboïde, comme un problème d'optimisation d'énergie. Nous appelons $r(x_c, y_c, w_c, d_c, h_c, \theta_z)$ la fonction qui dessine la silhouette du cuboïde défini par les paramètres donnés, telle qu'elle serait vue par la caméra ; m est le masque de mouvement courant ; $R1$ la boite englobante de la silhouette, $R2$ la boite englobante du blob, et $I_{R1 \cup R2}$ la fonction indicatrice de la région $R1 \cup R2$. Nous définissons alors l'énergie suivante :

$$E(x_c, y_c, w_c, d_c, h_c, m) = \\ ||(r(x_c, y_c, w_c, d_c, h_c, \theta_z) - m).I_{R1 \cup R2}||_{L2} \quad (6)$$

Cette énergie correspond à la distance L2 entre la silhouette prédite du cuboïde et le masque de détection sur la région $R1 \cup R2$.

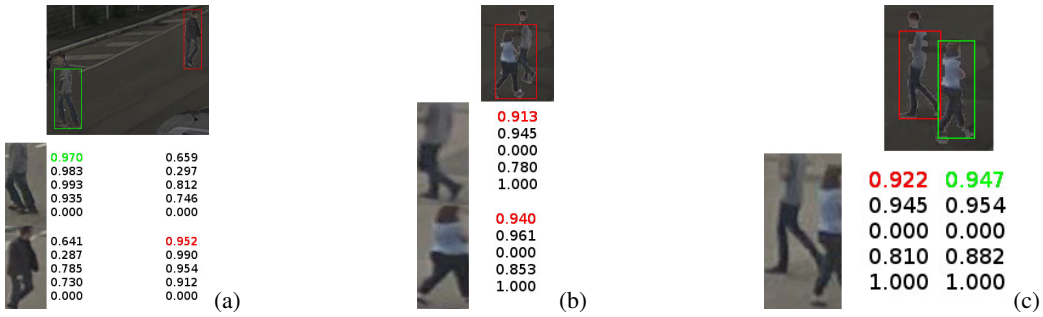


Figure 3: *Illustration des matrices de similarité et de différents cas d'association. Sur les matrices, les modèles d'objets sont sur la première colonne, les blobs détectés sur la première ligne. Un quintuplet de valeurs (s_{ij}, s^p_{ij}, s^s_{ij}, s^a_{ij}, s^o_{ij}) est donné dans cet ordre pour chaque couple (objet i, blob j). Les associations réalisées sont colorées. (a) Cas d'associations simples, chaque objet est associé au blob lui correspondant le mieux. (b) Cas de début d'occultation détecté (ie deux objets correspondent au même blob), un groupe sera donc créé à partir de ces deux objets. (c) Cas de fin d'occultation détecté, (ie un groupe correspond à deux blobs), le groupe sera donc détruit et les objets composant le groupe seront réaffectés aux blobs.*

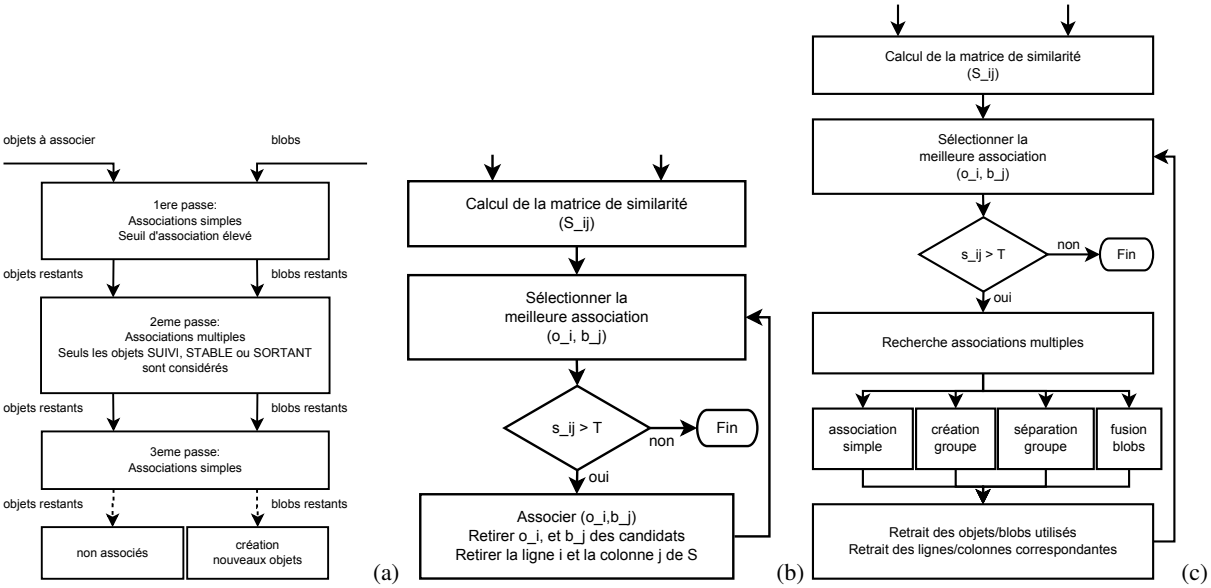


Figure 4: *Algorithmes d'association : (a) : aperçu ; (b) : algorithme glouton pour les associations simples ; (c) algorithme glouton pour les associations multiples*

L'estimation des paramètres du cuboïde est alors réalisée via la minimisation suivante :

$$[x_c, y_c, w_c, d_c, h_c, \theta_z]_0 = \text{argmin}(E(x_c, y_c, w_c, d_c, h_c, \theta_z)) \quad (7)$$

Ne disposant pas des informations de gradient d'énergie pour guider l'optimisation, nous utilisons la méthode de Nelder-Mead [NM65], qui ne requière que les valeurs de la fonction (pas ses dérivées). Cette méthode d'optimisation itérative, utilise la notion de simplexe, que l'on déforme, déplace et réduit progressivement afin que ses sommets se rapprochent d'un point où la fonction est minimale. Cette approche est heuristique, et ne garantie pas la convergence vers un minimum global, c'est pourquoi, nous adoptons la stratégie donnée par l'algorithme 1 pour pallier ces inconvénients et garantir un temps d'exécution borné ($N \times M$ itérations).

Pour faciliter l'estimation initiale, nous réduisons le nombre de paramètres à optimiser en forçant une orientation $\theta_z = 0$, et en imposant une base carrée pour le cuboïde ($w_c = d_c$). Ces contraintes sont relâchées pour les estimations ultérieures afin de permettre au système de converger vers une solution plus proche de la réalité.

Comme nous l'avons déjà évoqué dans la partie 2.2, nous utilisons un filtre de Kalman pour lisser les tailles et positions de la boite englobante 2d et 3d des objets suivis. Plus précisément, nous faisons l'hypothèse de déplacement à vitesse constante, ainsi le vecteur d'état du filtre de Kalman est le suivant :

$$[x, y, \dot{x}, \dot{y}, w, h, \dot{w}, \dot{h}, x_c, y_c, \dot{x_c}, \dot{y_c}, w_c, d_c, h_c, \dot{w_c}, \dot{d_c}, \dot{h_c}, \theta_z] \quad (8)$$

où les quantités marquées d'un point représentent les dérivées instantanées des quantités sans point correspondantes.

Listing 1: *Algorithme d'optimisation permettant d'améliorer la convergence vers un minimum global. Partant d'une estimation initiale x_0 l'algorithme réalise N essais de M itérations de l'algorithme de Nelder-Mead et renvoie la meilleure estimation x_{best}. Chacun de ces essais part d'une position aléatoire autour de la dernière meilleure estimation x_{best}.*

```
x_best = run_optimisation(x_0) {
  x_best = x_0;
  for(i = 1; i < N; ++i) {
    // initialise un simplexe X
    // aléatoirement autour
    // du point x_best
    X = init_simplex_around(x_best);

    // exécute M itérations
    // de Nelder-Mead
    for(j = 1; j < M; ++j)
      X = nelder_mead(X);

    // stocke le centre x du simplexe
    x = barycentre(X);

    // met à jour le meilleur état
    if (cost(x) < cost(x_best))
      x_best = x;
  }
  return x_best;
}
```

Id	Description condition de transition
c_1	l'objet est complètement dans la scène
c_2	l'objet n'est pas complètement dans la scène
c_3	l'objet n'a pas été associé à un blob
c_4	l'objet a disparu depuis trop longtemps ($> T_f$)
c_5	l'objet a été associé
c_6	l'objet entre dans un groupe
c_7	l'objet sort d'un groupe
c_8	l'objet a été associé durant la 1repasse
c_9	l'objet n'a pas été associé durant la 1repasse

Table 1: *Description des conditions de transition.*

2.5. Évolution d'un objet et Automate Fini

L'évolution d'un objet est modélisée à l'aide de l'automate fini représenté figure 5. La table 1 présente les significations de chacun des états et des conditions de transition.

Nous détaillons, à présent, les états considérés ainsi que les transitions possibles entre ces états :

NOUVEAU (s_0) : c'est l'état initial de tout objet d'intérêt. Cet état correspond au cas où l'objet vient d'être créé, mais n'est pas encore entièrement rentré dans la scène (par exemple sur un bord de la scène). Si l'objet entre entièrement dans la scène (c_1), il passe alors dans l'état SUIVI (s_1). En revanche, si l'objet disparait de la scène

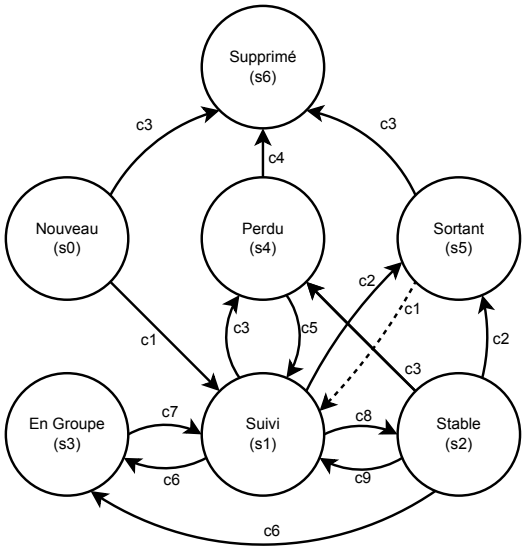


Figure 5: *Schématisation de l'automate fini utilisé. L'état initial est* NOUVEAU *(s_0), l'état final est* SUPPRIMÉ *(s_6). Les conditions de transition notées de c_1 à c_9 sont décrites dans le tableau 1. La transition marquée en pointillés a été ajoutée par rapport à l'article original.*

(c_3), il passe dans l'état SUPPRIMÉ (s_6). L'intérêt de l'état NOUVEAU (s_0) est d'une part de retenir provisoirement les objets qui sont en train de rentrer dans la scène, sans pour autant être totalement visible, et d'autre part, de permettre de filtrer les détections ponctuelles insignifiantes qui ne passeront jamais à l'état SUIVI (s_1).

SUIVI (s_1) : l'objet est entièrement dans la scène, mais sa stabilité n'est pas encore établie. Si l'objet est entré dans un groupe (c_6), celui-ci passe à l'état EN GROUPE (s_3). Si l'objet n'a pas été associé à un blob (c_3), celui-ci passe à l'état PERDU (s_4). Si l'objet n'est plus entièrement dans la scène (c_2), il passe à l'état SORTANT (s_5). Si l'objet est associé durant la première phase d'association (c_8), il passe à l'état STABLE (s_2).

STABLE (s_2) : l'objet est entièrement dans la scène et est considéré stable. Si l'objet est entré dans un groupe (c_6), celui-ci passe à l'état EN GROUPE (s_3). Si l'objet n'a pas été associé à un blob (c_3), celui-ci passe à l'état PERDU (s_4). Si l'objet n'est plus entièrement dans la scène (c_2), il passe à l'état SORTANT (s_5). Si l'objet n'est pas associé durant la première phase d'association (c_9), il passe à l'état SUIVI (s_1). L'utilité de l'état STABLE (s_2) est de traduire, pour les éventuels traitements de plus haut niveau que le suivi, un degré de confiance accru par rapport aux objets SUIVI (s_1).

EN GROUPE (s_3) : l'objet n'est plus suivi individuellement, seul le groupe auquel il appartient est suivi. Cet état permet de conserver les objets qui ne peuvent plus être suivi individuellement afin de permettre de réaffecter correctement ceux-ci lorsque le groupe se scindera (c_7). Dans ce cas, l'objet retourne à l'état SUIVI (s_1).

PERDU (s_4) : l'objet n'a pas été associé, par exemple à cause d'une occultation par un obstacle ou une mauvaise

détection. Cependant, son modèle est maintenu en mémoire, afin de permettre de le retrouver s'il venait à réapparaitre. Si l'objet n'est pas réapparu dans la scène depuis trop longtemps (c_4), c'est à dire, non associé consécutivement plus de T_f fois, il passe alors à SUPPRIMÉ (s_6). Si l'objet est retrouvé (c_5), il passe à l'état SUIVI (s_1).

SORTANT (s_5) : l'objet est en train de sortir de la scène (par exemple il touche le bord). Si l'objet n'est plus détecté (c_3), il passe à l'état SUPPRIMÉ (s_6). Si l'objet re-rentre dans la scène (c_1), il passe alors dans l'état SUIVI (s_1). À la différence de l'état PERDU (s_4), qui traduit la perte d'un objet généralement due à un problème de détection, l'état SORTANT (s_5) traduit l'intention d'un objet suivi de sortir de la scène. Il faut également le distinguer de l'état NOUVEAU (s_0), car bien qu'à la bordure de la scène considérée, l'objet SORTANT (s_5) est entré dans la scène et manifeste la volonté d'en sortir.

SUPPRIMÉ (s_6) : l'objet n'est plus suivi et peut être oublié. C'est l'état final de tout objet d'intérêt.

Si pour un état donné, aucune des conditions de transition ne s'applique, l'objet garde son état.

2.6. Intégration du contexte 3d

L'utilisation d'une caméra calibrée rend possible un ensemble de raisonnements tridimensionnels que nous allons détailler. À noter cependant, que comme seule une seule caméra est utilisée, il n'est pas possible de résoudre avec certitude l'ambiguïté de profondeur sans apport d'informations supplémentaires.

Les critères de transition c_1 et c_2 nécessitent de définir la zone correspondant à la scène surveillée, ainsi que le critère d'appartenance ou non à celle-ci. Étant donné que nous faisons l'hypothèse que les objets se déplacent sur le sol, il est naturel, de considérer qu'un objet est dans la scène quand sa boite englobante est incluse dans l'image, et que le bas de l'objet touche le sol. Cette définition permet d'éviter le suivi des éventuels blobs inintéressants se déplaçant dans le ciel par exemple. Pour générer un masque de sol automatiquement, nous utilisons l'approche proposée par [RTR13] qui permet de faire des rendus synthétiques de vues caméras, prenant en compte les bâtiments environnants grâce à une coordonnée GPS, la base de données OpenStreetMap et la calibration de la caméra. En choisissant les couleurs de rendu de manière appropriée (le sol en blanc, tout le reste en noir), nous obtenons ainsi un moyen simple et automatique d'obtenir un masque de sol prenant en compte les occultations liées aux bâtiments environnants (figure 6).

Par ailleurs, pour calculer les silhouettes des cuboïdes prédits nécessaires pour le calcul du critère de recouvrement s^o_{ij} (équation 4), nous réalisons un rendu synthétique des cuboïdes dans la scène tels qu'ils sont vus par la caméra, chaque cuboïde ayant une couleur unique. Ce rendu tient compte des occultations inter-objets et de celles liées à la scène.

Finalement nous attirons l'attention sur le fait que cette formulation tridimensionnelle a été mise en place dans le but de pouvoir également filtrer les ombres projetées des objets. Cependant, par manque de temps, ce filtrage n'a pas été mis en place. L'idée est d'intégrer le modèle de prédiction d'ombres projetées développé par [RTR13] dans la phase d'estimation du cuboïde via l'ajout d'une variable booléenne présence/absence de l'ombre dans la fonction de rendu. L'optimisation de l'énergie sera alors capable de déduire si la présence de l'ombre est compatible avec le masque de mouvement, et dans ce cas la filtrer.

3. Évaluation

3.1. Éléments d'implémentation

Nous avons implémenté en C++ l'algorithme décrit dans [LFP*13]. Cette implémentation, que nous espérons fidèle à la méthode originale, constitue notre référence, et nous permet d'évaluer l'apport de nos modifications pour le rendre pleinement utilisable dans le contexte de la vidéosurveillance.

Les masques de mouvement sont obtenus à l'aide du modèle de fond ViBe [BVD11], suivi d'opérations morphologiques filtrant les pixels isolés.

Le seuil d'association est fixé à 0.66 pour la méthode originale, et pour notre méthode, nous utilisons les valeurs fixes suivantes : $T_1 = 0.9, T_2 = T_g = 0.66, T_3 = 0.75$. De la même manière que pour la méthode originale, nous fixons T_f de manière à oublier les objets perdus depuis plus de 1 seconde.

3.2. Validation qualitative

Afin de valider l'approche proposée, nous avons réalisé un ensemble de vidéos synthétiques courtes à l'aide du logiciel de création et d'animation 3d Muvizu [†]. Chaque séquence synthétique a été réalisée afin de tester un aspect précis d'un algorithme de suivi. La première vidéo montre deux piétons marchant vers la caméra ; à mi-chemin l'un passe devant l'autre, puis avant d'arriver au pied de la caméra, ils repartent chacun de leur côté. La deuxième vidéo montre le croisement d'un piéton et d'une voiture ; le piéton vient de la gauche, la voiture de la droite. Nous avons également utilisé la vidéo de test de la séquence 1 du challenge PETS 2001 [‡], qui met en scène des piétons et des voitures. Nous présentons dans ce qui suit quelques exemples de résultats montrant les apports de notre méthode. Une validation qualitative sur un autre jeu de données (pour lesquelles nous disposons de la vérité terrain) est présenté dans la section 3.3.

La première séquence (lignes 1 et 2 de la figure 7) montre la difficulté qu'a l'algorithme original (ligne 2) à détecter les débuts d'occultation. On note cependant que grâce à l'état PERDU, l'algorithme retrouve et réassocie correctement le piéton qui a été masqué totalement. En revanche, pour notre version (ligne 1), la détection du début d'occulation, la création du groupe associé et la fin d'occultation ont été correctement réalisées.

La deuxième séquence (lignes 3 et 4 de la figure 7) montre un cas où l'algorithme original (ligne 4) n'arrive pas à détecter le début d'occultation (à cause de la différence de taille

†. http ://muvizu.com
‡. http ://www.cvg.reading.ac.uk/PETS2001

Figure 6: *Exemple de génération du masque de sol utilisant le framework de rendu de [RTR13] : (a) scène originale, (b) carte sémantique synthétique (sol en rouge, ciel en noir, bâtiments en vert), (c) masque de sol*

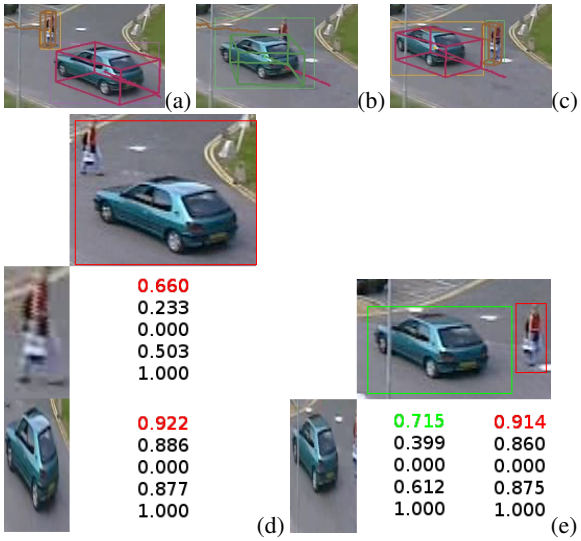


Figure 8: *Cas d'occultation d'un piéton par une voiture (a,b,c) et matrices d'associations ayant permis la détection du début (d) et de la fin (e) d'occultation. La fusion du blob du piéton et de la voiture est détectée (d), un groupe est alors créé et associé au blob regroupant le piéton et la voiture (b). À la scission de ce blob (e), les modèles individuels du piéton de la voiture sont correctement réassociés (c).*

des objets) et cause donc une perte, puis la création d'un nouvel objet. Notre algorithme (ligne 3), quant à lui, détecte correctement le début et la fin de l'occultation, permettant ainsi de suivre correctement la voiture et le piéton.

La séquence de test 1 du challenge PETS 2001 confirme les observations faites sur les séquences synthétiques : notre algorithme améliore la détection de début de fin d'occultation. Une illustration plus précise d'occultation entre un piéton et une voiture est donné à la figure 8.

3.3. Validation quantitative

Pour l'évaluation quantitative, nous avons utilisé deux vidéos réelles : PETS09.S2L1, qui fait partie du challenge PETS [§] ; et une vidéo (parking1) capturée sur un parking proche de nos locaux. Les caractéristiques principales des

Vidéo	Résolution	# Img	Img/s	# objets
PETS.S2L1	768x576	795	7	19
parking1	1280x960	3600	25	27

Table 2: *Caractéristiques des séquences vidéos de test*

vidéos (résolution, nombre d'images, nombre d'images par seconde et nombre d'objets) utilisées pour la validation sont répertoriées dans le tableau 2. Pour la vidéo PETS09.S2L1, les données de calibrations sont fournies et ont donc été employées. Les coordonnées GPS permettant de retrouver les bâtiments environnants sont également utilisées. Pour la vidéo (parking1), nous avons réalisé une calibration sommaire à l'aide d'un logiciel interne de simulation et de placement de caméras.

Pour la vidéo PETS.S2L1, nous utilisons la vérité terrain mise à disposition par A. Milan [¶]. Pour la vidéo parking1, nous avons étiqueté manuellement la séquence.

Nous utilisons les métriques d'évaluation CLEARMOT décritent dans [BS08], en particulier les mesures : *Multiple Object Tracking Accuracy* (MOTA), *Multiple Object Tracking Precision* (MOTP), *False Positive* (FP), *False Negative* (FN), et *ID Switch* (IDs). Nous classons également les trajectoires en terme de *Mostly Tracked* (MT), *Partially Tracked* (PT) et *Mostly Lost* (ML) selon le pourcentage de la trajectoire effectivement suivi [LHN09] : si 80% ou plus de la trajectoire d'un objet est correctement suivie, la trajectoire est considérée MT, si moins de 20% de la trajectoire d'un objet est correctement suivie, la trajectoire est considérée ML, sinon elle est considérée PT.

Avant toute remarque, nous souhaitons attirer l'attention sur la difficulté d'évaluer objectivement un algorithme de suivi étant donné la dépendance à la qualité des détections, la qualité de la vérité terrain, ou l'implémentation de l'algorithme d'évaluation [MSR13]. Nous précisons également que nous exportons les groupes et non les objets le composant en cas d'occultations ; ainsi une certaine baisse de performance est observée car la vérité terrain considère les objets séparément et non le groupe.

L'algorithme proposé améliore sensiblement l'algorithme initial (table 3), et semble mieux gérer les cas de début

§. http ://www.cvg.rdg.ac.uk/PETS2009/

¶. `http://www.milanton.de/data.html`

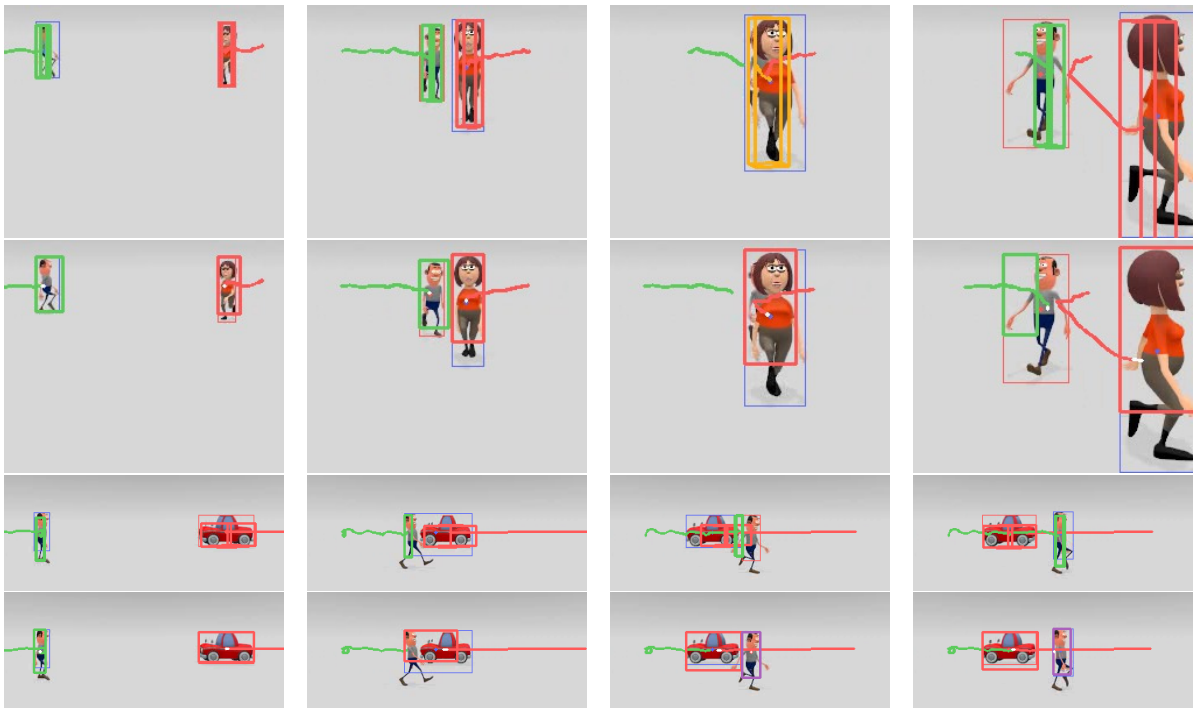


Figure 7: *Illustration sur séquences synthétiques, et comportement des algorithmes sur deux cas d'occultations (piéton/piéton et piéton/voiture). La première (resp. deuxième) ligne montre le comportement de notre algorithme (resp. l'algorithme original) sur la première séquence synthétique (cas piéton/piéton). La troisième (resp. quatrième) ligne montre le comportement de notre algorithme (resp. l'algorithme original) sur la deuxième séquence synthétique (cas piéton/voiture).*

Video	Methode	MOTA	MOTP	GT	FP	FN	IDs	Précision	Recall	MT	PT	ML
PETS.S2L1	[LFP*13]	0.759	0.594	4644	176	762	179	0.957	0.836	13	6	0
	Notre Méthode	0.774	0.617	4644	114	737	199	0.972	0.841	14	5	0
parking1	[LFP*13]	0.705	0.510	15439	226	4196	135	0.980	0.728	12	15	0
	Notre Méthode	0.729	0.516	15439	200	3754	228	0.983	0.757	12	15	0

Table 3: *Performance des algorithmes de suivi*

et fin d'occultation (figure 9). Quasiment tous les indicateurs montrent une amélioration, à l'exception du nombre d'échanges d'identité (IDs). Une des causes expliquant le nombre plus important d'échanges d'identité concerne la destruction des groupes : dans les cas où notre algorithme détecte la séparation d'un groupe alors que l'algorithme original ne le détecte pas (par exemple à la figure 9), la réaffectation d'identité à la destruction du groupe causera potentiellement un échange d'identité supplémentaire (ID groupe vers ID objet), que l'algorithme original ne fera pas. Par ailleurs, on notera la baisse du nombre de faux positifs (FP) et du nombre de faux négatifs (FN) qui sont appréciables dans un contexte de vidéosurveillance : les omissions (faux négatifs) ne sont pas tolérables car elles sont contraires aux objectifs d'un système de détection d'intrusion ; alors que les fausses alarmes (faux positifs) tendent à réduire la confiance des opérateurs en leur système de surveillance.

Il est à noter que notre méthode n'est pas encore capable d'améliorer les cas où les objets allant former un groupe sont de taille très inégale (figure 10) : dans ce cas, l'objet de dimension prépondérante risque d'être associé dès la première

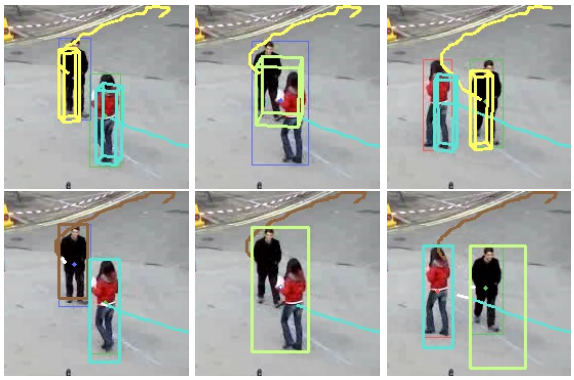


Figure 9: *Exemple de création et destruction de groupe mieux géré par notre approche : dans notre cas, le groupe est créé puis détruit correctement. Alors que dans l'algorithme original, un nouvel objet est créé, et il vole l'identité d'un des piétons.*

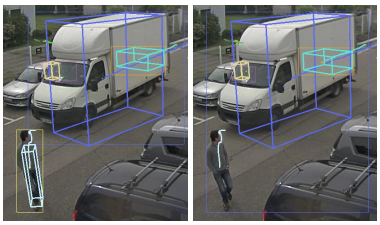


Figure 10: *Cas où les objets sont de dimensions trop inégales, et où la création de groupe n'a pas pu être détectée : Le piéton est "absorbé" par le camion.*

passe (le petit objet est en fait négligeable), ce qui empêche la création du groupe.

Par ailleurs, l'estimation des cuboïdes des objets suivis est délicate et requière un masque de mouvement peu bruité où les pieds/points de contact au sol soient visibles. De plus elle ralentit l'algorithme du fait du nombre d'itérations et de la formulation de l'énergie ; cependant, cela autorise un raisonnement en 3d que nous souhaitons utiliser afin de détecter et filtrer les ombres présentes sur le masque de mouvement (l'estimation des ombres projetées peut être réalisée comme indiqué dans [RTR13]).

Notons que l'utilisation de la 3d (bâtiments environnants, masque de sol, estimation des cuboïdes) nécessite une caméra calibrée, cependant, pour une application en vidéosurveillance, cette calibration, bien que grossière, peu facilement être obtenue à partir des études de prédéploiement des caméras, ce qui n'est donc pas un frein à son utilisation. Nous l'avons montré sur les vidéos synthétiques et parking1 où une calibration grossière a été effectuée à l'aide du logiciel interne de simulation et de placement de caméras, et non par une procédure de calibration classique.

4. Conclusion

Nous avons proposé un algorithme de suivi multi-objets adapté au contexte de la vidéosurveillance. Celui-ci étend les travaux de [LFP*13] en généralisant le suivi, en améliorant les mécanismes de formation et destruction des groupes, et en intégrant la 3d dans les raisonnements. Nous avons montré l'impact de notre contribution par rapport à l'algorithme initial à la fois sur des exemples synthétiques et sur des exemples réels. En terme de travaux futurs, nous souhaitons étendre le raisonnement 3d afin de prédire et filtrer les éventuelles ombres présentent dans le masque de mouvement.

Références

[BRL*09] BREITENSTEIN M., REICHLIN F., LEIBE B., KOLLER-MEIER E., VAN GOOL L. : Robust tracking-by-detection using a detector confidence particle filter. In *Computer Vision, 2009 IEEE 12th International Conference on* (Sept 2009), pp. 1515–1522.

[BS08] BERNARDIN K., STIEFELHAGEN R. : Evaluating multiple object tracking performance : The clear mot metrics. *J. Image Video Process.. Vol. 2008* (janvier 2008), 1 :1–1 :10.

[BVD11] BARNICH O., VAN DROOGENBROECK M. : Vibe : A universal background subtraction algorithm for video sequences. *Image Processing, IEEE Transactions on. Vol. 20*, Num. 6 (2011), 1709–1724.

[CFPV10] CONTE D., FOGGIA P., PERCANNELLA G., VENTO M. : Performance evaluation of a people tracking system on pets2009 database. In *Proceedings of the 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance* (Washington, DC, USA, 2010), AVSS '10, IEEE Computer Society, pp. 119–126.

[FJ14] FÜHR G., JUNG C. R. : Combining patch matching and detection for robust pedestrian tracking in monocular calibrated cameras. *Pattern Recognition Letters. Vol. 39*, Num. 0 (2014), 11 – 20. Advances in Pattern Recognition and Computer Vision.

[GMG12] GODBEHERE A., MATSUKAWA A., GOLDBERG K. : Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. In *American Control Conference (ACC), 2012* (June 2012), pp. 4305–4312.

[LFP*13] LASCIO R. D., FOGGIA P., PERCANNELLA G., SAGGESE A., VENTO M. : A real time algorithm for people tracking using contextual reasoning. *Computer Vision and Image Understanding. Vol. 117*, Num. 8 (2013), 892–908.

[LFSV12] LASCIO R. D., FOGGIA P., SAGGESE A., VENTO M. : Tracking interacting objects in complex situations by using contextual reasoning. In *VISAPP (2)* (2012), Csurka G., Braz J., (Eds.), SciTePress, pp. 104–113.

[LHN09] LI Y., HUANG C., NEVATIA R. : Learning to associate : Hybridboosted multi-target tracker for crowded scene. In *CVPR'09* (2009), pp. 2953–2960.

[MSR13] MILAN A., SCHINDLER K., ROTH S. : Challenges of ground truth evaluation of multi-target tracking. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on* (June 2013), pp. 735–742.

[NM65] NELDER J. A., MEAD R. : A simplex method for function minimization. *The Computer Journal. Vol. 7*, Num. 4 (1965), 308–313.

[PA10] PAPADOURAKIS V., ARGYROS A. : Multiple objects tracking in the presence of long-term occlusions. *Comput. Vis. Image Underst.. Vol. 114*, Num. 7 (juillet 2010), 835–846.

[RKDL12] RAHEJA J. L., KALITA S., DUTTA P. J., LOVENDRA S. : A robust real time people tracking and counting incorporating shadow detection and removal. *International Journal of Computer Applications. Vol. 46*, Num. 4 (May 2012), 51–58. Published by Foundation of Computer Science, New York, USA.

[RTR13] ROGEZ M., TOUGNE L., ROBINAULT L. : A Prior-Knowledge Based Casted Shadows Prediction Model Featuring OpenStreetMap Data. In *VISAPP* (février 2013), pp. 602–607.