POLYTECHNIQUE MONTRÉAL

INF6804 - Vision par ordinateur

TP3 - Détection et suivi d'un objet d'intérêt

Travail présenté à

Guillaume-Alexandre Bilodeau
Soufiane Lamghari

par
Marc-Olivier Bélanger - 1840713
Pierre-Luc Chartier - 1805679

Polytechnique de Montréal
19 avril 2021

1 Description de la méthode choisie

Notre algorithme se base sur les trois étapes de suivi multi-objets vues en classe, soit la *détection*, la *description* et l'*association*. L'objectif était de créer un algorithme réutilisable d'un ensemble de données à l'autre. Pour se faire, avant de lancer l'algorithme, nous créons un ensemble qui contient tous les objets d'intérêt que nous voulons suivre au cours de la séquence. Nous avons créé une classe *TrackedObject* pour représenter un objet d'intérêt. Ses attributs sont les coordonnées du rectangle englobant l'objet, l'histogramme de couleurs courant de l'objet ainsi qu'une variable booléenne *was_ detected* afin de différencier les objets d'intérêt ayant été détectés dans une trame de ceux qui ne l'ont pas été. Les objets *TrackedObject* sont initialisés avec les coordonnées du rectangle englobant provenant de la vérité de terrain de la première trame où ils apparaissent dans la séquence ainsi qu'avec leur histogramme de couleurs par rapport à cette même trame. La valeur de l'attribut *was_ detected* est initialisée à **Faux**.

Après avoir créé l'ensemble des objets d'intérêt *tracked_ objects*, nous procédons à la boucle principale de l'algorithme. Dans cette boucle, nous parcourons les trames en ordre chronologique. Nous commençons d'abord par détecter les objets à l'aide du modèle *Mask R-CNN* pré-entraîné disponible dans la librairie *torchvision*. Ce réseau convolutif utilise l'architecture *ResNet50 FPN* comme *backbone*. Nous extrayons les coordonnées des rectangles englobants des objets détectés par le réseau pour ensuite calculer leur histogramme de couleurs et ainsi obtenir leur description. Nous utilisons ces histogrammes de couleurs afin de procéder à l'association des données. On compare les histogrammes de couleurs des objets détectés à ceux de nos objets d'intérêt et appliquons l'algorithme hongrois afin de trouver la combinaison qui minimise le coût d'association.

C'est à ce moment que le coeur de notre algorithme de suivi entre en jeu. Pour chaque association retournée par l'algorithme hongrois, on calcule l'intersection sur l'union du rectangle englobant de l'objet d'intérêt avec celui provenant de la prédiction associée du modèle. Si le résultat est supérieur à 50%, on met à jour l'objet d'intérêt, i.e. les coordonnées de son rectangle englobant, son histogramme de couleurs et changeons la valeur de son attribut *was_ detected* à **Vrai**. Nous considérons cette détection comme **valide**, c'est pour cette raison que nous mettons à jour la description de l'objet.

Il est très important de mentionner ici que l'algorithme hongrois fait l'association uniquement avec les objets détectés par le modèle. Par conséquent, pour une trame où un objet n'est pas reconnu par le modèle ou s'il est occlus, il est impossible de pouvoir le ré-identifier directement à partir des prédictions du modèle. Nous le considérons donc comme non-détecté et laissons la valeur de son attribut *was_ detected* à **Faux**. Pour tous les objets n'ayant pas été détectés, nous appliquons la méthode *Mean Shift* afin de tenter de les ré-identifier. Nous mettons à jour les coordonnées du rectangle englobant d'un objet d'intérêt non-détecté uniquement si l'intersection sur l'union de son rectangle englobant avec celui de la prédiction de *Mean Shift* est supérieure à 50%. Nous ne mettons pas à jour sa description dans cette situation afin de conserver celle provenant de sa

dernière détection **valide** par le modèle.

FIGURE 1 – (À gauche) Trame 124 : la tasse 0 est détectée par le modèle. (À droite) Trame 125 : la tasse 0 n'est pas détectée par le modèle. Nous appliquons *Mean Shift* sans mettre sa description à jour.

En résumé, si nous sommes confiants qu'un objet est détecté par notre modèle, basé sur l'intersection sur l'union avec sa prédiction associée par l'algorithme hongrois, nous mettons à jour les coordonnées de son rectangle englobant et son histogramme de couleurs. Si ce n'est pas le cas, nous mettons uniquement les coordonnées de son rectangle englobant à jour si la prédiction de *Mean Shift* a une intersection sur l'union supérieure à 50%.

Une des forces de notre méthode est qu'elle n'identifie jamais une tasse incorrectement. Ceci s'explique par notre choix de mise à jour de la description des objets d'intérêt. Ainsi, peu importe la tasse, si son histogramme de couleurs ne correspond pas à ceux de nos objets d'intérêt, on ne l'identifiera pas comme telle. Cette mise à jour de la description des objets d'intérêt est cependant grandement affectée par une des faiblesses de notre méthode, soit sa capacité à évaluer la qualité d'une détection. Advenant le cas où un objet n'est pas détecté pendant plusieurs trames et que toutes les prédictions de *Mean Shift* sont trop éloignées, les coordonnées de son rectangle englobant ne sont simplement pas mises à jour. Lorsque l'objet est détecté de nouveau par le modèle, nous sommes limités par le résultat de l'intersection sur l'union. Ceci fait en sorte que si l'objet est redétecté trop loin de sa dernière détection valide, notre méthode effectuera un faux négatif et le considérera comme non-détecté. Même avec l'application de *Mean Shift*, nous nous retrouvons limité par l'intersection sur l'union. Si l'objet repasse suffisamment près de l'endroit où il a disparu, notre méthode sera par contre en mesure de le ré-identifier correctement et de mettre à jour les coordonnées de son rectangle englobant et sa description.

2 Identification des difficultés dans la séquence fournie

Notre méthode performe bien pour les 100 premières trames. Il faut cependant mentionner qu'elles ne contiennent aucune difficulté et ne mettent pas la puissance de détection de notre modèle à l'épreuve. Les premières difficultés de la séquence apparaissent à la trame 125 où il y a de l'occlusion sur une tasse. Les trames qui suivent mettent le modèle à l'épreuve puisqu'on applique une rotation dans le plan à cette tasse occluse. Cette portion est donc plus difficile à traiter puisqu'il faudrait mettre la description de la tasse à jour.

On peut d'ailleurs observer une dérive du modèle résultant de ce manque de mise à jour de la description à partir de la trame 155. Bien que les prédictions du modèle soient probablement exactes, elles ne sont pas suffisamment près des anciennes positions des tasses. La performance de notre méthode est très affectée par la limitation de l'intersection sur l'union avec la prédiction du modèle tel que mentionné plus haut. Les trames 200 à 250 contiennent plus de rotations hors du plan et d'occlusions et notre méthode n'y performe pas très bien.

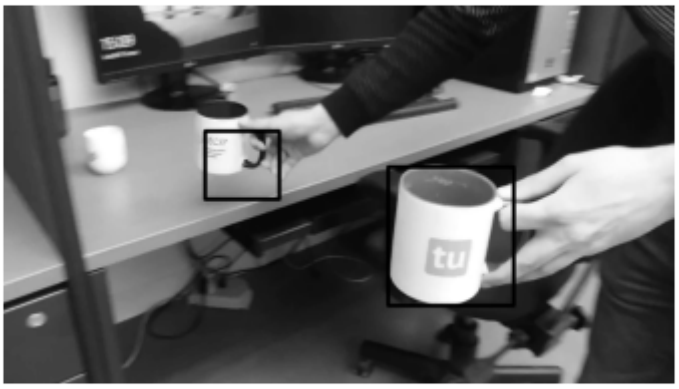

FIGURE 2 – Trame 155 : Dérive du modèle suite à une occlusion et un manque de détection par le modèle.

FIGURE 3 – Trame 200 : (Tasse de gauche) Suite de la dérive du modèle.
(Tasse de droite) Rotation hors du plan et éloignement considérable de l'ancien rectangle englobant

Une autre difficulté de la séquence se manifeste lorsque les autres tasses apparaissent en arrière-plan. Par contre, tel que mentionné plus haut, notre méthode arrive à distinguer ces tasses de nos objets d'intérêt pour toutes les trames restantes. Les autres difficultés qui se manifestent pour la suite de la séquence sont d'autres occlusions, des mises à l'échelle et quelques rotations hors du plan. Plusieurs occlussions durent assez longtemps pour dérouter notre méthode.

FIGURE 4 – Trame 291 : Les tasses en arrière-plan apparaissent de façon nette mais n'influencent pas les prédictions de la méthode

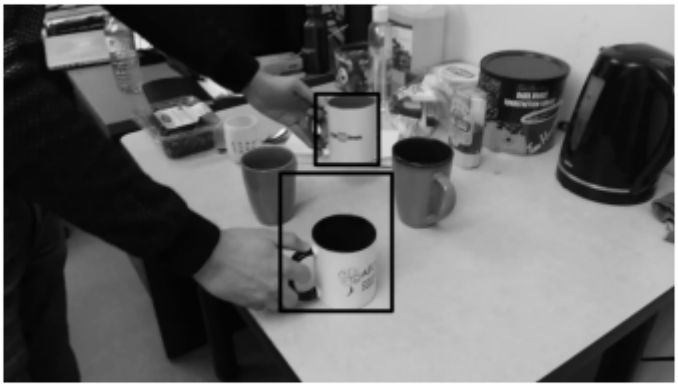

FIGURE 5 – Trame 510 : La méthode n'est pas influencée par les nouvelles tasses. Le modèle retrouve quelque peu ses repères.

FIGURE 6 – Trame 680 : La mise à l'échelle de la tasse de droite déstabilise la méthode.

3 Justification de la méthode

L'objectif principal de notre méthode était d'être simple. Nous ne savions pas exactement à quoi nous attendre par rapport à la puissance de détection du modèle *Mask R-CNN*, mais nous nous attendions à ce qu'il performe mieux pour les rotations hors du plan et les mises à l'échelle. Certaines détections étaient également

assez évidentes à l'oeil humain et nous pensions que le modèle aurait été en mesure d'en faire la prédiction. Quant aux occlusions, ce sont des problèmes qui restent difficiles à traiter puisque notre gestion des trajectoires n'est pas très bonne, autant pour leur naissance que pour leur mort. Nous avons essayé plusieurs approches pour prédire la trajectoire des objets non-détectés afin de les ré-identifier, mais l'approche *Mean Shift* avait l'avantage d'être très rapide et donnait de meilleurs résultats. Un des problèmes que notre méthode résout facilement est la discrimination des objets, i.e. la distinction entre les objets que nous voulons suivre et ceux qui ne sont pas d'intérêt. Tel que mentionné précédemment, ceci s'explique par notre condition de mise à jour de la description de l'objet.

4 Description de l'implémentation

L'implémentation détaillée de notre méthode se trouve à la section 1. Nous avons écrit tout le code nous-mêmes à partir de notre compréhension de la matière du cours, principalement celle du chapitre 4. Nous n'avons pas entraîné un détecteur d'objets nous-mêmes par souci de simplicité. Le modèle *Mask R-CNN* pré-entraîné n'a pas demandé de modifications et nous nous sommes inspirés des exemples du cours pour l'utiliser. Ce modèle prend la matrice des pixels d'une image comme paramètre et retourne des *tensors* qui nous permettent d'extraire les informations qui nous intéressent. La matrice des pixels d'une image est obtenue à l'aide de la librairie *OpenCV*. La seule méthode provenant d'une source externe est celle pour calculer l'intersection entre deux rectangles, tirée ici. C'est une méthode fort simple qui prend les coordonnées de deux rectangles comme paramètres et retourne l'aire de leur intersection. Nous nous en sommes servis pour calculer l'intersection sur l'union de deux rectangles.

Nous avons également utilisé la librairie *MOTMetrics* afin de calculer nos précisions sur les ensembles de données **MOT17**. Les métriques calculées par le module utilisent un objet *MOTAccumulator* qu'on met à jour à chaque trame. Cette mise à jour prend en paramètre les objets de la vérité de terrain présent dans la trame, les prédictions associées de notre méthode et la matrice des distances entre ces-derniers. Ces distances sont calculées à l'aide de la distance de Bhattacharyya appliquée à la comparaison des histogrammes de couleurs des objets d'intérêt et des rectangles englobants de nos prédictions.

5 Présentation des résultats de validation

Nous avons choisi trois ensembles de données du défi **MOT17** afin de valider notre méthode pendant son développement. Le premier ensemble que nous avons utilisé est MOT17-13-FRCNN *Filmed from a bus on a busy intersection*. Cet ensemble de données est un bon barème pour notre modèle puisqu'il y a plusieurs objets de différentes classes à détecter en même temps et que la caméra est en mouvement tout comme la séquence des tasses. Étant donné cette prise de vue mobile, il y a nécessairement des rotations hors du plan, par exemple

lorsque l'autobus tourne à l'intersection, et de multiples objets qui apparaissent et disparaissent du champ de vision. Les figures 7, 8 et 9 présentent des exemples de détection de notre méthode comparés à la vérité de terrain. Nous pouvons observer que le modèle *Mask R-CNN* détecte à la base beaucoup moins d'objets que ceux présents dans la vérité de terrain. Le modèle a également une performance médiocre lors d'occlusions, comme on peut le voir aux figures 8 et 9 avec la voiture qui cache les piétons ou la poubelle qui obstrue la moitié inférieure d'un passant.

FIGURE 7 – (À gauche) Vérité de terrain (À droite) Détections de *Mask R-CNN*

FIGURE 8 – (À gauche) Vérité de terrain (À droite) Détections de *Mask R-CNN*

FIGURE 9 – (À gauche) Vérité de terrain (À droite) Détections de *Mask R-CNN*

Le deuxième ensemble que nous avons utilisé est MOT17-11-FRCNN *Forward moving camera in a busy shopping mall*. Cet ensemble de données est également une bonne mesure de notre méthode puisque, tout comme dans la séquence des tasses, la prise de vue est mobile et plutôt instable (*jittering*). Comparativement à MOT17-13-FRCNN, cette prise de vue est frontale, il n'y a donc pas de rotations hors du plan ou même dans le plan. Il y a également beaucoup de personnes à détecter en même temps et des objets apparaissent et disparaissent constamment du champ de vision. Les figures 10 et 11 présentent des exemples de détection

de notre méthode comparés à la vérité de terrain. À la figure 10, nous pouvons observer que l'instabilité de la caméra semble faire en sorte que les personnes dans la vitrine de gauche ne sont pas détectées. À la figure 11, le modèle ne détecte pas l'homme à la casquette blanche obstrué par les deux hommes aux vêtements bleus à l'avant-plan.

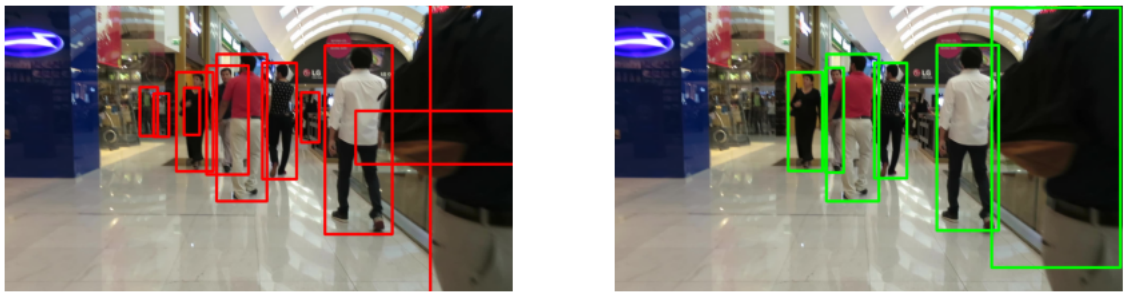

FIGURE 10 – (À gauche) Vérité de terrain (À droite) Détections de *Mask R-CNN*

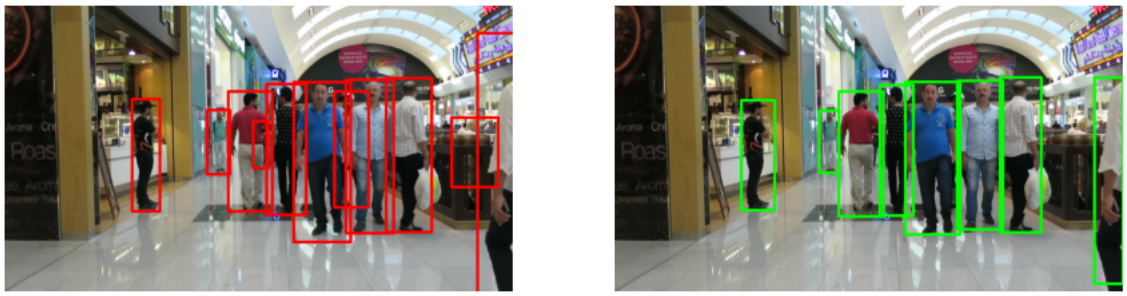

FIGURE 11 – (À gauche) Vérité de terrain (À droite) Détections de *Mask R-CNN*

Finalement, le dernier ensemble de validation que nous avons utilisé est MOT17-05-FRCNN *Street scene from a moving platform*. Cet ensemble de données contient une séquence vidéo du point de vue d'un piéton traversant une intersection bondée. En termes de difficultés, c'est une séquence similaire à MOT17-11-FRCNN au niveau de la prise de vue mobile, frontale et instable. Il y a cependant un nombre d'occlusions beaucoup plus élevé. Il faut souligner que la caméra présente aussi des aberrations sphériques qui rendent les bordures de l'image légèrement floues. La séquence vidéo présente une séquence où la caméra est statique et une séquence où la caméra est mobile. Les figures 12, 13 et 14 présentent des exemples de détection de notre méthode comparés à la vérité de terrain. Nous pouvons observer que notre méthode a beaucoup de difficultés avec les occlusions.

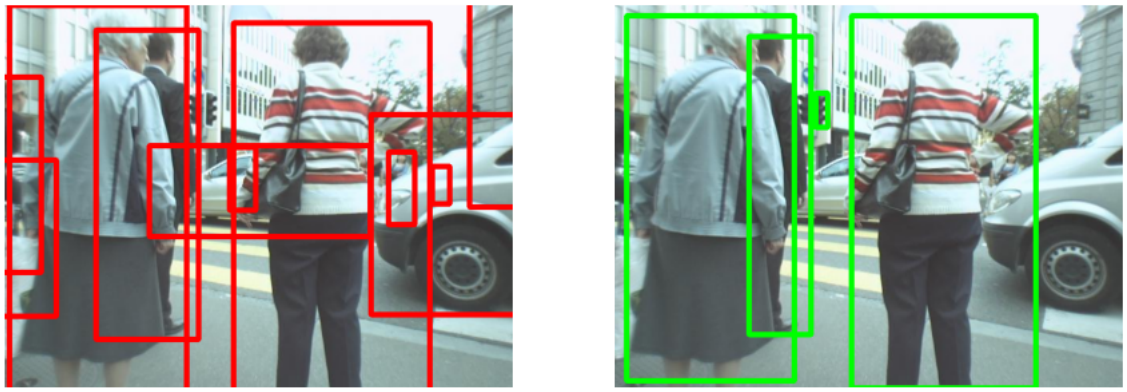

FIGURE 12 – (À gauche) Vérité de terrain (À droite) Détections de *Mask R-CNN*. Les deux voitures ne sont pas détectées.

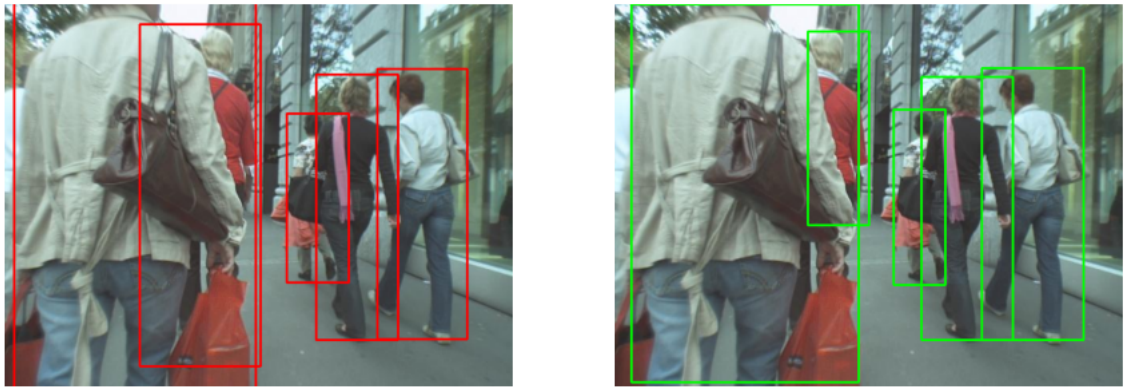

FIGURE 13 – Occlusions faibles de la dame en rouge et de celle au chandail à motifs

FIGURE 14 – Occlusions fortes de la dame en rouge et de celle au chandail à motifs

Les résultats de nos suivis multi-objets pour les trois séquences vidéos sont colligés ci-dessous.

Ensemble de données	**MOTA (%)**	**MOTP (%)**
MOT17-13 Intersection	0,312246	0,419061
MOT17-11 Shopping Mall	0,543185	0,356561
MOT17-05 Street Scene	0,469986	0,300729

TABLE 1 – Résultats des tests d'évaluation de performance de notre méthode avec les métriques MOTA et MOTP pour trois ensembles de données du défi **MOT17**

6 Discussion des résultats

Bien que nous ayons un peu discuté des résultats à la section précédente, nous allons présenter plus en détails les métriques de précision afin de bien comprendre les résultats de la performance de notre méthode. Les métriques d'évaluation utilisées sont le *Multiple Object Tracking Precision* (MOTP) et le *Multiple Object Tracking Accuracy* (MOTA). Les métriques sont calculées par les formules suivantes :

$$MOTP = \frac{\sum_{i,t} d_{i,t}}{\sum_t c_t} \tag{1}$$

où c_t est le nombre de détections et $d_{i,t}$ la distance de la détection par rapport à la vérité terrain à la trame t et pour l'objet i.

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \tag{2}$$

où m_t est le nombre d'objets à détecter manqués à la trame t , fp_t est le nombre de détections faux-positif ,mme_t le nombre de fausses correspondances et g_t le nombre d'objets à détecter présents. Ainsi, le MOTP permet de quantifier la capacité de notre méthode à correctement placer les rectangles englobants autour de l'objet détecté. Le MOTA quant à lui prend en compte toutes les erreurs de prédiction faites par le modèle.

En général, le MOTP de nos simulations est faible et ne dépasse pas les 42%. Tel que mentionné dans la section présentant notre méthode, nous avons de la difficulté à gérer la situation lorsque les objets sont obstrués par des occlusions. Si l'occlusion fait en sorte que le modèle *Mask R-CNN* propose un rectangle englobant avec une valeur d'intersection sur l'union de moins de 50%, la méthode *Mean Shift* est utilisée, ce qui est moins précis qu'une méthode utilisant des réseaux de neurones. De plus, notre méthode est très sensible à la dérive de modèle. Cette cause d'erreur explique pourquoi la séquence vidéo MOT17-05-FRCNN *Street Scene* est la séquence vidéo avec la moins bonne MOTP. C'est dans cette séquence vidéo que les objets identifiés restent le plus longtemps dans le champ de vision de la caméra puisque le point de vue bouge dans la même direction que la majorité des piétons et qu'il y a une séquence statique. Les erreurs induites par les occlusions et la dérive des modèles des objets s'accumulent plus longtemps pour cet ensemble de données. Inversement, la séquence avec l'autobus est celle avec la plus grande MOTP puisque les objets ne restent pas longtemps dans le plan de la caméra.

La métrique du MOTA est aussi généralement faible pour nos simulations. Une grande partie de ces résultats peut être expliquée par le fait que le modèle *Mask R-CNN* utilise l'architecture *ResNet50 FPN* entraîné sur 80 catégories d'objets pour faire la classification. Parmi ces catégories, on retrouve les personnes, les vélos, les véhicules, les camions, les feux de circulations et les autobus. Certains objets identifiés dans la vérité de terrain ne font pas partie des catégories d'objets avec lesquelles le réseau de neurones a été entraîné, par exemple des poubelles ou les poteaux des feux de circulation, comme nous pouvons l'observer aux figures 8 et 9. La métrique MOTA de nos tests d'évaluation est donc faible puisque le nombre d'objets à détecter manqués est grand. La valeur la plus élevée obtenue est avec la séquence dans le centre commercial. Il s'agit de la séquence avec le

moins d'objets qui ne sont pas reconnus par la méthode *Mask R-CNN* et comporte en grande majorité des personnes à identifier.

7 Conclusion

Au cours de ce travail, nous avons présenté notre méthode de suivi multi-objets, identifier les difficultés de la trame fournie afin de justifier notre approche en plus de présenter nos résultats préliminaires sur trois ensembles du défi **MOT17**. Bien que la performance de notre méthode ne soit pas satisfaisante, nous avons beaucoup appris sur le suivi d'objets. L'étape d'association nous a posé un défi considérable et nous a fait constater l'ampleur du problème en suivi multi-objets. Au terme de notre analyse, il serait très intéressant de voir comment les modèles de l'état de l'art procèdent pour répondre à cette problématique.