Titre: Title:	Suivi multi-objets : au-delà des vecteurs de réidentification pour l'association de données
Auteur: Author:	Mehdi Naim Miah
Date:	2024
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Miah, M. N. (2024). Suivi multi-objets : au-delà des vecteurs de réidentification pour l'association de données [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/59164/

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Suivi multi-objets : au-delà des vecteurs de réidentification pour l'association de données

MEHDI NAIM MIAH
Département de génie informatique et génie logiciel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie informatique

Août 2024

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Suivi multi-objets : au-delà des vecteurs de réidentification pour l'association de données

présentée par **Mehdi Naim MIAH**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Daniel ALOISE, président
Guillaume-Alexandre BILODEAU, membre et directeur de recherche
Nicolas SAUNIER, membre et codirecteur de recherche
Lama SÉOUD, membre
Philippe GIGUÈRE, membre externe

DÉDICACE

À la communauté scientifique qui partage librement la connaissance et les ressources.

REMERCIEMENTS

Je tiens à exprimer ma profonde gratitude à mon directeur Guillaume-Alexandre Bilodeau pour son soutien, ses conseils et sa bienveillance. Je remercie également mon codirecteur Nicolas Saunier pour ses encouragements constants et ses précieuses révisions. Leur encadrement tout au long de mon doctorat a été inestimable.

Je remercie chaleureusement le professeur Bruno Agard et le laboratoire LID pour m'avoir encouragé à poursuivre un doctorat et m'avoir guidé vers mes directeurs de recherche. Leur accueil et leur soutien ont été déterminants pour mon intégration à Montréal.

Je tiens à exprimer toute ma gratitude à Yann pour son soutien inébranlable et les innombrables heures passées à m'écouter discuter de mes théories, bonnes comme mauvaises. Merci également pour m'avoir fait découvrir la province du Québec. Je remercie également la famille Trottier pour leur accueil chaleureux et leur hospitalité. Je souhaite également remercier Zahra pour les discussions si variées et si enrichissantes que nous avons eues.

J'exprime également ma profonde gratitude à ma famille pour leur soutien indéfectible et leur patience tout au long de ces années de doctorat. Leur encouragement a été une source inestimable de motivation.

Et enfin, je tiens à remercier chaleureusement mes collègues du LITIV pour les discussions enrichissantes, les précieux conseils et l'entraide constante. Leur camaraderie et leur soutien ont grandement contribué à mon parcours doctoral.

RÉSUMÉ

Le suivi multi-objets consiste à détecter et à attribuer une identité unique à des objets d'intérêt, tels que des piétons et des véhicules, dans une vidéo. Appliqué au génie des transports, il peut permettre d'obtenir les trajectoires des usagers de la route et notamment d'identifier des événements potentiellement dangereux qui ont été évités de justesse. La tâche de suivi peut se décomposer en deux sous-tâches : celle de détection qui consiste à localiser les objets à chaque trame et celle d'association qui consiste à déterminer les détections qui appartiennent au même objet.

Cette thèse s'intéresse à la tâche de suivi multi-objets dans le cadre du domaine des transports et en particulier à la sous-tâche d'association. En effet, malgré l'introduction de nombreuses mesures de performance et de bases de données sur le suivi multi-objets, celles-ci présentent des biais. Tout d'abord, la mesure principale MOTA capture principalement la qualité de la détection au détriment de l'association. Ensuite, les bases de données contiennent principalement des vidéos où les objets d'intérêt ont des apparences variées et des mouvements linéaires. Cela a naturellement poussé à l'apparition de nouvelles méthodes de détection et a privilégié des mesures d'affinité basées sur des descripteurs visuels et sur des modèles de mouvement linéaire.

La première étude présentée porte sur une méthodologie de comparaison entre des descripteurs visuels. Étant donnée une détection d'un objet à un instant d'une vidéo, l'objectif est de retrouver la détection correspondant au même objet à un second instant. Cette recherche est faite en décrivant les détections par un descripteur visuel et en comparant les vecteurs de représentation à l'aide d'une mesure d'affinité. Cette étude a montré que les approches à base de vecteurs de réidentification sont les plus performantes, en plus de présenter une grande robustesse vis-à-vis de l'écart temporel entre les détections et de leur qualité.

Dans un second temps, une approche concurrente aux vecteurs de réidentification est présentée. En effet, ces vecteurs sont vulnérables aux cas d'occlusion lorsqu'un objet secondaire occulte un objet principal. Une localisation des objets à l'échelle du pixel permet de gérer plus exactement la sous-tâche d'association. En particulier, ce travail développe une mesure d'affinité mesurant la capacité à reconstruire une séquence de détections à partir d'une autre séquence de détections. Celle-ci repose sur des réseaux à mémoire spatio-temporelle [1] capables de retrouver la position d'un objet dans une trame adjacente. Cette mesure d'affinité est intégrée à un algorithme de suivi nommé MeNToS, pour *Memory Network-based Tracker of Segments*. Il atteint l'état de l'art sur deux jeux de données centrés sur les usagers de la

route.

Enfin, dans un troisième temps, une autre approche concurrente aux vecteurs de réidentification est présentée. En effet, ces vecteurs sont aussi vulnérables au cas où les objets ont une apparence similaire. Une approche basée uniquement sur la position et le mouvement permet de gérer plus exactement la sous-tâche d'association. En particulier, ce travail développe une mesure d'affinité mesurant la capacité à discriminer des paires de séquences de détections appartenant au même objet ou non. Celle-ci repose sur les réseaux Transformer, qui ont été développés pour décrire des séquences et des ensembles et sont capables de retrouver des schémas en considérant le contexte. Cette mesure d'affinité est nommé TWiX, pour *Tracklets in Windows* et repose sur des paires, comme les fameuses barres de chocolat. L'algorithme de suivi C-TWiX intègre cette mesure d'affinité et atteint l'état de l'art sur trois jeux de données.

Finalement, cette thèse présente une première analyse sur les vecteurs de description visuelle qui montre que les vecteurs de réidentification sont ceux qui sont les plus adéquats quant à l'association de données. Puis, deux approches innovantes sont proposées pour palier quelques faiblesses des vecteurs de réidentification. La première repose sur une approche générative et la seconde sur une approche purement discriminante.

ABSTRACT

Multi-object tracking consists of detecting and assigning a unique identity to objects of interest in a video, such as pedestrians and vehicles. In the field of transportation, it returns the trajectories of road users and in particular helps to identify potentially dangerous events which have been narrowly avoided. The tracking task can be divided into two subtasks: that of detection which consists of locating the objects at each frame and that of association which consists of determining the detections which belong to the same object.

This thesis focuses on the multi-object tracking task in the context of the transportation domain and in particular on the association subtask. Indeed, despite the introduction of numerous performance measures and datasets on multi-object tracking, they both present biases. First of all, the main measure MOTA, mainly captures the detection quality at the detriment of the association. Next, the datasets contain mainly videos where the objects of interest have diverse appearances and linear motions. This naturally led to the emergence of new detection methods and favored affinity measures based on visual descriptors and linear motion models.

The first study presents a methodology for comparing visual descriptors. Given a detection of an object in a frame of a video, the objective is to find the detection corresponding to the same object in another frame. This analysis was done by describing the detections by a visual descriptor and by comparing the representation vectors using an affinity measure. This study shows that approaches based on re-identification vectors are the most efficient, in addition to presenting great robustness with regard to the temporal gap between detections and their quality.

Secondly, a competing approach to re-identification vectors is presented. Indeed, these vectors are vulnerable to occlusion cases when a secondary object hides a main object. Localization at the pixel level makes it possible to manage the association subtask more precisely. In particular, this work develops an affinity measure evaluating the ability to reconstruct a sequence of detections from another sequence of detections. This is based on spatio-temporal memory networks [1] that are able to find the position of an object in an adjacent frame. This affinity measure was integrated into a tracking algorithm called MeNToS, for *Memory Network-based Tracker of Segments*. It achieves the state-of-the-art results on two datasets focused on road users.

Finally, in a third step, another competing approach to re-identification vectors is presented. Indeed, these vectors are also vulnerable in the case where the objects have similar appear-

ances. Here, an approach based solely on position and motion allows the association subtask to be managed more accurately. In particular, this work developed an affinity measure evaluating the ability to discriminate pairs of sequences of detections belonging to the same object or not. This is based on Transformer networks, developed to describe sequences and sets and capable of recognizing patterns by considering the context. This affinity measure is called TWiX, for *Tracklets in Windows* and is based on pairs, like the famous chocolate bars. The C-TWiX tracking algorithm integrates this affinity measure and achieves the state-of-the-art results on three datasets.

Finally, this thesis presents a first analysis on visual description vectors, which shows that the re-identification vectors are those which are the most adequate in terms of data association. Then, two innovative approaches are proposed to overcome some weaknesses of the re-identification vectors. The first one is based on a generative approach and the second one on a purely discriminative approach.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

AP	Average Precision
AssA	Association Accuracy
BIoU	Buffered Intersection over Union
CNN	Convolutional Neural Network
DetA	Detection Accuracy
DIoU	Distance Intersection over Union
FN	False Negative
FNA	False Negative Association
FP	False Positive
FPA	False Positive Association
GIoU	Generalized Intersection over Union
GNN	Graph Neural Network
GT	Ground Truth
HOTA	Higher Order Tracking Accuracy
IDF_1	Identification F_1
IDSw	Identity Switch, ID Switch
IoU	Intersection over Union
LTA	Long-Term Association
mIoU	Mask Intersection over Union
MOT	Multiple Object Tracking, Multi-Object Tracking
MOTA	Multiple Object Tracking Accuracy
MOTS	Multiple Object Tracking and Segmentation, Multi-Object Tracking and Segmentation
OSVOS	One-Shot Video Object Segmentation
PR	Precision-Recall
reID	Réidentification
ROC	Receiver Operating Characteristic
sIoU	Signed Intersection over Union
STA	Short-Term Association
STM	Space-Time Memory
TIoU	Track Intersection over Union
TP	True Positive
TPA	True Positive Association

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

En 2012, la vision par ordinateur a connu une importante révolution. Cette année-là, a lieu la troisième édition d'une compétition organisée par l'université de Princeton : la *ImageNet Large Scale Visual Recognition Challenge* [2]. L'objectif y est de faire concourir des algorithmes de vision par ordinateur pour reconnaître des objets dans des images. C'est plus d'un million d'images qui ont été annotées, réparties sur un millier de classes d'objets. Compte tenu du grand nombre de classes, une certaine tolérance est admise dans le calcul des scores : une prédiction est correcte dès lors que la véritable classe se trouve parmi les cinq choix les plus probables du modèle.

Avant 2012, cette erreur de classification à top-5 était de 25%. Lors de cette troisième édition, une équipe de l'université de Toronto [3] arriva à atteindre une erreur de classification à top-5 de 15.3%, tout en ayant plus de 11 points d'avance sur les deuxièmes. Cette prouesse a été réalisée notamment grâce à des réseaux neuronaux convolutifs et à l'inférence des paramètres du modèle à l'aide de cartes graphiques pour accélérer les calculs au lieu de milliers de coeurs de processeurs, comme ce fut tenté par d'autres chercheurs [4].

Depuis, grâce au partage du savoir scientifique, à l'accès à de meilleurs composants informatiques, à la création de bases de données de plus en plus larges et à une meilleure compréhension théorique des modèles, les algorithmes de traitement d'images battent désormais les humains sur le challenge ImageNet [5].

Dès lors, puisque la tâche de classification d'images atteignait ses limites, il était naturel de passer à un défi plus difficile. Outre la classe de l'objet, il fallait maintenant indiquer la position spatiale de l'objet à l'aide d'une boîte englobante : c'est ainsi que des compétitions de détection ont vu le jour [6]. De même, l'aspect temporal a été ajouté pour obtenir des compétitions de suivi d'objets [7]. Le tableau 1.1 décrit les informations à retourner par les algorithmes sur ces trois tâches et un exemple est présenté pour chacun d'entre eux dans la figure 1.1.

Le sujet de la thèse porte sur le suivi multi-objets, et notamment pour une application dans le domaine des transports.

(a) Classification (b) Détections (c) Suivi multi-objets

FIGURE 1.1 Exemple de classification d'images, de détections et de suivi multi-objets. Images extraites de DanceTrack [8].

TABLEAU 1.1 De la classification au suivi : un nombre croissant d'informations à extraire

Problème	Classification	Détection	Suivi multi-objets
Gestion des classes	étiquette	étiquette	étiquette
Gestion spatiale	-	coordonnées (x,y,w,h)	coordonnées (x,y,w,h)
Gestion temporelle	-	-	identité des objets

1.1 Définitions et concepts de base

1.1.1 Définitions

Le **suivi multi-objets** (*multi-object tracking* ou *multiple object tracking*, abbréviés en MOT) est une tâche en vision artificielle qui consiste à détecter et à suivre dans une vidéo des objets d'intérêt pour extraire leurs trajectoires. Par exemple, comme illustré dans la figure 1.1c, le suivi multi-objets consiste à retrouver toutes les personnes dans une vidéo, localisées à l'aide d'une boîte englobante rectangulaire, et à leur attribuer une identité qui sera la même le long de la vidéo. Cette tâche se décompose naturellement en deux sous-tâches : la **phase de détection** qui consiste à trouver la localisation des objets d'intérêt dans l'ensemble des trames d'une vidéo et la **phase d'association** qui consiste à attribuer une identité à chaque objet.

Dans cet exemple, nous constatons une **occlusion** de la personne indiquée par la boîte englobante bleue à la deuxième trame : il s'agit d'un instant durant lequel un objet n'est plus visible pour la caméra. Il est possible de définir deux niveaux d'occlusion :

1. l'**occlusion partielle** se produit lorsqu'une partie d'un objet d'intérêt est cachée.

L'obstruction visuelle peut provenir d'un autre objet d'intérêt comme une autre personne, d'un élément de décor comme un arbre, ou bien par le bord de l'image ;

2. l'**occlusion totale** se produit lorsque l'entièreté de l'objet d'intérêt est cachée.

Ces événements rendent plus difficiles les sous-tâches de détection et d'association. Comment détecter un objet qui n'est plus du tout visible depuis la caméra ? Comment identifier une personne à partir d'une chaussure ? En raison de ces occlusions, il est possible de définir trois éléments pour la construction des trajectoires :

1. la **détection** : elle indique la position spatiale à un instant donné d'un objet. Aucune identité n'est encore attribuée à une détection à ce niveau. Historiquement, le terme de détection désigne la localisation spatiale à l'aide d'une boîte englobante tandis que le terme de segmentation d'instances (*instance segmentation*) désigne la localisation spatiale au niveau du pixel. Dans la suite de cette thèse, nous désignerons par détection toute indication spatiale, peu importe la nature de cette indication ;
2. le **tracklet** : il indique toutes les positions spatiales entre deux instants d'un objet. Il est à noter qu'ici une identité est attribuée à un tracklet correspondant à un objet. De plus, un tracklet est *continu* : aucune détection n'est manquante au sein d'un tracklet ;
3. le **track** (trajectoire) : il indique toutes les positions spatiales d'un objet. Contrairement au tracklet, certaines détections peuvent être manquantes comme dans le cas d'une occlusion totale. En particulier, les tracklets sont des tracks.

De plus, à travers ces définitions, une hiérarchisation apparaît : en associant les détections entre deux instants consécutifs, il est possible de créer des tracklets. Cette phase d'association de données est nommée la **phase d'association à court terme**. En cas d'occlusion d'un objet, son tracklet est interrompu et un nouveau tracklet est créé lorsqu'un objet non identifié apparaît. Une fois les tracklets obtenus sur l'ensemble de la vidéo, il suffit d'associer les tracklets entre-eux : cette phase d'association de données est nommée la **phase d'association à long terme**.

1.1.2 Application au génie des transports

Le suivi multi-objets peut être appliqué au génie des transports et notamment à la sécurité routière. L'identification d'infrastructures à risque ou de comportements risqués peut exploiter les rapports d'accidents. Cependant, ces évènements sont rares, ce qui limite la portée des analyses. Au contraire, les événements potentiellement dangereux qui ont été évités de justesse, comme une collision évitée entre un véhicule et un cycliste, sont plus nombreux, mais moins bien documentés. Pour pallier ce problème, il est possible d'exploiter des vidéos pour

identifier ces événements. Cela passe par une estimation des trajectoires des usagers de la route rendue possible grâce aux infrastructures déjà existantes comme les caméras de vidéo-surveillance. Une fois les vidéos collectées, il suffit d'appliquer des méthodes automatiques pour repérer les évènements dangereux [9].

Notons ici l'importance de l'association de données dans l'identification de collisions évitées entre deux véhicules. En effet, dans un tel scénario, ces deux véhicules sont spatialement proches, ce qui peut provoquer des occlusions partielles et ainsi compliquer leur détection et leur association. Notamment, une mauvaise association, comme une permutation entre les identités des véhicules, faussera les deux trajectoires ce qui privera l'identification de la collision qui a été évitée de justesse.

1.2 Éléments de la problématique

Dans cette section, nous détaillons les principales difficultés rencontrées dans le développement d'algorithmes de suivi ainsi que le manque d'attention porté sur la phase d'association, qui est pourtant ce qui le distingue de la tâche de détection d'objets dans des vidéos [5].

1.2.1 Principales difficultés dans le suivi multi-objets

Les difficultés portent à la fois sur la capacité d'un algorithme à détecter les objets d'intérêt ainsi que sur sa capacité à assigner la bonne identité.

Difficultés dans la phase de détection

Les difficultés dans la phase de détection dans le cadre de suivi sont en partie les mêmes que celles dans la tâche de détection classique dans des images : la présence d'objets de petite taille, les occlusions partielles, des couleurs similaires à celles de l'arrière-plan ou une diversité de forme d'objets.

De plus, certaines de ces difficultés sont exacerbées dans le cadre d'un suivi. Par exemple, bien que les images présentées dans le cadre de la tâche de détection soient nettes, celles présentes dans les vidéos peuvent être floues en raison du au mouvement de la personne ou de la caméra. Puis, les vidéos de suivi peuvent présenter des distracteurs : ce sont des objets ressemblant fortement aux objets d'intérêt, mais qui n'en sont pas, comme des mannequins dans un magasin de vêtements.

Et enfin, certaines difficultés dans la phase de détection sont inhérentes à la tâche de suivi comme les occlusions totales. Dans ce cas, il est impossible de localiser ces objets à moins de

correctement associer les détections avant et après les occlusions et d'estimer correctement la trajectoire lors de l'occlusion.

Difficultés dans la phase d'association

Outre ces difficultés dans la phase de détection, la phase d'association n'est pas épargnée. Cette phase repose sur des similarités spatiales (on associe deux objets proches) et/ou visuelles (on associe deux objets se ressemblant).

Ainsi, sur le plan spatial, un mouvement trop brusque de la caméra ou d'un objet peut fortement modifier les coordonnées dans l'image ce qui peut entraîner une erreur d'association. De manière analogue à la phase de détection, les occlusions perturbent la phase d'association : pour estimer correctement la nouvelle position de l'objet après une occlusion, il faut développer un modèle de mouvement. Or, selon le caractère statique ou non de la caméra et de l'angle de vue de celle-ci, l'estimation de la trajectoire peut être rendue compliquée.

De même, sur le plan visuel, une forte déformation d'un objet entre deux instants, par exemple une personne qui danse, peut rendre compliquée son association, car son apparence visuelle change beaucoup. Finalement, une dernière difficulté est lorsque les objets d'intérêt ont une apparence visuelle proche comme dans l'exemple de la figure 1.1c. Dans ce cas, le caractère discriminatoire de l'algorithme doit être suffisamment élevé pour distinguer ces deux personnes.

Et enfin, il arrive que des objets d'intérêt sortent du champ de la caméra et que de nouveaux objets y entrent. Dans le cas où cela se produirait de manière simultanée, il est primordial de bien identifier une sortie et une entrée et non une simple occlusion.

Difficultés dans la conception d'algorithmes de suivi

Contrairement à la tâche de classification où l'objectif est de prédire la classe d'un objet, il est nécessaire de localiser les objets d'intérêt et d'assigner une identité de manière consistante dans le cadre d'un suivi multi-objets. Cela nécessite alors la création d'algorithmes capables de gérer à la fois la dimension spatiale et temporelle. Toutefois, ces deux dimensions ne sont pas indépendantes : une estimation des trajectoires (donc dans le plan spatial) peut permettre une meilleure association (dans le plan temporel). Et inversement, une association peut permettre d'estimer des positions dans le cas d'une occlusion.

De plus, comme cela sera mentionné dans la section 2.1, le suivi multi-objets comportent de nombreuses déclinaisons qui rendent compliquée l'apparition d'un cadre commun. Il est possible de trouver des modèles de suivi exploitant de l'information future ou non, en temps

réel ou non et à différents niveaux de localisation.

Et enfin, comme cela sera mentionné dans la section 2.6, les bases de données en suivi sont comparativement plus petites que les bases de données dans les tâches de classification ou de détection. Cela explique pourquoi les chercheurs pré-entraînent leurs modèles sur ces deux tâches en amont. Toutefois, cela complique une comparaison juste entre les modèles.

1.2.2 Un déséquilibre entre détection et association

Étant donnée la définition du suivi multi-objets, deux phases apparaissent naturellement : la phase de détection et la phase d'association. Or, une attention moindre a été portée sur la seconde phase que sur la première.

Bases de données préalablement disponibles

Le premier déséquilibre provient des bases de données. Tout d'abord, pour résoudre la phase de détection, il est possible de se baser sur un modèle pré-entraîné sur une base de données de détection. Dans ce cas, nous disposons de grandes bases de données de détection. De même, la tâche de détection d'objets dans des vidéos (*video object detection*) peut servir d'intermédiaire entre la tâche de détection (localisation dans une image) et la tâche de suivi multi-objets (localisation et assignation d'identité dans une vidéo).

En ce qui concerne la phase d'association, il existe des banques de données d'images pour la tâche de réidentification. L'objectif est de retrouver un objet, généralement une personne, parmi une banque d'images. Ces bases de données sont obtenues à partir de vidéos capturées par plusieurs caméras pointant sur une même scène ce qui fournit plusieurs angles de vue de la même personne. La réidentification peut s'effectuer entre deux images, ou bien entre des séquences (tracklets) de deux personnes. Ici, l'aspect temporel pourra être exploité pour affiner le modèle d'apparence. Toutefois, avec ces bases de données de réidentification, il n'est pas possible de combiner l'information spatiale comme la position ou le mouvement et l'information d'apparence, car ces informations spatiales ne sont pas toujours disponibles. C'est pourquoi les modèles de mouvements sont généralement des heuristiques et non le fruit d'un entraînement sur une base de données.

De plus, dans le contexte d'un suivi multi-objets, il peut être intéressant de tenir compte du contexte, c'est-à-dire des autres objets se déplaçant dans le voisinage d'un objet d'intérêt. Par exemple, pour suivre une personne au sein d'un groupe habillé de manière similaire, connaître la position des autres individus peut aider à résoudre le problème de réidentification.

Mesure de la performance d'un algorithme de suivi

Le deuxième déséquilibre concerne la mesure de la performance d'un algorithme de suivi. Lors du colloque CLassification of Events, Activities and Relationships (CLEAR) en 2006, la mesure MOTA [10] a été proposée pour évaluer la qualité d'un algorithme de suivi. Celle-ci est calculée sur le nombre d'erreurs de détection et d'erreurs d'association. De plus amples informations sont fournies à son sujet dans la section 2. Or, il s'avère que cette mesure n'équilibre pas les deux types d'erreurs : Luiten et al [11] ont trouvé que la mesure MOTA avait une corrélation linéaire de plus de 99% avec la qualité de la détection. Ainsi la mesure MOTA, très longtemps plébiscitée pour mesurer la qualité d'un algorithme de suivi, ne mesure que sa composante de détection.

Biais dans les bases de données de suivi

En effet, une troisième cause du déséquilibre entre détection et suivi a pour origine les bases de données de suivi multi-objets. Sun et al [8] ont montré que celles-ci contenaient des séquences vidéos où l'apparence des personnes était assez diversifiée de telle sorte qu'un algorithme basé sur l'apparence suffisait à résoudre le problème d'association. Ainsi, la composante mouvante des objets a été peu étudiée. De plus, les mouvements observés dans les vidéos étaient souvent linéaires, ce qui limitait l'élaboration de méthodes plus sophistiquées. C'est pourquoi ils ont proposé une nouvelle base de données, DanceTrack, dans laquelle l'apparence des individus est similaire tout en ayant des mouvements à la fois rapides et non linéaires.

1.2.3 La nature du problème d'association de données

La tâche d'association de données est un problème complexe à résoudre en raison de sa nature très particulière.

Tout d'abord, il est possible de résoudre cette tâche avec des algorithmes de regroupement de données (*clustering*) [12]. En effet, ce regroupement pourrait se faire en appliquant un encodeur sur les détections suivi d'un algorithme tel que K-means ou DBSCAN pour créer des groupes de détections auxquelles la même identité sera assignée. Or, il ne s'agit pas d'un problème de cette nature. Contrairement au problème de regroupement de données qui est un problème non supervisé, le problème d'association de données est un problème supervisé. Il est possible d'attribuer une identité de vérité terrain pour toutes les détections obtenues. Il s'agit en fait d'**un problème de classification binaire** : étant donné deux détections, s'agit-il du même objet ?

Il est possible d'observer que le problème est **déséquilibré** : il y a beaucoup plus de paires

négatives que de paires positives. Ainsi, si on souhaite associer 10 objets entre deux trames consécutives, il est possible de créer 100 paires dont seules 10 seront au plus positives.

Ensuite, ce problème de classification binaire est hautement **hiérarchique** : il est bien plus simple de déterminer si deux détections appartiennent au même objet lorsqu'ils sont séparés d'une trame que lorsqu'ils sont séparés de plusieurs trames. Il est possible de distinguer les cas où les détections proviennent de deux trames adjacentes (association à court terme, STA pour *short-term association*) des cas où des détections sont manquantes en raison d'une non-détection ou d'une occlusion (association à long terme, LTA pour *long-term association*). Les seconds sont à la fois plus complexes à résoudre, mais également plus rares dans les banques de données. Ainsi, **l'association à long terme est un problème où les données sont rares**.

Lorsqu'un modèle de mouvement est implémenté, il ne tient généralement pas en compte de la présence des autres objets. Or, dans le cas des piétons, leur circulation dépend des autres piétons. Ils peuvent soit se mouvoir dans le même sens, dans le cas d'un groupe d'amis, ou bien s'éviter dans le cas de deux flux de piétons de sens opposé. Ainsi, que cela soit au sein de la même trame ou bien entre deux trames temporellement proches, **les détections ne sont pas indépendantes entre elles**. Traiter les associations sans tenir compte tenir du contexte rend le problème ambigu.

Et enfin, il existe plusieurs signaux de nature différente pour résoudre cette tâche : **l'association des données est un problème multimodal**. Ainsi, il est possible d'exploiter la position, le mouvement, l'apparence ou encore les scores de confiance obtenus lors de la détection pour mesurer une similarité entre les détections. De plus, plusieurs niveaux de précision existent parmi ces signaux : pour la position, il est possible d'utiliser les coordonnées des boîtes englobantes ou bien celles des joints dans le cas de l'estimation de pose d'une personne. À cela s'ajoute **la structure séquentielle** des données : au cours de l'association des données, il devient possible non pas d'associer deux détections, mais bien une séquence de détections (tracklet) avec une détection.

1.3 Objectifs de recherche

L'objectif principal de cette thèse est d'étudier et de développer des algorithmes d'association de données dans le cadre d'un suivi multi-objets à prise de vue unique (*single camera multi-object tracking*) où les détections sont décrites par des boîtes englobantes (MOT) ou à l'échelle du pixel (MOTS). Plus spécifiquement, nos objectifs sont de :

1. Concevoir une méthodologie pour mesurer la qualité de descripteurs visuels quant à

l'association de données dans le cadre d'un suivi MOT. En effet, les descripteurs visuels sont très utilisés dans la littérature pour cette sous-tâche. Cette évaluation prendra en considération la mesure d'affinité vectorielle, la distance temporelle entre les détections et la robustesse des descripteurs à la qualité des détections. La méthodologie doit permettre d'étudier des descripteurs visuels pour la sous-tâche d'association de données et d'en comprendre les limites ;

2. Développer une méthode sans descripteur visuel qui utilise les informations spatiales et visuelles pour le MOTS. La connaissance à l'échelle des pixels permet une exploitation plus fine de l'apparence visuelle. Celle-ci ne sera pas décrite par des descripteurs visuels. Cette méthode sera intégrée à un algorithme de suivi par détection pour la mesure de performance ;
3. Développer une méthode qui utilise les informations spatiales pour le MOT. En effet, la position et le mouvement ont été moins étudiés dans le cadre d'un problème d'association de données. Elle reposera sur un modèle n'exploitant que les coordonnées spatio-temporelles des détections, sans tenir compte de l'apparence des objets. Ce modèle doit retourner des matrices d'affinité où l'affinité entre deux détections dépend de leurs caractéristiques, mais également de celles des autres objets avoisinants. Cette méthode sera intégrée à un algorithme de suivi par détection pour la mesure de performance.

1.4 Contributions

Les travaux décrits dans cette thèse ont été effectués en trois phases entre 2019 et 2024. Voici les contributions réalisées au sein de chaque article répondant aux objectifs de recherche. Le cœur de ces contributions porte sur la sous-tâche d'association et en particulier l'estimation de l'affinité entre les objets dans le cadre d'un suivi.

Évaluation de descripteurs visuels : pour ce premier objectif, nous avons développé une méthodologie dans l'article *An empirical analysis of visual features for multiple object tracking in urban scenes* [13] présentée à la conférence ICPR de 2020 (article du chapitre 4). Elle a été développée en mesurant la capacité d'un descripteur visuel à associer correctement deux détections distantes de Δt trames dans un cadre de suivi. Une analyse de neuf descripteurs associés à cinq mesures d'affinité vectorielle a été menée pour en évaluer leur performance dans l'association de données et leurs limites.

Méthode d'association à partir des positions et de l'apparence pour le suivi MOTS : pour le deuxième objectif, nous avons développé l'algorithme MeNToS présenté dans l'article *Multi-object tracking and segmentation with a space-time memory network* [14] à la conférence CRV de 2023 (article du chapitre 5). Il repose sur un module d'association qui génère des cartes de chaleur en considérant deux trames et un masque de référence. Évalué sur deux bases de données de suivi MOTS, le module d'association a montré qu'il fournissait une mesure d'affinité plus robuste au choix du seuil d'association que des descripteurs visuels classiques.

Méthode d'association à partir des coordonnées pour le suivi MOT : pour le troisième objectif, nous avons développé l'algorithme C-TWiX présenté dans l'article *Learning data association for multi-object tracking using only coordinates* soumis à la revue Pattern Recognition en 2024 (article du chapitre 6). Il intègre un nouveau module, nommé TWiX, qui retourne une matrice d'affinité d'un ensemble de tracklets en considérant les positions passées et courantes des objets. Évalué sur trois bases de données de suivi MOT, le module d'association a montré qu'il fournissait une mesure d'affinité plus discriminante que les autres mesures d'affinités classiques basées sur la position et/ou le mouvement.

1.5 Plan de la thèse

Cette thèse est structurée en huit chapitres. Après avoir introduit le sujet de la thèse dans ce chapitre, le chapitre 2 présente une revue de la littérature relative au suivi d'objets et à la sous-tâche d'association. Le chapitre 3 explique la démarche expliquant les liens entre les articles composant le corps de la thèse. Ainsi, le chapitre 4 porte sur l'évaluation de descripteurs visuels, le chapitre 5 sur un algorithme de suivi à l'échelle du pixel et le chapitre 6 sur un algorithme de suivi à l'aide de boîtes englobantes. Ensuite, le chapitre 7 présente une discussion sur les travaux, les difficultés rencontrées et des solutions apportées. Pour finir, le chapitre 8 présente une conclusion de la thèse donnant des pistes de recherche pour de futurs travaux en suivi d'objets.

CHAPITRE 2 REVUE DE LITTÉRATURE

Ce chapitre traite de la revue de littérature concernant le suivi d'objets dans des vidéos. Tout d'abord, la section 2.1 montrera que la tâche de suivi d'objets dans des vidéos peut prendre des facettes très variées. Puis seront présentées les deux grandes sous-tâches du suivi, à savoir la localisation d'objets dans la section 2.2 et l'association de données. Cette dernière peut se faire en deux temps : d'abord par un calcul d'affinité, qui sera présenté dans la section 2.3, suivi d'une étape d'assignation dans la section 2.4. Puis, certains algorithmes de suivi et certains post-traitements qui leur sont spécifiques seront présentés dans la section 2.5. Ensuite, les bases de données les plus populaires ainsi que les mesures de performance pour la tâche de suivi multi-objets seront présentées dans la section 2.6. Pour terminer, l'apprentissage par contraste (*contrastive learning*) sera présenté dans la section 2.7, qui vise à apprendre des représentations en plaçant des exemples similaires plus proches les uns des autres dans l'espace de représentation, tout en éloignant les exemples dissimilaires.

2.1 Suivi d'objets

Le suivi d'objets trouve ses applications dans de nombreux domaines. Par exemple, dans le secteur du génie des transports, l'utilisation des vidéos extraites des caméras de surveillance peut permettre de compter les différents usagers de la route, d'identifier des zones à risque d'accident ou encore de mesurer la pertinence de politiques d'urbanisation. Pour les véhicules autonomes, utiliser un système de suivi au-delà de la simple détection des usagers de la route permet de tenir compte de la présence de piétons dans le cas où ceux-ci font l'objet d'une occlusion totale et de prédire leurs trajectoires pour anticiper des collisions. Ou encore, pour le besoin d'édition dans le cinéma, il est parfois nécessaire de supprimer un élément dans une vidéo. Dans ce cas, il faut suivre l'objet en question pour le supprimer dans chacune des trames.

Ces exemples montrent que la tâche de suivi d'objets peut prendre plusieurs facettes selon son domaine d'application. Les différences portent sur plusieurs aspects :

1. la nature de la localisation ;
2. la nature du conditionnement initial ;
3. la nécessité de fonctionner en temps réel ;
4. les capteurs exploités.

Pour l'analyse de vidéos de la circulation routière, une localisation des usagers de la route à

l'aide de boîtes englobantes peut suffire. Pour la conduite de véhicules autonomes, la prise en compte de la profondeur est nécessaire : généralement, les usagers de la route sont localisés à l'aide de boîtes en trois dimensions. Quant à l'édition vidéo, il est nécessaire de localiser les objets à l'échelle du pixel pour réaliser des trucages vidéos. De plus, pour l'édition de vidéo, il est nécessaire d'indiquer en amont l'objet à suivre pour le supprimer. Un tel renseignement n'est pas nécessaire dans les deux autres exemples. Et enfin, seule la tâche de conduite autonome requiert le développement d'un algorithme en temps réel. Quant aux capteurs, les véhicules autonomes peuvent disposer de plusieurs caméras et/ou d'un LiDAR pour estimer la distance des objets.

Dans cette thèse, nous ne nous intéresserons qu'au suivi multi-objets, conditionné par des classes d'objets, à l'aide de boîtes englobantes ou de masques et filmés par une unique caméra en RGB. Il s'agit des tâches de MOT (*multiple object tracking*) et de MOTS (*multiple object tracking and segmentation*).

2.1.1 Suivi en ligne et hors ligne

Lorsque l'algorithme de suivi doit s'appliquer en temps réel, il doit prendre des décisions (en termes de détection et d'association) au fur et à mesure de l'arrivée des données. Ainsi, c'est un algorithme en ligne : il ne peut ni utiliser d'information provenant du futur, ni modifier ses résultats du passé. En opposition, un algorithme hors ligne peut traiter l'ensemble de l'information (ici, la vidéo au complet) avant de prendre une décision.

Les algorithmes de suivi en temps réel doivent non seulement obtenir des détections en temps réel, mais aussi disposer d'un algorithme d'association en ligne. Tandis que les autres algorithmes peuvent exploiter des algorithmes de détection plus performants au détriment de la rapidité et des algorithmes d'association hors ligne.

Dans le cadre d'un algorithme de suivi pour l'analyse de vidéo pour des études de circulation, de comportements et de sécurité des usagers de la route, un algorithme en temps réel n'est pas nécessaire. Nous pouvons à la fois utiliser des algorithmes de détection plus lents, mais plus performants, mais également des algorithmes d'association en mode hors ligne ainsi que des post-traitements pour améliorer les résultats.

2.1.2 Paradigmes des algorithmes de suivi

Les deux sous-tâches d'un algorithme de suivi sont la détection et l'association. Lorsque l'association est appliquée sur des détections préalablement obtenues, on parle de paradigme de **suivi par détection**. Ce paradigme est populaire, car, outre le fait de simplifier la

tâche de suivi à un problème d'association, il peut reposer sur des algorithmes de détection pré-entraînés et plus performants. De plus, il peut suivre l'amélioration des algorithmes de détection en les mettant à jour.

Il est également possible d'exploiter l'association pour améliorer la qualité des détections. Par exemple, avec le paradigme du **suivi par régression** [15], les détections à l'instant $t+1$ sont obtenues à partir des détections à l'instant t. Dans ce cas, l'association des détections est faite naturellement. Les difficultés restantes sont la détection d'une sortie d'un objet suivi ou d'une entrée d'un nouvel objet.

Avec le paradigme de **suivi par attention** [16], le mécanisme d'attention est utilisé pour détecter et associer les objets simultanément. Ici, un token représente un objet au cours de la vidéo. Celui-ci est mis à jour de manière auto-régressif pour s'adapter à la nouvelle position et/ou apparence de l'objet suivi.

Dans notre cas, nous avons privilégié le paradigme de suivi par détection pour plusieurs raisons. Tout d'abord, c'est celui qui obtient les plus hautes performances. De plus, comme mentionné en introduction, la tâche de suivi multi-objets est principalement une tâche d'association. Ainsi, en fixant les détections, il est plus simple de comparer des algorithmes d'association. Et enfin, la sous-tâche d'association est celle qui nous intéresse le plus, comme mentionné dans la section 1.1.2.

2.2 Localisation d'objets

En suivant le paradigme de suivi par détection, la première étape consiste à détecter les objets d'intérêts, décrits par leur classe (e.g. véhicules, piétons, train, etc) dans la vidéo. Cette détection peut être appliquée à différents niveaux de précision.

2.2.1 Différents niveaux de localisation d'objets

Pour le MOT, les détections sont faites à l'aide de boîtes englobantes, et pour le MOTS, elles sont à l'échelle du pixel avec des masques binaires. Les premières seront présentées dans la section 2.2.2 et les secondes dans la section 2.2.3.

Ces localisations peuvent être décrites à partir des coordonnées dans l'image (*image coordinates*) ou bien à partir des coordonnées dans la scène filmée (*world coordinates*), après calibration de la caméra. Dans les cas de MOT et MOTS, les localisations sont typiquement décrites à partir des coordonnées dans l'image. En complément, le tableau 2.1 présente d'autres stratégies de localisation d'objets. Ces localisations permettent soit d'augmenter la

performance du suivi [17] comme avec des boîtes englobantes avec rotation, soit de simplifier la localisation pour accélérer la détection [18], comme avec des polygones.

TABLEAU 2.1 Localisation d'un objet dans une image de taille $H \times W$. Une référence est fournie pour chaque localisation employée dans un algorithme de suivi ou de détection.

Localisation	Degrés de liberté	Exemple
Point [19]	2	(x_c, y_c)
Boîte englobante [20]	4	$(x_{min}, y_{min}, x_{max}, y_{max})$
Boîte englobante avec rotation [17]	5	$(x_{min}, y_{min}, x_{max}, y_{max}, \theta)$
Ellipse/Gaussienne 2D [17]	5	$((x_c, y_c), \Sigma \in \mathbb{R}^{2\times2})$
Boîte 3D reposant sur le sol [21]	7	$(x, y, z, h, w, d, \theta)$
Polygone à s sommets [22]	$2s$	$((x_1, y_1), \ldots, (x_s, y_s))$
Squelette à j joints [23]	$2j$	$((x_1, y_1), \ldots, (x_j, y_j))$
Masque binaire [12]	$O(HW)$	cf Figure 2.1
Carte de chaleur [14]	$O(HW)$	cf Figure 2.1

La figure 2.1 illustre les différentes stratégies de localisation d'une personne. Une localisation intermédiaire plus précise est parfois employée dans des algorithmes de suivi telle que des boîtes englobantes avec rotation [17] ou des cartes de chaleur [14].

2.2.2 Méthodes pour localiser à l'aide de boîte englobante

Lorsque la nature de la localisation est une boîte englobante, on parle généralement de tâche de détection. On distingue généralement celles nécessitant une ou deux étapes. Parmi les méthodes à deux étapes, Faster R-CNN [24] est un des principaux représentants. Il est composé d'un premier réseau de neurones convolutif (*convolutional neural network*, CNN) qui propose des régions candidates. Un second réseau considère chaque candidat et lui attribue une classe et une position précise dans le cas où la région candidate correspond bien à un objet d'intérêt.

À l'inverse, YOLO [25] est un détecteur à une étape. Au lieu de proposer des régions candidates, cet algorithme divise une image en une grille où chaque cellule est responsable de détecter les objets qui s'y trouvent. Ce détecteur est ainsi capable de fonctionner en temps réel. Il a été amélioré par les auteurs originaux [26, 27] et par d'autres groupes de chercheurs [28]. Ainsi, YOLOX, proposé en 2021, intègre quelques modifications d'architectures (têtes du réseau découplé entre la partie de régression et de classification), une plus grande utilisation d'augmentation de données ou encore une meilleure assignation des cellules responsables.

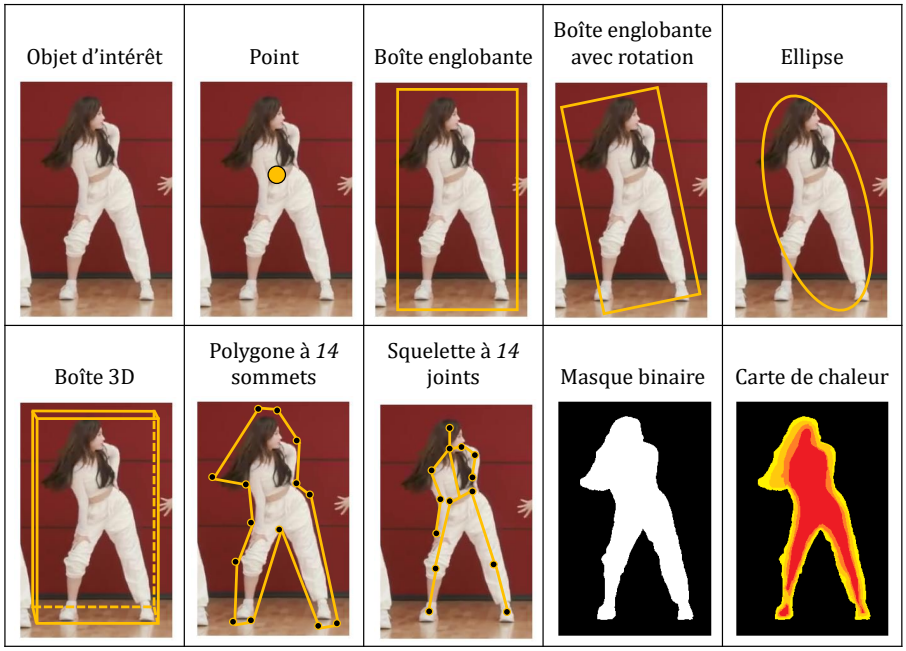


FIGURE 2.1 Différentes stratégies de localisation d'une personne

2.2.3 Méthodes pour localiser à l'échelle du pixel

La localisation d'objets à l'échelle du pixel peut être le résultat de plusieurs tâches. La segmentation sémantique (*semantic segmentation*) consiste à retourner une classe pour chaque pixel d'un objet d'intérêt. Ainsi, deux pixels appartenant à deux objets de même nature devront avoir la même sortie. La segmentation d'instance (*instance segmentation*) retourne une classe et une identité à chaque pixel. Ainsi, deux pixels appartenant au même objet devront avoir la même identité, et elle devra être différente dans le cas contraire. La segmentation panoptique (*panoptic segmentation*) [29] combine les forces de ces deux approches : pour chaque pixel, s'il correspond à un *stuff* (une région non dénombrable et qui n'a pas de forme ou de contours distincts comme le ciel, l'herbe et les arbres), alors seule la classe d'objet est retournée et s'il correspond à un *thing* (un objet dénombrable qui peut être identifié et segmenté individuellement comme des véhicules, des piétons, des chiens), alors on retourne à la fois la classe et l'identité de l'objet. Et enfin, la segmentation par prompt (*promptable segmentation*) [30] retourne une segmentation à partir d'un conditionnement initial. La condition peut être un point appartenant à un objet d'intérêt, une boîte ou encore une classe. L'algorithme doit alors retourner le masque de l'objet qui lui est associé. Dans le cadre du suivi d'objets, c'est la segmentation d'instances qui répond au plus près à la problématique de MOTS.

Pour localiser plus finement des objets, il est possible de modifier Faster R-CNN en ajoutant

une branche pour retourner un masque. Ainsi Mask R-CNN [31] extrait d'abord des régions d'intérêt, puis il retourne un masque pour chacune d'elle. Les auteurs ont également innové en alignant correctement les cartes de représentation avec les régions d'intérêt à travers l'utilisation de ROIAlign au lieu de ROIPool qui utilise une interpolation bi-linéaire au lieu d'une quantification trop imprécise. SOLO [32] propose une reformulation de la tâche de segmentation d'instances : une image est découpée en grille où chaque cellule est responsable de la prédiction de la catégorie sémantique et de la segmentation d'instance de l'objet qui s'y trouve.

Il est également possible d'entraîner des algorithmes à raffiner des boîtes [33, 34], ou des masques binaires [35]. Par exemple, Box2Seg [34] est un réseau de neurones qui a été entraîné à retourner une segmentation pour une image dont une boîte englobante a été fournie en tant que quatrième canal. Ou encore BPR [35] est entraîné à retourner des segmentations de haute résolution à partir de sous-images extraites à la frontière des objets. En effet, les masques obtenus par segmentation présentent généralement des défauts à leur frontière et non en leur centre.

2.3 Calcul d'affinité pour la phase d'association

Avec le paradigme de suivi par détection, la seconde tâche consiste à associer les détections entre deux trames. Celle-ci peut reposer sur des informations de nature différente : la position spatiale, le mouvement, l'apparence ou bien une combinaison de ces signaux. L'objectif est ici d'obtenir une matrice de coût entre deux ensembles d'objets. Cela permet à la fois d'associer des objets dans le cadre du suivi, mais aussi de mesurer la performance d'un algorithme de suivi en considérant les détections et la vérité terrain au même instant. L'ensemble des méthodes présentées ci-dessous permet d'obtenir une matrice d'affinité entre deux ensembles d'instances.

2.3.1 Affinité basée sur la position spatiale

L'affinité basée sur la position spatiale peut être calculée en comparant deux localisations ou deux ensembles de localisations.

Affinité avec les distances L_p. Pour deux localisations décrites par les coordonnées $x = (x_1, \cdots, x_n)$ et $y = (y_1, \cdots, y_n)$, la distance L_p est calculée de la manière suivante :

$$L_p(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \tag{2.1}$$

Pour $p = 1$, la distance est appelée la distance de Manhattan, et pour $p = 2$, elle est appelée la distance euclidienne.

Distance de Mahalanobis. Cette distance d_M introduite par Mahalanobis [36] permet de généraliser la distance euclidienne. Pour deux vecteurs x et y, elle est donnée par :

$$d_M(x, y) = \sqrt{(x - y)^\top \Sigma^{-1} (x - y)}, \tag{2.2}$$

où Σ est une matrice de covariance. Cette matrice permet à la fois de quantifier l'incertitude comme dans l'algorithme Deep SORT [37] et de considérer la corrélation entre les différents axes de représentation pour tenir compte de l'angle de vue d'une caméra [38].

Affinité entre deux boîtes englobantes. Avec deux boîtes, b_1 et b_2, la mesure d'intersection sur union (IoU pour *intersection over union*) peut être calculée de la manière suivante :

$$IoU(b_1, b_2) = \frac{I}{A_1 + A_2 - I} \tag{2.3}$$

avec A_1 (resp. A_2) l'aire de la boîte englobante b_1 (resp. b_2) et I l'aire de l'intersection entre b_1 et b_2 (si l'intersection est vide, I vaut 0).

Cette mesure a pour avantage d'être rapide et simple à calculer, symétrique entre les deux boîtes, bornée entre 0 et 1, invariante par translation et par changement d'échelle. Toutefois, l'un des désavantages de IoU est qu'elle atteint zéro dès que l'intersection entre les deux boîtes devient vide. Ainsi, l'IoU entre deux boîtes distantes de quelques pixels sans intersection et entre deux boîtes très distantes seront toutes les deux à zéro. Dans le cadre d'un suivi d'objets, cela se produit lorsqu'un objet se déplace rapidement, ou bien lorsque la caméra bouge, ou encore que l'enregistrement a été faite à une faible cadence d'images par seconde.

Plusieurs mesures alternatives ont été proposées pour palier ce problème. La *Generalized IoU* (GIoU) [39] a été proposée en 2019 en tant que fonction de perte pour la tâche de détection. Elle est ainsi une borne inférieure de IoU, car elle ajoute une pénalité qui correspond au ratio d'écart d'alignement entre les boîtes. De même, la *Distance IoU* (DIoU) [40] a été proposée pour accélérer la convergence de la fonction de perte. Enfin, la *Signed IoU* (sIoU) [41] a

introduit des intersections négatives au cas où l'intersection est vide. Toutes ces mesures ont été originellement introduites en tant que fonctions de perte, mais elles peuvent être utilisées pour décrire l'affinité entre deux boîtes. Et enfin, la mesure *Buffered IoU* (BIoU) [42] a été proposée en agrandissant proportionnellement la taille des deux boîtes.

Les figures 2.2 et 2.3 illustrent le calcul des cinq mesures à base de IoU calculées sur des paires de boîtes englobantes.

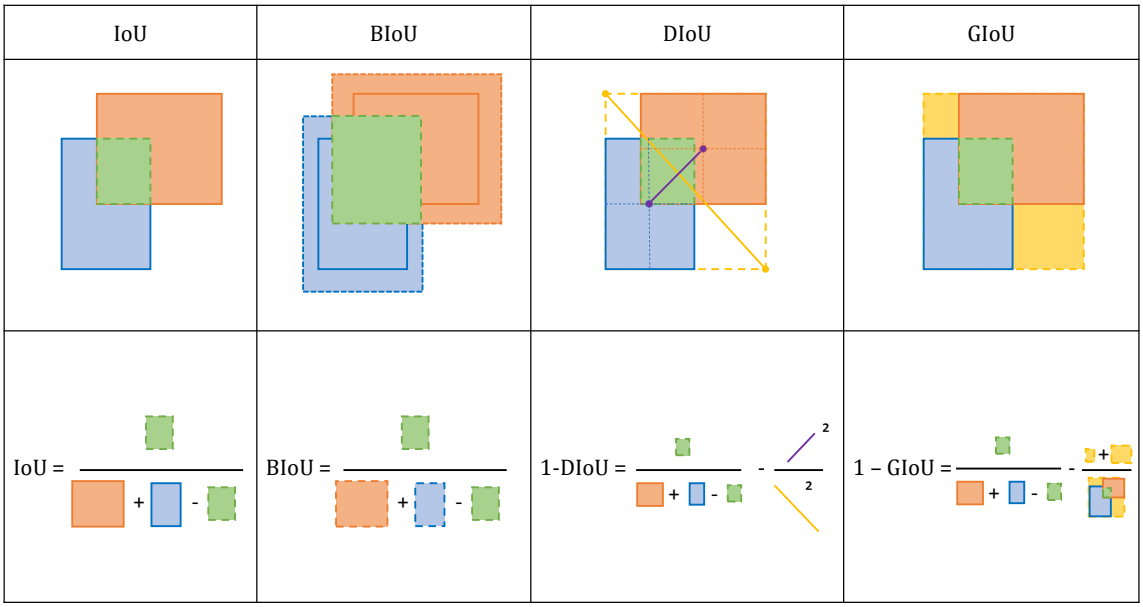


FIGURE 2.2 Illustration des mesures d'affinité à base de IoU (excepté sIoU)

Affinité entre deux masques binaires. La mesure IoU peut être généralisée à des masques binaires. En notant M_1 et M_2, deux masques binaires de taille $H \times W$, le *mask intersection over union* (mIoU) [1] est défini de la manière suivante :

$$mIoU(M_1, M_2) = \frac{|M_1 \cap M_2|}{|M_1 \cup M_2|}. \tag{2.4}$$

En particulier, la mesure IoU coïncide avec mIoU si les boîtes englobantes sont transformées en des masques rectangulaires.

1. Dans la littérature, on peut parfois trouver le terme de segment IoU (sIoU) [43] pour désigner mIoU

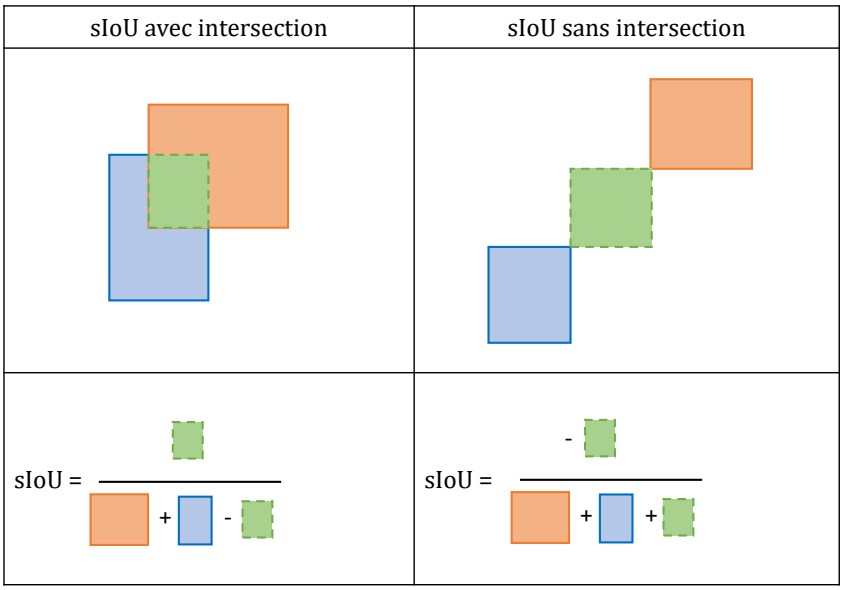


FIGURE 2.3 Illustration des mesures d'affinité à base de sIoU, dans le cas où l'intersection est vide ou non

Affinité entre deux ensembles de boîtes. Il est également possible de mesurer une affinité entre deux ensembles de boîtes $T_1 = (b_1^j)_{j \in \mathcal{S}_1}$ et $T_2 = (b_2^j)_{j \in \mathcal{S}_2}$ respectivement présents durant les trames des ensembles $\mathcal{S}_1$ et $\mathcal{S}_2$. Cela est nécessaire dans le cas où l'on souhaite connaître l'identité d'un tracklet T_1 par rapport au track de la vérité terrain T_2 (potentiellement partiel, c'est-à-dire $\mathcal{S}_1 \subset \mathcal{S}_2$). Ainsi, le Track IoU (TIoU) est une généralisation de la mesure IoU appliquée sur des tracks :

$$TIoU(T_1, T_2) = \frac{1}{|\mathcal{S}_1 \cap \mathcal{S}_2|} \sum_{i \in \mathcal{S}_1 \cap \mathcal{S}_2} IoU(b_1^i, b_2^i) \tag{2.5}$$

La mesure TIoU peut être aisément interprétée comme un IoU moyen sur les trames en commun [2].

Si l'on souhaite associer un track constitué de détections à un track issu de la vérité terrain, il est possible d'adapter TIoU en considérant l'union entre $\mathcal{S}_1$ et $\mathcal{S}_2$ au lieu de leur intersection.

2. Une autre formulation existe [44] calculée avec $TIoU(T_1, T_2) = \frac{\sum_{i \in \mathcal{S}_1 \cap \mathcal{S}_2} b_1^i \cap b_2^i}{\sum_{i \in \mathcal{S}_1 \cap \mathcal{S}_2} b_1^i \cup b_2^i}$

2.3.2 Affinité basée sur le mouvement

Dans le cadre d'un problème d'association, mesurer l'affinité entre deux boîtes à des instants t et $t+1$ par la position spatiale est une approximation, car cela repose sur l'hypothèse que les objets bougent peu entre deux instants consécutifs. Il est possible de raffiner les mesures d'affinité en tenant compte du mouvement, qui se décompose entre le mouvement propre de la caméra et le mouvement des objets. L'estimation de la première partie permet de supprimer une composante dans l'estimation des coordonnées des objets et ainsi améliorer l'estimation du mouvement intrinsèque des objets.

Estimation du mouvement d'une caméra

Dans certains cas, le mouvement d'une caméra est connu comme par exemple pour les véhicules, car les contrôles des roues sont accessibles : on parle d'*ego-motion*. Dans les autres cas, il faut exploiter les images RGB pour en extraire un mouvement intrinsèque de la caméra.

Pour cela, le recalage d'image (*image registration*) est la méthode la plus populaire. Elle consiste à trouver des correspondances entre des pixels de l'instant t et $t+1$, comme avec ORB [45] puis à estimer le mouvement global de la caméra par RANSAC [46] qui est une méthode robuste à la présence de valeurs aberrantes.

L'estimation du mouvement d'une caméra doit être particulièrement rapide car elle est appliquée à chaque paire de trames. C'est pourquoi le flux optique n'est pas utilisé ici.

Estimation du mouvement d'un objet

Cette section traite des méthodes estimant le mouvement d'un objet, dans un premier temps pour des boîtes englobantes, puis pour des masques à l'aide du flux optique.

Mouvement rectiligne uniforme. Le modèle le plus simple consiste à supposer la trajectoire comme rectiligne et uniforme : la trajectoire suit une ligne droite et sa vitesse est constante. Cette approche est simple et ne fait pas intervenir d'hyper-paramètres. Appliquée sur un track de boîtes englobantes, on peut soit supposer une hauteur et largeur constantes, soit les extrapoler au risque d'obtenir des valeurs négatives après un certain horizon (par exemple, quand un objet devient plus petit au cours du temps). C'est pourquoi des modèles plus avancés ont été développés.

Estimation par un filtre de Kalman. Le filtre de Kalman [47] est un algorithme bayésien qui permet d'estimer de manière auto-régressive les valeurs d'une série bruitée. Il réside sur

des hypothèses de linéarité et de bruits gaussiens : il considère des changements d'état linéaires d'une variable cachée et l'état d'une variable observable est lié linéairement à cette variable cachée.

Parmi les limitations de l'utilisation du filtre de Kalman dans le suivi d'objets, on peut citer une sensibilité au bruit et une augmentation de l'erreur lors d'une occlusion en raison de l'accumulation du bruit. Ainsi, Cao et al. [48] ont proposé de centrer les équations non sur les variables d'observation, mais sur les variables cachées. Les auteurs ont montré que cela rend l'algorithme de suivi plus robuste aux occlusions et aux mouvements non linéaires.

Estimation par un réseau de neurones. Le mouvement d'un objet peut être également appris à l'aide d'un réseau de neurones comme un réseau récurrent tel que LSTM [49] ou un Transformer [50]. Dans le premier cas, le mouvement d'une personne est modélisé de manière auto-régressive en prédisant le mouvement d'abord à $t + 1$ puis à $t + 2$ et ainsi de suite à partir de l'état précédent. Dans le second cas, l'ensemble des observations passées est exploité pour prédire l'ensemble des positions futures, ce qui permet d'obtenir de meilleures prédictions, comme l'a montré TrajNet [51]. Dans les deux cas, la position future d'un objet est déterminée indépendamment de celles des autres objets.

Estimation conjointe des trajectoires. Particulièrement dans le cas des piétons, les objets ne bougent pas indépendamment les uns des autres. Ils vont naturellement se regrouper (e.g. un groupe d'amis), éviter les collisions (e.g. deux personnes marchant l'un vers l'autre sur un trottoir) et suivre des flux (e.g. une organisation de flux en sens opposés dans un couloir de métro).

C'est pourquoi des modèles, dits sociaux, ont été développés [52] pour prédire les trajectoires des piétons. Ils reposent sur des hypothèses simplificatrices comme l'omniscience (chaque piéton connaît la position et la vitesse à l'instant t des autres piétons), des vitesses uniformes et des mouvements fluides (une pénalité est accordée sur les trajectoires à forte variation). Chaque piéton souhaite éviter les collisions et être à une distance raisonnable d'inconnus, à travers une modélisation de forces d'attraction et de répulsion.

D'autres algorithmes ont été entraînés pour tenir compte de la présence des autres piétons en exploitant des LSTM en partageant les états cachés d'un piéton avec ses voisins les plus proches [53] ou avec des réseaux neuronaux de graphes (*graph neural network*, GNN) [54].

Estimation du flux optique Toutes les méthodes précédemment discutées estiment le mouvement d'un objet à partir des observations passées uniquement. Or, il est possible d'ex-

ploiter une information à l'instant t pour estimer la position des objets à cet instant grâce à l'estimation du flux optique. Cela consiste à trouver une correspondance entre tous les pixels de l'instant t et ceux de l'instant $t+1$. Une fois le flux estimé, il suffit par la suite de déformer la localisation de l'instant t à travers le flux (*warping*) pour estimer la nouvelle position à l'instant $t+1$. En général, cela est appliqué sur des masques binaires et non sur des boîtes englobantes, car l'arrière-plan peut affecter la qualité de la nouvelle position.

Le flux optique peut être estimé empiriquement entre deux images I_t et I_{t+1} en trouvant les déplacements $(\Delta x, \Delta y)$ (*disparity vector*) pour chaque pixel (x, y) de sorte à minimiser les différences de couleurs entre $I_t(x, y)$ et $I_{t+1}(x + \Delta x, y + \Delta y)$ [55]. Il est également possible d'entraîner un réseau de neurones convolutif en lui fournissant des données artificielles, générées par ordinateur, où le flux optique est à estimer à partir d'une paire d'images adjacentes [56, 57]. Ainsi, FlowNet [56] estime directement la carte du flux optique à travers un réseau de neurones convolutif alors que son successeur FlowNet2 [57] décompose cette tâche en plusieurs sous-tâches qui sont résolues par des CNN spécialisés, en estimant d'abord les larges mouvements puis les petits. Cela permet notamment d'améliorer la qualité de l'estimation, en particulier sur les petits déplacements et sur des vidéos réelles, sans compromis sur le temps de calcul. RAFT [58] au contraire propose d'estimer le flux optique en réduisant les a priori sur les mouvements et en travaillant sur une unique carte de représentation en haute résolution, améliorant ainsi les prédictions du flux optique.

Il est à noter que la tâche d'estimation du flux optique est très similaire à celle d'estimation de la profondeur (*stereo matching*) : dans les deux cas, les entrées sont des paires d'images et la sortie est une carte de correspondance. La différence réside sur la nature des paires : pour le flux optique, il s'agit des images à deux instants I_t et I_{t+1} alors que pour l'estimation de la profondeur, il s'agit de deux images du même instant issues de deux caméras. En connaissant la distance séparant les caméras et leur focale, il est possible de générer des données annotées à partir de véritables vidéos. Ainsi, il est possible de limiter l'écart de domaine (*domain gap*) entre les données en entraînant un modèle de flux optique sur des données initialement collectées pour l'estimation de la profondeur.

Pour conclure cette section portée sur l'estimation du mouvement, les travaux de Dendorfer et al. [59] peuvent être cités, lesquels ont estimé que seules 10% des occlusions supérieures à trois secondes étaient correctement estimées. Ils ont proposé une approche à base d'une vue d'oiseau (*bird's-eye view*) en générant les positions des objets vues d'en haut par une méthode de recalage d'images et d'estimation de profondeur et par une prédiction des trajectoires futures. Mais l'ensemble de l'algorithme se révèle être très lent.

2.3.3 Affinité basée sur l'apparence visuelle

Outre la proximité spatiale et l'estimation des trajectoires, mesurées toutes les deux à partir des localisations, l'apparence des objets peut permettre d'effectuer l'association. Cette tâche est appelée la réidentification (reID). Elle consiste à déterminer si deux images circonscrites par des rectangles englobants centrés sur deux objets correspondent au même objet. Ainsi, tandis que la tâche de classification présentée en introduction répond à la question "quoi ?", la tâche de réidentification répond à la question "qui ?". L'objectif est ainsi d'extraire un vecteur de représentation pour chaque image de sorte que ces vecteurs soient similaires pour deux objets ayant la même identité, et inversement quand ce n'est pas le cas. Une fois les vecteurs de reID obtenus, leur affinité est généralement mesurée par leur similarité cosinus.

En général, les modèles de reID sont développés sur une catégorie d'objets précis (e.g. les personnes ou les véhicules). Ils sont alors entraînés sur des bases de données comportant des images de plusieurs objets, généralement issues de plusieurs caméras multipliant naturellement les angles de vue. Pour entraîner un modèle de reID, une fonction de perte par défaut est l'entropie croisée lorsque l'objectif est une classification binaire. Toutefois, le signal pour le calcul des gradients lors de la rétro-propagation n'est pas assez riche. Une manière de l'enrichir est d'utiliser une troisième image. Considérons une image ancre a, une seconde image correspondant au même objet p (pour positif) et une troisième image correspondant à un autre objet n (pour négatif). L'objectif est alors de rendre la représentation de a aussi similaire de celle de p, et aussi dissimilaire de celle de n. La fonction de perte par triplet (*triplet loss*) [60] pour un réseau f est définie de cette manière :

$$\mathcal{L}_{triplet}(a, p, n) = \max(m + ||f(a) - f(p)||^2 - ||f(a) - f(n)||^2, 0), \tag{2.6}$$

où m est un hyper-paramètre de marge pour empêcher d'avoir une valeur trop grande pour des triplets faciles. Plusieurs variantes existent, par exemple considérer plusieurs triplets provenant des mêmes objets pour multiplier les cas difficiles (*hard positive mining* et *hard negative mining*), ou encore remplacer la fonction max qui est trop abrupte par la fonction softplus $x \mapsto \ln(1 + \exp(x))$ [61]. Une autre manière de complexifier le signal provenant de la fonction de perte est de considérer quatre images au lieu de trois [62]. Dans ce cas, il y a deux négatifs qu'il faut également différencier en rendant leurs représentations dissimilaires. Ainsi, l'usage de la perte par quadruplet (*quadruplet loss*) donne des représentations avec une distance intra-classe (même objet) plus petite et une distance inter-classe (entre deux objets différents) plus grande.

Une particularité qui a été remarquée par certains chercheurs est l'influence de l'arrière-

plan, qui n'est pas pertinent pour la tâche de reID. Une manière de procéder est d'ajouter à l'image d'un objet son masque binaire en tant que quatrième canal [63]. Ainsi, l'ajout de ce quatrième canal a augmenté le pouvoir discriminant des vecteurs de représentation, surpassant la version à trois canaux RGB, qui elle-même a battu la version à trois canaux où un masque noir a été appliqué sur les pixels de l'arrière-plan. Cela est cohérent, car les masques fournissent non seulement une indication sur la forme des objets, mais aussi guident sur les zones d'intérêt. De telles indications sont également nécessaires dans le cas où une tierce personne cache une partie du corps du sujet : désormais, une attention particulière est portée sur la réidentification lors d'occlusions partielles [64]. Des solutions existent comme des algorithmes de reconnaissance basés sur des portions de corps ou le stockage en mémoire des apparences passées.

2.3.4 Affinité basée sur une approche hybride

Et enfin, ces différentes modalités peuvent être combinées pour renforcer les affinités. Une méthode simple est une moyenne pondérée des affinités calculées sur le mouvement et sur l'apparence [65]. Bien que surprenamment simple, elle reste très efficace dans le cadre d'un suivi d'objets. D'autres approches apprennent une représentation en combinant les informations de mouvement et d'apparence. TrackFormer [16] exploite le paradigme de suivi par attention en détectant et en associant de manière auto-régressive. À l'instant $t + 1$, les caractéristiques des objets à l'instant t sont mises à jour en tenant compte de leur nouvelle apparence et de leur nouvelle position. Cette formulation simple permet également d'estimer des caractéristiques de manière jointe, comme cela a été présenté dans la section 2.3.2.

2.4 Méthode d'assignation d'identité

Jusqu'à maintenant, plusieurs approches ont été présentées pour obtenir des matrices d'affinité (on parlera de matrice de coût lorsque la matrice contient des valeurs de dissimilarité et de similarité sinon). Illustrons l'importance de la méthode d'assignation d'identité par un exemple dans un cadre du suivi d'objets par détection et en ligne. La matrice de coût $C = (c_{ij})$ suivante indique les dissimilarités entre deux ensembles composés de trois tracks (les lignes) et de quatre détections (les colonnes) :

$$\begin{pmatrix} 0.1 & 0.8 & 0.9 & 0.5 \\ 0.7 & 0.2 & 0.7 & 0.4 \\ 0.7 & 0.3 & 0.8 & 0.9 \end{pmatrix} \tag{2.7}$$

La valeur c_{ij} mesure la dissimilarité entre les objets i et j. Pouvez-vous trouver une assignation entre les trois tracks et les quatre détections ? La formulation d'un tel problème, appelé problème de couplage parfait de poids minimum, pour une matrice $C \in \mathbb{R}^{M \times N}$ est la suivante :

$$\min \sum_{\substack{1 \leq i \leq M \\ 1 \leq j \leq N}} y_{ij} c_{ij}, \quad \text{tels que} \begin{cases} \sum_i y_{ij} = 1, \, \forall \, 1 \leq i \leq M \\ \sum_j y_{ij} = 1, \, \forall \, 1 \leq j \leq N \\ y_{ij} \in \{0, 1\}, \, \forall \, 1 \leq i \leq M, 1 \leq j \leq N \end{cases} \tag{2.8}$$

2.4.1 Algorithme glouton

Une première approche naïve consisterait à associer récursivement les paires objets (i, j) du plus similaires au moins similaires, en respectant une condition : un objet ne peut pas être associé à plus de deux objets [3]. Dans l'exemple présenté dans l'équation 2.7, on associerait ainsi les paires (1, 1) au coût de 0.1, puis (2, 2) et enfin (3, 3) au coût de 0.8. Est-ce une bonne solution ? Bien que rapide, elle reste perfectible.

2.4.2 Algorithme de Kuhn-Munkres

La solution proposée précédemment laisse à désirer. En effet, en associant d'abord la paire (2, 2), il n'est plus possible d'associer la paire (3, 2) alors que son coût est à peine supérieur. L'algorithme de Kuhn-Munkres [66] (aussi appelé algorithme hongrois) a été proposé en 1955 pour associer des paires pour un tel problème. Il s'agit d'une version relaxée du problème initial en un problème d'optimisation linéaire, exécutable en un temps polynomial.

La solution est la matrice de permutation $P^\star$ tel que :

$$P^\star \in \arg\min_P Tr(PC) \tag{2.9}$$

La solution naïve a un coût de 1.1 tandis qu'il existe une solution avec une trace de seulement 0.8 constituée des paires (1, 1), (2, 4) et (3, 2).

2.4.3 Algorithme glouton avec condition de non-recouvrement

Dans le cadre d'un algorithme de suivi hors ligne, il est préférable d'ignorer les cas les plus difficiles dans un premier temps. Ainsi, une option serait de créer des portions de trajectoires

3. Ceci est une hypothèse simplificatrice. Parfois, il est utile de la violer comme dans le cas d'une fragmentation d'un masque binaire

sous la forme de tracklets, puis de joindre les tracklets pour former des tracks pour résoudre les occlusions. Dans ce cas, un premier algorithme d'assignation en ligne peut être appliqué (comme l'algorithme hongrois), suivi d'une assignation hors ligne. Celle-ci s'opère sur des paires de tracklets et devra respecter comme condition un non-recouvrement entre les tracklets, c'est-à-dire que deux tracks de deux objets apparaissant simultanément dans au moins une trame ne peuvent pas être fusionnés.

2.4.4 Algorithme par regroupement de données

Également dans le cas d'une association hors ligne, un algorithme par regroupement de données (*clustering*) peut permettre de trouver des groupes (*clusters*) d'observations similaires. Cela a été proposé par Yang et al. [12] en utilisant un regroupement hiérarchique (*hierarchical clustering*) et leur algorithme a atteint la première place de la compétition de MOTS organisée à la conférence CVPR de 2020.

2.5 Conception des algorithmes de suivi classiques

Désormais, toutes les notions de base relatives au suivi d'objets ont été présentées. En les combinant, nous pouvons créer des algorithmes de suivi. Avant de présenter leurs représentants les plus populaires, quelques stratégies particulières aux algorithmes de suivi seront présentées.

2.5.1 Stratégies particulières au suivi multi-objets

Ces stratégies portent sur des détails d'implémentation, parfois simples, mais nécessaires pour obtenir des résultats comparables à l'état de l'art.

Réactivation de tracks

Dans le cadre d'un suivi en ligne, si un objet n'est associé avec aucune observation, certains algorithmes de suivi le conservent tout de même en mémoire pour de futures associations : c'est la stratégie de réactivation de tracks [15] (*track rebirth*). L'objet est finalement oublié si aucune nouvelle association n'est faite après un laps de temps. À un instant donné, on parle de track actif lorsqu'une association a été faite à la trame précédente et inactif autrement. L'âge d'un track est la durée durant laquelle le track a été inactif [4]. Cette stratégie permet de générer des tracks au lieu de tracklets, résolvant certains cas de courtes occlusions.

4. parfois, l'âge désigne la durée d'un track [37]

Filtrage des tracks

Parmi les post-traitements, l'étape de filtrage consiste à ne conserver que des tracks ayant dépassé un seuil de confiance, basé généralement sur la moyenne ou le maximum du score de confiance des détections le composant. Cette stratégie permet de supprimer des tracks correspondant couramment à des faux positifs.

Interpolation des tracks

Un autre post-traitement porte sur l'interpolation des positions inobservées dans les tracks. En effet, certaines bases de données requièrent la présence de boîtes englobantes même lors d'occlusion (partielle ou totale). Cela est généralement effectué par une interpolation linéaire des coordonnées. Cette stratégie permet de combler les détections manquantes d'un track.

Série de traitements

Lors de l'association des données, certaines priorités sont données à des tracks en fonction de leur âge et de leur durée, ou à des détections en fonction de leur score de confiance. Ainsi SORT [20] propose une série de traitement (*pipeline*) simple, Deep SORT [37] une association en cascade en privilégiant d'abord les associations avec les tracks les plus longs, ByteTrack [67] en privilégiant les détections avec les plus hauts scores de confiance ou GHOST [65] en différenciant les tracks selon leur âge.

Dans le cadre d'un suivi hors ligne, des tracklets sont formées sur des trames adjacentes puis, ceux-ci sont ensuite regroupés pour former des tracks. Cette approche différenciée entre les cas simples et difficiles est également transposable dans le cadre d'un suivi en ligne [65] en fixant des seuils d'acceptation différents selon qu'un track soit actif ou inactif.

Seuils

De nombreux seuils doivent être sélectionnés dans un algorithme de suivi. Sans vouloir être exhaustif, on peut citer un seuil sur le score de confiance minimal sur les détections, un seuil minimal de similarité entre deux instances lors de l'association, un seuil sur la durée maximal autorisée lors de la réactivation de tracks et un seuil sur lors du filtrage des tracks.

2.5.2 Algorithmes de suivi

Le tableau 2.2 décrit les algorithmes de suivi les plus populaires pour la tâche de MOT à partir des composantes utilisées dans cette thèse pour décrire les algorithmes de suivi.

TABLEAU 2.2 Descriptions des algorithmes de suivi MOT les plus populaires

Algorithme	Année	Paradigme	En ligne	Localisation	Affinité	Assignation	Autres
SORT [20]	2016	par détection	oui	boîtes avec Faster R-CNN	filtre de Kalman ou rectiligne uniforme + IoU	algorithme hongrois	pas de réactivation de tracks
Deep SORT [37]	2017	par détection	oui	boîtes avec Faster R-CNN	Kalman + Mahalanobis, ReID + cosine, moyenne des signaux	algorithme hongrois	réactivation de tracks, mémoire visuelle, privilège sur la longueur d'un track
IoU-Tracker [68]	2017	par détection	oui	boîtes avec EB	IoU	algorithme glouton	filtre temporel et de score
VIoU-Tracker [69]	2018	par détection et par régression	non	boîtes	IoU puis visuellement avec un filtre KCF	glouton	
Tracktor [15]	2019	par régression	oui	boîtes	Faster R-CNN, reID, modèle de mouvement	aucune	
Neural Message Passing [70]	2020	par détection	non	boîtes	GNN avec reID	flot de coût minimum	
CenterTrack [19]	2020	par régression	oui	boîtes et points	-	glouton	
TransTrack [71]	2020	par attention	oui	boîtes	Transformer + IoU	par attention	réactivation de tracks
TrackFormer [16]	2021	par attention	oui	boîtes	Transformer + ReID	par attention	ne peut pas suivre plus de 100 objets, track NMS
MOTR [72]	2021	par attention	oui	boîtes	Transformer	par attention	
ByteTrack [67]	2021	par détection	oui	boîtes	Filtre de Kalman + IoU	algorithme hongrois	réactivation de tracks, cascade suivant le score de détection
OC-SORT [48]	2023	par détection	oui	boîtes	Filtre de Kalman modifié + IoU	algorithme hongrois	interpolation linéaire optionnelle
C-BIoU [42]	2023	par détection	oui	boîtes	BIoU	algorithme hongrois	association par cascade
GHOST [65]	2023	par détection	oui	boîtes	moyenne entre un modèle linéaire de mouvement et ReID	algorithme hongrois	différenciation à partir de l'âge des tracks

2.6 Jeux de données et évaluation

Cette section traite des bases de données les plus courantes pour les tâches de MOT et MOTS ainsi que les mesures de performances qui leur sont associées.

2.6.1 Bases de données

Compte tenu des domaines d'application du suivi d'objets, la plupart des bases de données de suivi portent sur des personnes et/ou sur des véhicules.

En 2012, l'équipe du professeur Geiger a proposé la base de données KITTI [21] pour les tâches de suivi d'objets dans le cadre de la conduite autonome. La caméra est ainsi fixée à un véhicule en déplacement sur 50 séquences, où l'objectif est de suivre les piétons et les véhicules.

L'équipe de la professeure Leal-Taixé a proposé le MOTChallenge qui consiste à suivre uniquement des piétons dans des vidéos. En 2015, la première version (MOT15) [7] est composée de 22 séquences dont la moitié est utilisée lors de l'évaluation en test. Outre la vérité terrain, les séquences disposent aussi de détections issues de détecteurs pour permettre une comparaison plus juste des méthodes d'association. Les caméras sont parfois fixes ou mobiles, avec des points de vue et des conditions météorologiques variées et la densité y est d'environ de 9 piétons par trame. En 2016, la seconde version (MOT16) [73] présente une nouvelle base de données avec de nouvelles séquences plus compliquées : la densité augmente en atteignant 19 personnes par trame. Cependant, l'annotation étant fastidieuse, les séquences sont au nombre de 14 pour un total de moins de huit minutes. L'année suivante, une mise à jour (MOT17) est proposée en corrigeant certaines annotations et en fournissant trois détections (dites publiques) de qualité variée pour mesurer la robustesse des algorithmes de suivi au choix d'un détecteur. Puis en 2020, une nouvelle base de données (MOT20) [74] est présentée pour adresser les cas où la densité est très haute : sur les huit séquences, on compte en moyenne 123 piétons par trame. Et enfin, pour obtenir de très grandes bases de données à moindre coût, le moteur graphique du jeu-vidéo GTA a été exploité pour générer 768 clips, avec plus de 40 millions de boîtes englobantes, avec en moyenne 30 piétons par trame. Outre les annotations à l'aide de boîtes englobantes, ce jeu de données (MOTSynth) [75] fournit également des annotations au niveau de la pose, des boîtes 3D, de la segmentation et de la profondeur.

Quant à la tâche de MOTS, elle a été introduite en 2019 [76] avec les bases de données KITTIMOTS et MOTSChallenge. Les annotations à l'échelle du pixel ont été faites par la correction manuelle de masques obtenues par des algorithmes à base de CNN. KITTIMOTS

est basée sur la base de données KITTI et MOTSChallenge sur celle de MOT16/MOT17.

Sun et al [8] ont fait remarquer en 2021 que les précédentes bases de données souffraient de plusieurs biais. Tout d'abord, l'apparence physique des objets à suivre était suffisamment différente de sorte qu'un algorithme à base de réidentification pouvait obtenir de bons résultats sans exploiter d'affinité à base de mouvement. De plus, ces mouvements étaient très souvent linéaires dans les anciennes bases de données. Ainsi, les auteurs ont proposé la base de données DanceTrack avec 100 séquences sur des personnes présentant des similarités sur l'apparence et des mouvements non linéaires. En analysant les performances de plusieurs algorithmes de suivi sur celle-ci et MOT17, ils ont montré que ce nouveau jeu de données présentait bien plus de difficulté sur la sous-tâche d'association et non celle de détection. Des statistiques sur ces bases de données sont présentées dans l'annexe A.

D'autres bases de données existent pour des cas très spécifiques, comme le suivi de véhicules depuis des caméras de vidéo-surveillance avec UA-DETRAC [77] ou depuis des drones avec UAVDT [78].

Il est également important de préciser quelques points à propos des annotations lorsqu'elles sont faites à l'aide de boites englobantes. Celles-ci peuvent englober soit uniquement la partie visible, soit l'entièreté des objets, et ce même en cas d'occlusion partielle ou lorsque les objets se retrouvent en bordure de l'image. Ainsi, dans MOT17, il faut détecter l'entièreté des piétons alors que pour DanceTrack, il ne faut détecter que les parties visibles. Puis, certaines annotations peuvent être présentes dans le cas d'une occlusion totale. Dans ce cas, il faut alors interpoler ces positions dans le cas d'un suivi par détection. Cela se produit pour MOT17 et non pour DanceTrack.

De plus, lorsque le suivi concerne des personnes, il est primordial d'éviter les faux positifs engendrés par des objets ressemblant à des personnes (comme les réflections dans des miroirs, des parasols, des affiches de personne, des personnes debout dans un tramway, des cyclistes, des personnes assises, etc). Il faut alors fournir ces précisions lors de la création d'une base de données pour éviter ces ambiguïtés. Parfois, les auteurs fournissent des zones à ignorer pour ne pas que de telles détections n'influencent la mesure de performance.

2.6.2 Mesures de performance d'un algorithme de suivi

Contrairement à la tâche de classification, la tâche de suivi multi-objets fait intervenir des composantes de détection et d'association, c'est-à-dire des caractéristiques spatiales et temporelles. Ainsi, la formulation d'une mesure de performance est difficile et a été sujette à changement au cours des années.

Erreurs fondamentales

Tout d'abord, on compte trois types d'erreurs fondamentales à un instant t :

1. détecter un objet qui n'est pas un objet d'intérêt (e.g. un arbre, un mannequin, un reflet dans une vitre) : c'est un faux positif (FP pour *false positive*) ;
2. ne pas détecter un objet d'intérêt (e.g. une personne qui se confond avec l'arrière-plan, une personne très éloignée de la caméra) : c'est un faux négatif (FN pour *false negative*) ;
3. assigner une mauvaise identité à une détection : c'est un changement d'identité (IDSw pour *identity switch*, ou encore appelé *mismatch error*).

À une trame donnée, l'IoU est calculée entre toutes les détections et la vérité terrain à partir de l'équation 2.3 puis l'association entre les détections et la vérité terrain est faite avec l'algorithme hongrois. Et seules les paires (détection, vérité terrain) avec une IoU supérieure à un seuil α, généralement 0.5, sont conservées.

Dans ce cas, à une trame donnée, un FP est une détection qui n'a été associée à aucune vérité terrain, et un FN est une vérité terrain qui n'a été associée à aucune détection. Quant à un IDSw, il se produit lorsque la même vérité terrain se voit attribuer à des détections ayant des identités différentes à des instants successifs.

Lorsqu'une détection est associée à une vérité terrain, on parle d'un vrai positif (TP pour *true positive*).

MOTA et IDF_1

Ces trois erreurs fondamentales permettent de définir la mesure MOTA (*Multiple Object Tracking Accuracy*

$$MOTA = 1 - \frac{\sum_t FP_t + FN_t + IDSw_t}{\sum_t GT_t}, \tag{2.10}$$

où FP_t, FN_t, $IDSw_t$ et GT_t sont respectivement le nombre de faux positifs, le nombre de faux négatifs, le nombre d'*identity switch* et le nombre d'objets dans la vérité terrain à la trame t. Plus le MOTA est haut (proche de 1), plus le résultat de l'algorithme de suivi est similaire à la vérité terrain. La mesure MOTA est simple à calculer à partir des erreurs commises à chaque trame.

Cette mesure a été complétée par la mesure IDF_1 [79] (*Identification* F_1) qui évalue davantage

la qualité d'association. Elle est calculée en trouvant une bijection, avec l'algorithme hongrois, entre les tracks prédits et les véritables tracks en obtenant d'abord la véritable identité de chaque détection puis en minimisant le nombre de mauvaises assignations d'identité entre tracks. Puis la mesure IDF_1 est calculée à partir des tracks où une correspondance a été trouvée (IDTP), des tracks prédits sans correspondance (IDFP) et des véritables tracks sans correspondance (IDFN) [5] :

$$IDF_1 = \frac{|IDTP|}{|IDTP| + 0.5 \times (|IDFN| + |IDFP|)} \tag{2.11}$$

HOTA, DetA et AssA

Toutefois, cette simplicité de la mesure MOTA cache un biais : elle mesure principalement la qualité de la détection en ignorant la qualité de l'association. Par exemple pour l'algorithme de suivi C-BIoU sur MOT17 [6], on a 30315 FP, 65712 FN et 1194 IDSw. La mesure MOTA pondère ainsi fortement les deux premières erreurs au détriment de la troisième.

Luiten et al. [11] ont montré que la mesure MOTA était linéairement corrélée à la mesure de la qualité de la détection, avec un R^2 de 99% sur 175 algorithmes de suivi. Ils ont ainsi proposé la mesure HOTA (*Higher Order Tracking Accuracy*) qui est plus équilibrée et qui repose sur une décomposition en sous-mesures sur la détection, la DetA (*Detection Accuracy*), et sur l'association, la AssA (*Association Accuracy*).

Plus précisément, au lieu de définir les TP, FP et FN à chaque trame en n'utilisant qu'un seul seuil α à 0.50, ils sont définis sur plusieurs seuils avec TP_α, FP_α et FN_α. Cela fournit la mesure de la qualité de la détection au seuil α, selon l'équation 2.12 :

$$DetA_\alpha = \frac{|TP_\alpha|}{|TP_\alpha| + |FN_\alpha| + |FP_\alpha|} \tag{2.12}$$

De même, les auteurs ont introduits les notions de TPA (*True Positive Association*), FPA (*False Positive Association*) et FNA (*False Negative Association*) qui généralisent les trois notions précédentes au cas des associations. La figure 2.4 illustre l'obtention de ces trois ensembles. Pour un seuil d'association α, étant donnée une détection d correspondant à une localisation d'un objet A, $TPA_\alpha(d)$ est l'ensemble des détections du track de d qui correspondent aussi à des localisations de l'objet A ; $FNA_\alpha(d)$ est l'ensemble des localisations de l'objet A qui ne sont pas dans le track de d ; et $FPA_\alpha(d)$ est l'ensemble des détections du

5. il s'agit du score F_1 où la précision est donnée par $\frac{|IDTP|}{|IDTP|+|IDFP|}$ et le rappel par $\frac{|IDTP|}{|IDTP|+|IDFN|}$

6. Les résultats sur la base de test de MOT17 à partir des détections privées sont disponibles sur le site web `https://motchallenge.net/results/MOT17/?det=Private`

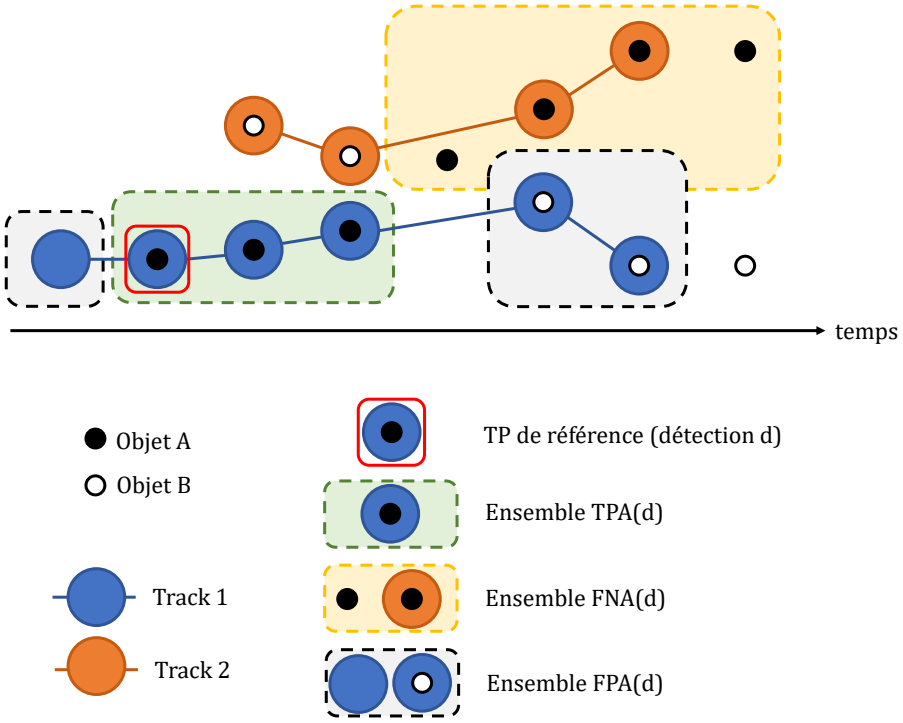


FIGURE 2.4 Illustration de la création des ensembles TPA, FNA et FPA. Deux véritables objets sont présents (● et ○) et deux tracks ont été prédits (et). À chaque trame, les détections sont appariées à des vérités terrain pour distinguer les TP des FP. Puis, étant donnée une détection correspondant à un TP, notée d (encadré en rouge), $TPA(d)$ est composé des éléments qui partagent la même véritable identité et la même identité prédite que d ( ). $FNA(d)$ est composé des éléments où la véritable identité est la même que celle de d mais dont l'identité prédite est différente ( ) et des faux négatifs du véritable objet (●). Ensuite, $FPA(d)$ est composé des éléments du track de d où la véritable identité est différente de celle de d ( ) et des faux positifs du track de d ( ). Inspiré de Luiten et al [11].

track de d qui ne correspondent à aucune localisation de l'objet A. Cela donne la mesure de la qualité de l'association au seuil α, selon l'équation 2.13 :

$$AssA_\alpha = \frac{1}{|TP_\alpha|} \sum_{d \in \{TP_\alpha\}} \frac{|TPA_\alpha(d)|}{|TPA_\alpha(d)| + |FNA_\alpha(d)| + |FPA_\alpha(d)|} \tag{2.13}$$

Et enfin, la mesure HOTA au seuil α est la moyenne géométrique des deux sous-mesures précédentes, suivant l'équation 2.14. Pour finir, la mesure HOTA est calculée avec l'équation 2.15 en tant qu'une moyenne arithmétique sur 19 valeurs de seuil α. Une formulation identique permet d'obtenir les sous-mesures DetA et AssA.

$$HOTA_\alpha = \sqrt{DetA_\alpha \times AssA_\alpha} \tag{2.14}$$

$$HOTA = \frac{1}{19} \sum_{\alpha \in \{0.05, 0.10, ..., 0.90, 0.95\}} HOTA_\alpha \tag{2.15}$$

Outre une meilleure prise en compte de la qualité des détections via différents seuils α et un meilleur équilibrage entre la qualité de la détection et de l'association, la mesure HOTA montre un meilleur alignement avec l'évaluation visuelle humaine. C'est pourquoi elle a supplanté en 2021 la traditionnelle mesure MOTA sur les bases de données du MOTChallenge et de KITTIMOT/KITTIMOTS.

2.7 Autre notion pertinente pour la thèse : l'apprentissage par contraste

Une généralisation de la perte par triplet de l'équation 2.6 à plusieurs paires négatives est la perte par contraste.

La fonction de perte par contraste pour l'élément z_i est calculée de la manière suivante :

$$\mathcal{L}_{contrastive}(z_i, \mathcal{Z}) = - \sum_{(i,j) \in \mathbb{P}} \log \left(\frac{e^{(z_i \cdot z_j)/\tau}}{\sum_{k=1}^{N} e^{(z_i \cdot z_k)/\tau}} \right), \tag{2.16}$$

où $\mathcal{Z} = (z_k)_{1 \leq k \leq N}$ est un ensemble de N représentations, τ un hyper-paramètre de température et $\mathbb{P}$ l'ensemble des paires positives. L'objectif est de maximiser le produit scalaire entre deux représentations d'une paire positive par rapport aux produits scalaires des autres paires. Cette formulation a été utilisée par exemple pour la vérification de signatures [80] en 1993 ou pour discriminer un signal d'un bruit généré artificiel [81] en 2010.

Plus récemment, cette fonction de perte a été au centre de l'apprentissage auto-supervisé (*self-supervised learning*). Il s'agit d'une branche de l'apprentissage non supervisé, c'est-à-dire, lorsqu'aucune étiquette y ne peut être exploitée pour entraîner un modèle. Dans ce cas, les chercheurs développent en amont un algorithme qui va créer artificiellement (et non par un annotateur humain) une étiquette y : cela peut par exemple être la localisation d'un morceau d'une image [82], la couleur d'une image en ton de gris [83] ou encore la rotation d'une image [84]. L'apprentissage de cette tâche prétexte (*pretext task*) permet l'estimation des paramètres du modèle. Ensuite, il suffit de régler finement les poids de ce modèle (soit l'ensemble, soit ceux d'un classifieur) sur une base de données annotées par des humains.

L'intuition est que l'apprentissage d'une telle tâche prétexte est une condition préliminaire à la compréhension sémantique d'une image.

Au lieu de définir des tâches prétextes ad hoc, l'apprentissage auto-supervisé avec la perte par contraste consiste à trouver une invariance entre plusieurs représentations. Ainsi, de nombreuses méthodes telles que SimCLR [85], MoCo [86] ou CLIP [87] reposent sur la capacité à discriminer des paires (image-image ou image-texte) positives de paires négatives.

La perte par contraste est de fait dissociée de l'apprentissage auto-supervisé : il est possible d'utiliser des annotations humaines avec la fonction de perte par contraste. Par exemple, pour la détection d'objets dans des vidéos, Quasi-Dense [88] utilise cette fonction pour apparier des détections des instants t et $t + 1$.

CHAPITRE 3 DÉMARCHE

Cette thèse suit une structure de thèse par articles pour les trois prochains chapitres. Chaque chapitre est soit un article publié en conférence ou soumis dans une revue. Les articles répondent aux objectifs fixés, présentent des analyses et des solutions pour la tâche de suivi d'objets MOT ou MOTS quant à l'association de données.

Le chapitre 4 présente une analyse sur l'association de boîtes englobantes à partir des caractéristiques visuelles. Puis, le chapitre 5 propose une solution de suivi MOTS à partir d'une méthode générative de cartes de chaleur et le chapitre 6 une solution de suivi MOT à partir d'une méthode purement discriminante à partir des coordonnées spatio-temporelles de boîtes englobantes.

L'étude de l'état de l'art a montré que le suivi multi-objets apportait peu d'attention à la sous-tâche d'association. Celle-ci était mal prise en compte dans la mesure de performance [11] et était biaisée en raison des bases de données [8] qui mettaient l'accent sur des mouvements linéaires et des apparences similaires. Mathématiquement, soient deux ensembles $\mathcal{S}_1 = \{a_1, \cdots, a_M\}$ et $\mathcal{S}_2 = \{b_1, \cdots, b_N\}$ de détections (ou de tracklets) de taille respective M et N. L'objectif de la sous-tâche d'association est de trouver des paires (i, j) telles que les détections a_i et b_j correspondent au même objet. Ces paires peuvent être obtenues en utilisant l'algorithme hongrois sur une matrice de coût (ou d'affinité) $C \in \mathbb{R}^{M \times N}$. Tester différentes méthodes pour construire la matrice C afin d'améliorer le suivi est un des objectifs de cette thèse.

3.1 Évaluation des descripteurs visuels

Deux grandes approches existent pour la sous-tâche d'association : celles basées sur la position et/ou le mouvement et celles basées sur l'apparence. Par exemple, Deep SORT [37] combine ces deux approches en employant des vecteurs de réidentification et la similarité cosinus pour associer des détections. Or, ces vecteurs ont été obtenus à partir d'un entraînement sur des bases de données de réidentification, qui ne tiennent pas compte du caractère séquentiel de la tâche de suivi et de la présence des autres objets.

L'article présenté dans le chapitre 4 propose une méthodologie pour mesurer la capacité de discrimination des descripteurs visuels. La première contribution est une étude comparative entre neufs descripteurs visuels associés à cinq mesures d'affinité. Appliquée à quatre bases de données, elle montre que les vecteurs de réidentification sont les descripteurs visuels qui

sont en moyenne les plus discriminants. La seconde contribution est une évaluation de la robustesse de la mesure d'affinité lorsque deux objets sont temporellement distants et lorsque les coordonnées des boîtes englobantes sont perturbées. Et la troisième contribution est une analyse sur les performances d'association par rapport à la taille des objets.

Mathématiquement, dans le chapitre 4, la matrice de coût C est calculée sans mémoire (il s'agit d'une comparaison entre détections et non entre tracklets), en n'exploitant que l'information visuelle présente dans les détections (et non leur position spatio-temporelle) et de manière indépendante : $C = (c_{ij})$ où $c_{ij} = s(h(a_i), h(b_j))$ où s est une mesure d'affinité et h un descripteur visuel.

3.2 Méthode d'association générative à l'échelle du pixel pour la tâche de MOTS

Les travaux du chapitre 4 ont été portés sur des vecteurs de description : ils décrivent un objet localisé par une boîte englobante indépendamment des autres objets. Les résultats ont montré que les vecteurs de réidentification sont ceux qui ont la meilleure capacité de discrimination dans le cadre d'un suivi multi-objets. Une limite de ces vecteurs est leur sensibilité lors d'une occlusion : si un objet secondaire est présent dans la boîte englobante, la représentation vectorielle de l'objet principal est fortement perturbée. C'est pourquoi une approche non basée sur des boîtes englobantes a été mise en œuvre.

L'article présenté dans le chapitre 5 propose un algorithme de suivi MOTS. Pour ne pas faire reposer le module d'association à long terme sur des descripteurs visuels, il a été nécessaire de développer une méthode qui exploitait les masques avec une approche générative. C'est pourquoi une méthode initialement développée dans le cadre de suivi de masques a été exploitée. Celle-ci repose sur les réseaux à mémoire spatio-temporelle [1] (STM) capables de retrouver dans une nouvelle image la localisation d'un masque présent dans une première image. Les réseaux STM peuvent aussi être employés sur plusieurs trames de sorte à considérer le caractère dynamique du suivi en mettant à jour la mémoire spatio-temporelle. La localisation prédite prend la forme d'une carte de chaleur qui est ensuite comparée à chaque masque binaire des objets présents dans la nouvelle image. Notre algorithme de suivi, nommé MeNToS, exploite ensuite ce module d'association basé sur des masques et des cartes de chaleur.

Mathématiquement, dans le chapitre 5, la matrice de coût C est calculée avec mémoire, en exploitant simultanément l'information visuelle à l'échelle du pixel et l'information de position, sans descripteur visuel : $C = (c_{ij})$ où $c_{ij} = s(a_i, b_j \,|\, \mathcal{S}_1 \cup \mathcal{S}_2)$ où s est une mesure d'affinité.

3.3 Méthode d'association discriminante non visuelle exploitant le contexte pour la tâche de MOT

En étudiant les capacités d'association du module développé précédemment, il s'est avéré que le travail à l'échelle du pixel, et ce sur toute l'image, rendait le suivi d'objets assez lent et nécessitait davantage de ressources en mémoire. De plus, des expériences consistant à supprimer l'arrière-plan montraient curieusement une baisse de la capacité à discriminer des paires de tracklets. Ainsi, le contexte (arrière-plan et la présence des autres objets) aidait le module de MeNToS à mesurer la similarité d'une paire.

Ces résultats contre-intuitifs ont poussé à l'élaboration d'une méthode reposant à la fois sur les coordonnées spatio-temporelles pour être plus rapide, et sur une approche tenant compte du contexte. Ainsi, l'article présenté dans le chapitre 6 propose un algorithme de suivi multi-objets, nommé C-TWiX, basé sur des réseaux Transformers, qui s'adaptent à la fois à des données *séquentielles* et à des *ensembles* de données. L'algorithme C-TWiX repose sur le module TWiX qui retourne une matrice d'affinité pour des paires de tracklets présents dans des fenêtres temporelles.

Mathématiquement, dans le chapitre 6, la matrice de coût C est calculée avec mémoire, en exploitant l'information de position et sans descripteur visuel : $C = (c_{ij})$ où $c_{ij} = s(a_i, b_j \,|\, \mathcal{S}_1 \cup \mathcal{S}_2)$ où s est une mesure d'affinité.

CHAPITRE 4 ARTICLE 1 : AN EMPIRICAL ANALYSIS OF VISUAL FEATURES FOR MULTIPLE OBJECT TRACKING IN URBAN SCENES

Miah M., Pepin J., Saunier N., Bilodeau G.-A.
25th International Conference on Pattern Recognition (ICPR), 2021, p. 5595-5602
Published on June 21, 2020
DOI : `https://doi.org/10.1109/ICPR48806.2021.9412206`

Nature de la contribution de Mehdi Naim Miah à l'article, telle que soumise à la conférence : optimisation du code informatique, évaluation des méthodes sur quatre bases de données (WildTrack, MOT17, DETRAC, UAVDT) au lieu d'une seule (UrbanTracker), ajout d'une expérience avec des données bruitées, rédaction de l'article (hormis première écriture de l'introduction et de la revue de la littérature), mise à jour de l'introduction et de la revue de littérature, vérification de la reproductibilité, visualisation des résultats et analyse des résultats

Abstract

This paper addresses the problem of selecting appearance features for multiple object tracking (MOT) in urban scenes. Over the years, a large number of features has been used for MOT. However, it is not clear whether some of them are better than others. Commonly used features are color histograms, histograms of oriented gradients, deep features from convolutional neural networks and re-identification (ReID) features. In this study, we assess how good these features are at discriminating objects enclosed by a bounding box in urban scene tracking scenarios. Several affinity measures, namely the L_1, L_2 and the Bhattacharyya distances, Rank-1 counts and the cosine similarity, are also assessed for their impact on the discriminative power of the features. Results on several datasets show that features from ReID networks are the best for discriminating instances from one another regardless of the quality of the detector. If a ReID model is not available, color histograms may be selected if the detector has a good recall and there are few occlusions ; otherwise, deep features are more robust to detectors with lower recall.

4.1 Introduction

Cities are faced with many challenges, including how to move people safely and efficiently for their daily activities. Data on the movement of all road users is therefore necessary. Such data can be collected automatically through various kinds of sensors, including video cameras

with computer vision algorithms. The main task is to detect and track all roads users, which is also called multiple object tracking (MOT). This is one example, among many, of the use of MOT.

Many state-of-the-art MOT methods rely on the strategy called "tracking-by-detection" [89] : first, they detect objects of interest, such as vehicles or pedestrians, then they link the detections between frames to create trajectories. For the second step, various features are used : appearance, spatial information and motion [90]. Even if MOT is a well-studied problem [7, 91–93], there are still many unsolved challenges limiting the quality of the results. One of them is describing objects' appearance. It should be possible to distinguish every tracked object from the others, while at the same time considering that the appearance of an object might change over time because of a viewpoint change and illumination variations. Therefore, selecting the most discriminative features and finding a proper way to compare them become two key elements in the process of visual appearance modeling.

Should handcrafted features be used, or should the object appearance be learned ? Given the fact that in MOT, several aspects, such as appearance, spatial and motion information, are usually investigated at the same time, it is difficult to tell whether a method is better because of the chosen feature, or the data association method, or the method to predict where the object should be in the future.

Recently, Kornblith et al. [94] studied how models with high performance on ImageNet [2, 5] actually performed for classification on other datasets. The answer is comforting : the better the models are on ImageNet, the better they are on other datasets. However, can the same conclusion be drawn on a downstream task such as tracking ? Indeed, MOT encounters atypical challenges such as the need for instance classification, deformations, illumination changes, occlusions, blur, etc. Some of these challenges are absent from the ImageNet dataset. Moreover, the classification required in tracking is more fine-grained to distinguish all object instances : models trained on ImageNet succeed when they correctly classify persons as persons whereas for the MOT task, these persons must be discriminated from one another.

In this paper, we assess the performance of popular visual features to describe objects in MOT in various urban scene scenarios. These are among the most popular scenarios in MOT, and the focus of the most popular MOT datasets. Therefore, the objects of interest to describe are mostly pedestrians and various vehicles. To avoid interference from other MOT components, we only focus on the visual appearance description and comparison for image regions enclosed by bounding boxes (BB). No spatial or motion information are used in this paper. The results suggest that re-identification (ReID) features are the best visual features for MOT tasks. When these features are not available, deep features may be used and give

better performance than the color histogram when objects are further apart in time. The HOG features critically degrade when the detector provides imprecise BBs.

The main contributions of this paper are :

— a comparison of visual descriptors on four MOT datasets ;
— a new methodology to compare features for the MOT task ;
— an analysis of descriptors and affinity measures performance according to the size of objects, the precision of the bounding boxes and the elapsed time between observations.

4.2 Related works

A large variety of appearance features have been used for tracking. Some of them are briefly reviewed in this section.

One of the most popular features for MOT is the color histogram. Among others, color histograms were used in the work of [95–97]. In the work of Riahi et al. [95], color histograms are combined with other features such as optical flow and a sparse representation. Optical flow calculates the motion vector of pixels between two frames, while a sparse representation reconstructs an image region using templates from the image regions of a model object and trivial templates, which contain only one non-zero value. If the visual appearance of an object is different from the model object, the reconstruction will require many trivial templates. In the case of the work of Zhu et al. [96] and Sun et al. [97], color histograms are combined with histograms of oriented gradients (HOG).

While the color histogram focuses on the general color appearance of an object, HOG focuses on the texture of an object (spatial arrangement of the colors). Because HOG features are calculated using gradient magnitudes as weights, they often also capture the general shape of an object. Therefore an HOG feature can be seen both as a texture and a shape descriptor. The MOT methods presented in [96–98] for example rely on HOG. In the work of Heimbach et al. [98], HOG is used solely in combination with a Kalman filter to predict object position.

Many works use deep features as universal descriptors for MOT [99–103]. The object appearance is described with features from VGG-16 in both the work of Tang et al. [101] and Sadeghian et al. [100], while the work of Wang et al. [99] uses a two-layer custom Convolutional Neural Network (CNN). Class labels (e.g. car, pedestrian, bike) were also used recently as a coarse description of an object appearance [91]. As for [102], the authors worked with VGG-19 from which multiple outputs from different layers were extracted. Finally, recent works include ReID features [37, 62, 104, 105]. These features are computed by learning a

model to predict if two detections from two points of view are instances of the same object.

Surprisingly, we could only find one work that compared features for MOT [106]. Because it dates back to 1996, the features that were compared are the distance between the center of gravity, the size of the BBs, and correlation between object pixels.

4.3 Tested visual features, affinity measures and datasets

Figure 4.1 gives an overview of our visual feature evaluation strategy. Given a bounding box-enclosed object extracted from a frame, it is first described with a feature descriptor. Then, we select another frame where this object is present, describe all objects in this frame with the same feature descriptor, and compute an affinity measure to select the most similar object in term of visual appearance. In the following, we describe the visual features, affinity measures and datasets selected for our evaluation. Given the available MOT datasets, we assume that objects are either pedestrians or vehicles.

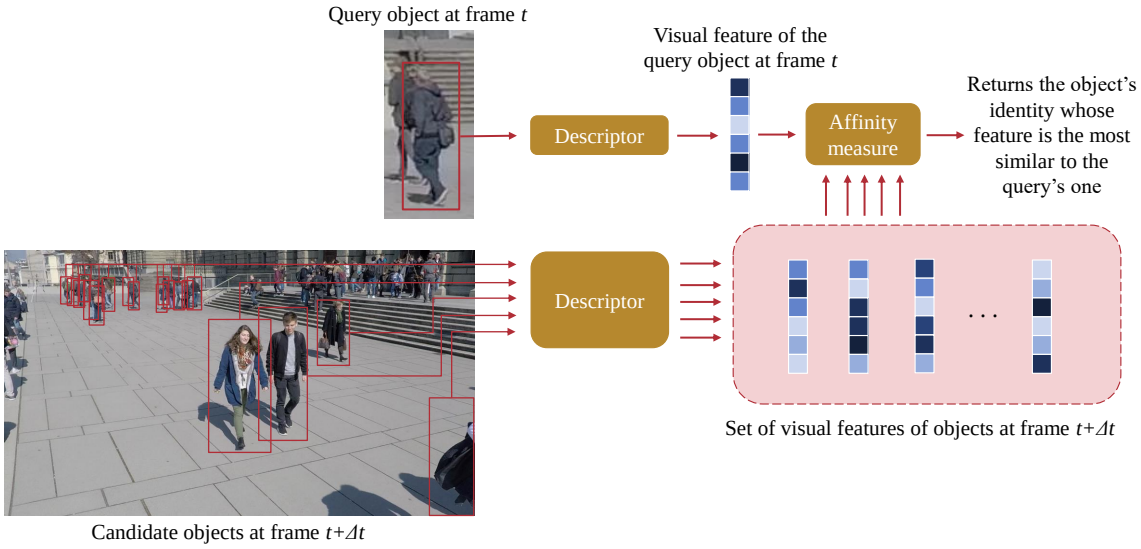


FIGURE 4.1 High-level explanation of our experimental methodology. From bounding boxes, a feature descriptor is calculated for both a query object and candidate matching objects in another frame. Then, the affinity measure is calculated for all query and candidate pairs, and the best match is returned and evaluated based on the ground truth.

4.3.1 Visual features

There are many ways to obtain a description (a numerical vector) from an image of an object enclosed by a BB. We selected eight popular visual descriptors among four categories : color

histograms-based, gradient-based, CNN-based and ReID-based models.

Color and grayscale histograms The RGB color histogram descriptor consists in counting the number of occurrences of each color inside a BB. We elected to use quantified histograms because they are more resilient to noise. For grayscale 1D-histogram, we used 32 bins. For color histogram, similarly, each channel of the image gives an histogram ; after concatenation, we obtain a vector of size 96.

Histograms of oriented gradients (HOG) Contrarily to color-based histograms, gradient-based descriptors are more robust to illumination change. HOG [107] is one famous example. It captures texture information in the image as well as shape information about the object. To obtain an HOG descriptor, the image is first convolved with kernels to extract vertical and horizontal gradients. From these gradients, their angles and magnitudes can be obtained. The angles are typically quantified between 0 and 180°, as it was shown experimentally that ignoring the sign of the angle gives better results. Histograms are then constructed for several overlapping cells by counting the occurrence of quantified angles weighted by their gradient magnitudes. Since in our experimental setup, the datasets contain either vehicles or pedestrians, the HOG vectors are the same size for all candidate objects in each dataset : the BB is resized to 64×128 pixels for pedestrians and 64×64 for vehicles, with cells of size 8×8 and blocks of size 16×16. By quantifying angles into nine bins every 20°, this results in a vector of 1764 elements for vehicles and 3528 for pedestrians.

CNN-based features Since their breakthrough in 2012 [3], CNNs are commonly used in computer vision for classification tasks. We used four different architectures which have great performance on ImageNet to extract visual features. For each network, we removed the last fully-connected layer to obtain a descriptor [108]. We evaluated VGG-19 [109], ResNet-18 [110], DenseNet-121 [111] and EfficientNet-B0 [112] architectures, respectively providing a description vector of size 4096, 512, 1024 and 1280, having respectively 140, 11, 7 and 4 millions parameters and resulting ImageNet top-1 error rates of respectively 27.6%, 30.2%, 25.4% and 23.4%.

Re-identification network In the case of pedestrian tracking, we evaluated a re-identification method named OSNet-AIN [105] containing 3 millions parameters giving ReID features of size 512. For vehicles, we extracted ReID features from the model of [113] containing 24 millions parameters providing vectors of size 2048.

4.3.2 Descriptor affinity measures

Each object BB in a frame is described with a numerical vector $\mathbf{x}$. So, given another image containing m object BBs, each of them described by a vector $\mathbf{y}_j$, the aim is now to find the "most similar" vector to $\mathbf{x}$ among $\{\mathbf{y}_j | j \in [\![1, m]\!]\}$ hoping that the two vectors are instances of the same object.

The following five different affinity measures were used to compare the visual feature vectors.

L_1 and L_2 distances two common ways to compute affinity between vectors are the L_1 and L_2 distances. Given two vectors $\mathbf{x}$ and $\mathbf{y}$, for $p \geq 1$, the L_p distance is given by

$$L_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}} \tag{4.1}$$

where i refers to the i^{th} element of each vector and n is the length of the vectors to compare. The smaller the distance is, the more similar the vectors are.

Rank-1 counts To compare feature vectors from a CNN, the Rank-1 counts was proposed in [114]. It was shown to be efficient to compare deep features. It works by comparing a pair of vectors $(\mathbf{x}, \mathbf{y})$ to other possible pairs $(\mathbf{x}, \mathbf{z})$ such that $\mathbf{z} \neq \mathbf{y}$. The underlying principle is to find the vector whose elements are closer to a query vector. It is computed as follows :

$$C_{\text{rank1}}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} 1\Big(|x_i - y_i| < \min_{\mathbf{z}, \mathbf{z} \neq \mathbf{y}} |x_i - z_i| \Big) \tag{4.2}$$

where 1 is an indicator function that takes value 1 if the expression in the argument is true and 0 otherwise. The expression verifies whether the i^{th} element of $\mathbf{x}$ is strictly closer to the corresponding element of $\mathbf{y}$ compared to all other candidate vectors $\mathbf{z}$. The larger the C_{rank1} is, the more similar the objects are.

Bhattacharyya distance This distance [115] measures the affinity between two distributions as follows :

$$D_B(\mathbf{x}, \mathbf{y}) = -\ln(\sqrt{\mathbf{x}}^\top \sqrt{\mathbf{y}}) \tag{4.3}$$

The smaller the D_B is, the more similar the objects are.

Cosine similarity For two vectors $\mathbf{x}$ and $\mathbf{y}$, the cosine similarity is given by :

$$S_C(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} \tag{4.4}$$

The smaller the S_C is, the less similar the objects are.

4.3.3 Datasets

We tested the visual features on four datasets commonly used in MOT. Two of them focus on pedestrians and two others on vehicles. Table 4.1 summarizes some of their characteristics.

Tableau 4.1 Dataset statistics : FPS : framerate, $\#S$: number of sequences, $\bar{F}$: average number of frames per sequence, $\bar{P}$: average number of pedestrians per frame, $\bar{V}$: average number of vehicles per frame and $\bar{S}$: average object size

Name	FPS	$\#S$	$\bar{F}$	$\bar{P}$	$\bar{V}$	$\bar{S}$
WildTrack [116]	2	7	400	15	0	85x286
MOT17 [7]	14/25/30	7	759	21	0	85x230
DETRAC [77]	25	60	1368	0	7	97x66
UAVDT [78]	30	50	808	0	20	39x34

4.4 Experimental methodology

To evaluate the performance of 35 descriptor-affinity pairs (each pair composed of a feature and an affinity measure except pairs between a non histogram-based feature and the Bhattacharyya distance), we tried to link two bounding boxes referring to the same object throughout a video. For that, given a BB-enclosed object extracted from a frame, we described it with a feature descriptor. Then we select another frame where this object is present, described all objects in this frame with the same feature descriptor, and compute an affinity measure to select the most similar object to the one in the first frame (Figure 4.1). We then verify if the match is correct based on the ground truth.

4.4.1 Data preparation

It should be noted that working with the true BBs is a too ideal scenario. In practice, when "tracking-by-detection" is applied, the detection algorithm may omit an object, detect non-object elements or the predicted BBs may be slightly shifted. In order to simulate the BBs returned by a detector, we introduced noise in two ways.

Noisy coordinates the first way is to add a white Gaussian noise to each coordinate independently. Given a BB (x_m, y_m, x_M, y_M) where (x_m, y_m) is the top-left coordinate of the BB and (x_M, y_M) the bottom-right one, noisy coordinates are obtained by sampling as follows :

$$\widetilde{x_m} \sim \mathcal{N}(x_m, (\sigma w)^2) \tag{4.5}$$

$$\widetilde{y_m} \sim \mathcal{N}(y_m, (\sigma h)^2), \tag{4.6}$$

where w and h are the BB width and the height. $\widetilde{x_M}$ and $\widetilde{y_M}$ are calculated similarly from x_M and y_M. The parameter σ allows to modify the variance of the Gaussian. Figure 4.2 illustrates the effect of σ on BB coordinates. By introducing noise in this way, it is still possible to get access to the true identity of each object, which is not possible if we used a detector. Indeed, a detector may miss some hard to detect objects, resulting in a biased analysis.

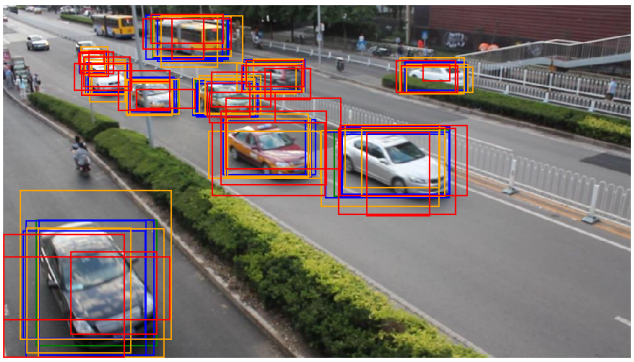


FIGURE 4.2 Examples of noisy bounding boxes on a frame of the DETRAC dataset : for each object of interest, three examples for each σ are displayed to illustrate the variability of noisy BBs. The color code is as follows : green for $\sigma = 0$ (the ground truth BB), blue for $\sigma = 0.05$, orange for $\sigma = 0.1$ and red for $\sigma = 0.2$. Best viewed in color.

Note that adding a Gaussian noise is not sufficient : we have to make sure that the new coordinates are valid (integers such that $0 \leq \widetilde{x_m} < \widetilde{x_M} <$ W and $0 \leq \widetilde{y_m} < \widetilde{y_M} <$ H, where W and H represent the width and the height of the frame, respectively). We chose the following σ parameters : 0, 0.05, 0.1 and 0.2.

Sampling step the second way to simulate noise is by skipping frames : instead of comparing two consecutive frames, we increase the sampling step. Therefore, the visual appearance of objects changes more and this simulates the case when the detector missed some objects or the object was not visible for several frames. We chose the following sampling steps : 1, 2, 4, 8, 16 and 32 frames. Note that this can result in different temporal skip in seconds depending

on the video frame rate. This should simply be viewed as gradually including more and more missing detections, making the matching more difficult.

4.4.2 Performance measure

For a given descriptor-affinity pair, a configuration σ-step and a sequence of a dataset, we evaluated the average precision for pairs (query object, set of candidate objects) by calculating the ratio of number of correct matches (when we find the same object among the set) over the total number of tested query objects. We reported the mean average precision over sequences on each dataset.

The configuration ($\sigma = 0$, step $= 1$) refers to the case where the detector can perfectly detect all objects.

4.4.3 Implementation details

For HOG, color and grayscale histograms, we used the implementations from OpenCV [117]. Models and weights for VGG-19, ResNet-18 and DenseNet-121 come directly from Pytorch [118]. As for EfficientNet-B0, we used the implementation provided by [119]. We relied on pretrained models learned on ImageNet. When using CNN-based models, as recommended by Pytorch, RGB BBs are resized to 224×224 and normalized. OSNet-AIN weights were pretrained by [105] on ImageNet and fine-tuned on Market1501 [120] and are available in the torchreid library [121]. Weights for vehicle ReID were trained by [113] on VeRi [122, 123], CompCars Surveillance [124], BoxCars [125] and unsupervisedly fine-tuned on AI City Challenge dataset [126].

4.5 Results and analysis

4.5.1 General feature performance

We summarized all results from the four datasets into four figures to rank the descriptor-affinity pairs according to 24 σ-step configurations (σ and sampling step). For each case and for each dataset, we only reported the best five descriptor-affinity pairs, and for categories of features which were not in the top-5, the best model among them.

Tables 4.2 and 4.3 explain the color and hatching codes used in figures 4.3, 4.4, 4.5 and 4.6.

TABLEAU 4.2 Color of the descriptor in figures 4.3, 4.4, 4.5 and 4.6

Descriptor	Color
Color histogram (RGB)	black
Grayscale histogram (GR)	gray
HOG (HOG)	purple
VGG-19 (VGG)	orange
ResNet-18 (RSN)	red
DenseNet-121 (DNS)	green
EfficientNet-B0 (EFF)	blue
OSNet-AIN (OSN)	pink
Vehicle ReID (VID)	pink

TABLEAU 4.3 Hatching of the affinity measure in figures 4.3, 4.4, 4.5 and 4.6

Affinity	Hatching
L_1 (L1)	\ \ \
L_2 (L2)	/ / /
C_{rank1} (R1)	OO
D_B (B)	XXX
S_C (C)	none

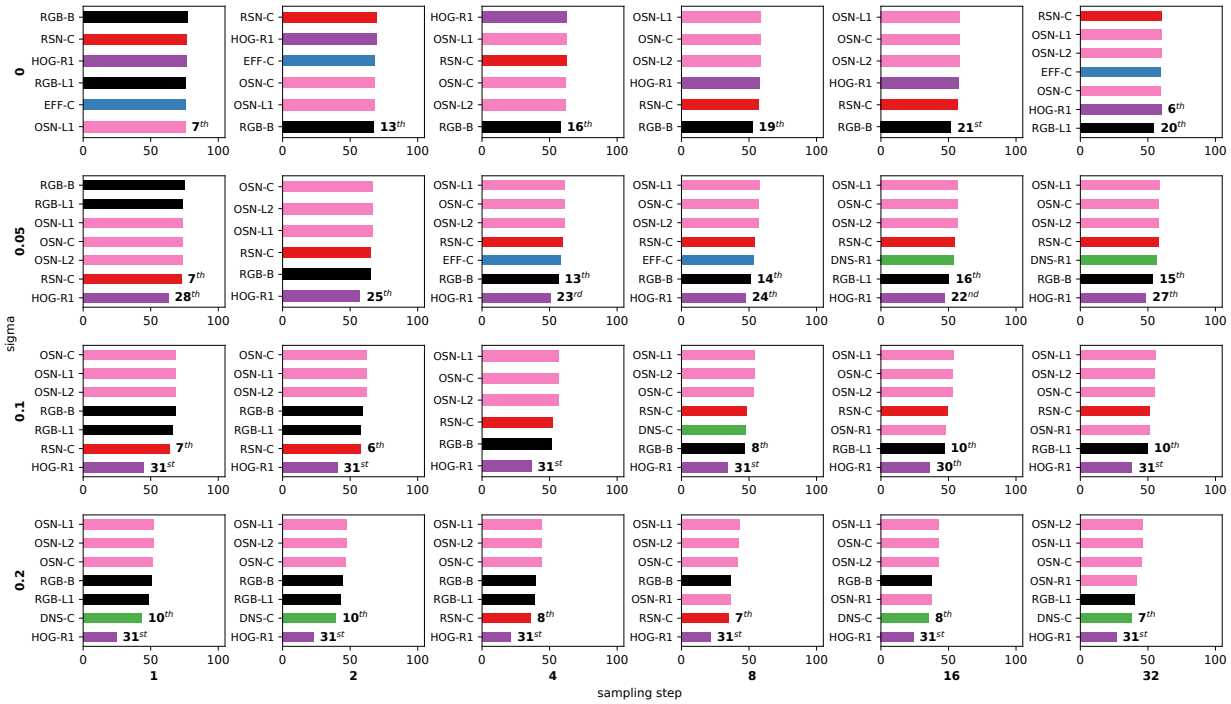


FIGURE 4.3 Mean average precision on WildTrack of the five best descriptor-affinity for each configuration σ-step (when one category of descriptors is not in the top-5, the best result is added). See Tables 4.2 and 4.3 for the color and hatching codes used in the figure. Best viewed in color.

σ-step configuration

unsurprisingly, the four figures show that increasing the parameter σ and/or the sampling step decreases the matching performance of the best descriptor-affinity model.

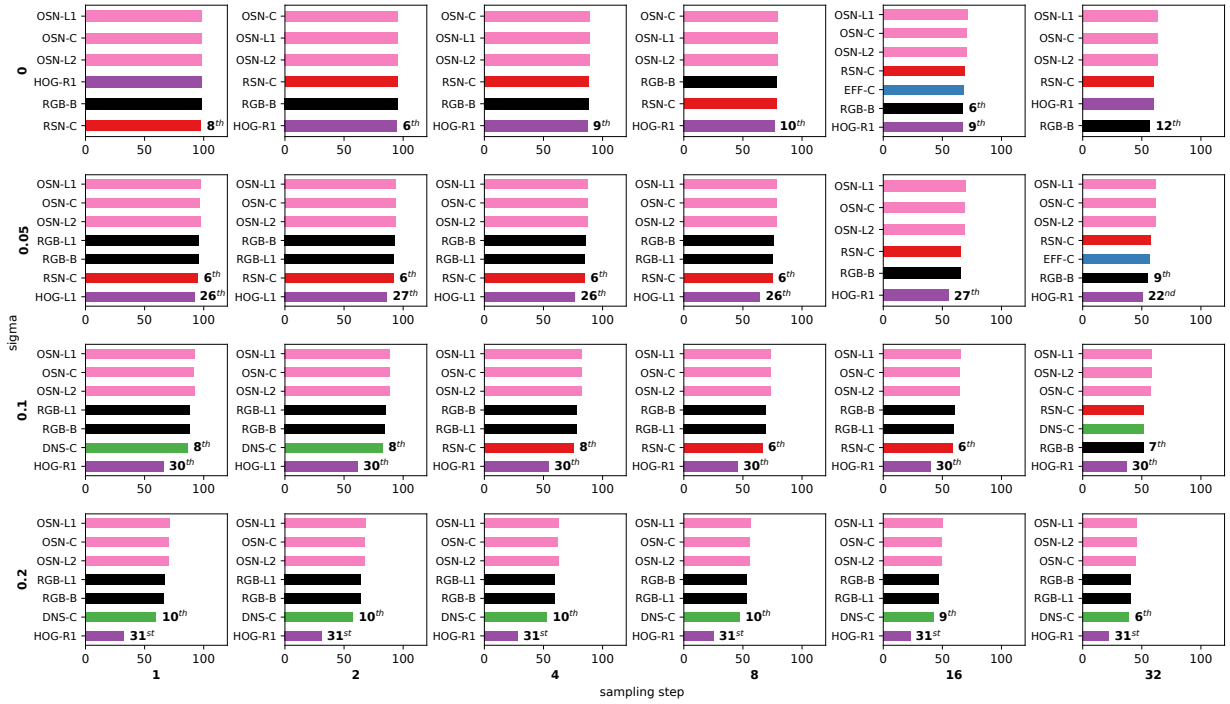


FIGURE 4.4 Mean average precision on MOT17 of the five best descriptor-affinity for each configuration σ-step (when one category of descriptors is not in the top-5, the best result is added). See Tables 4.2 and 4.3 for the color and hatching codes used in the figure. Best viewed in color.

Color and grayscale histograms

color histograms are competitive features, in particular for vehicles-centered datasets, for a low sampling step and a low σ, especially when combined with the Bhattacharyya distance. For these configurations, depending on the datasets, this model is almost always in the top-5. On WildTrack, due to the low framerate, it is not able to discriminate pedestrians when their BBs are separated by more than two seconds, meaning that the objects should not be occluded for too long, or that the detector should have a good recall. We explained this by their color appearance changing too much between these two frames. However, when the BBs do not enclose the object precisely ($\sigma = 0.2$), these models ranked almost always in the top-5. This is due to the excessive loss of semantic information when BBs coordinates are imprecise : low-level characteristics such as colors are in that case relevant.

Histograms of oriented gradients

HOG is a good appearance feature descriptor when the BB coordinates correspond to the ground truth. As soon as they get imprecise, the performance of HOG decreases dramatically.

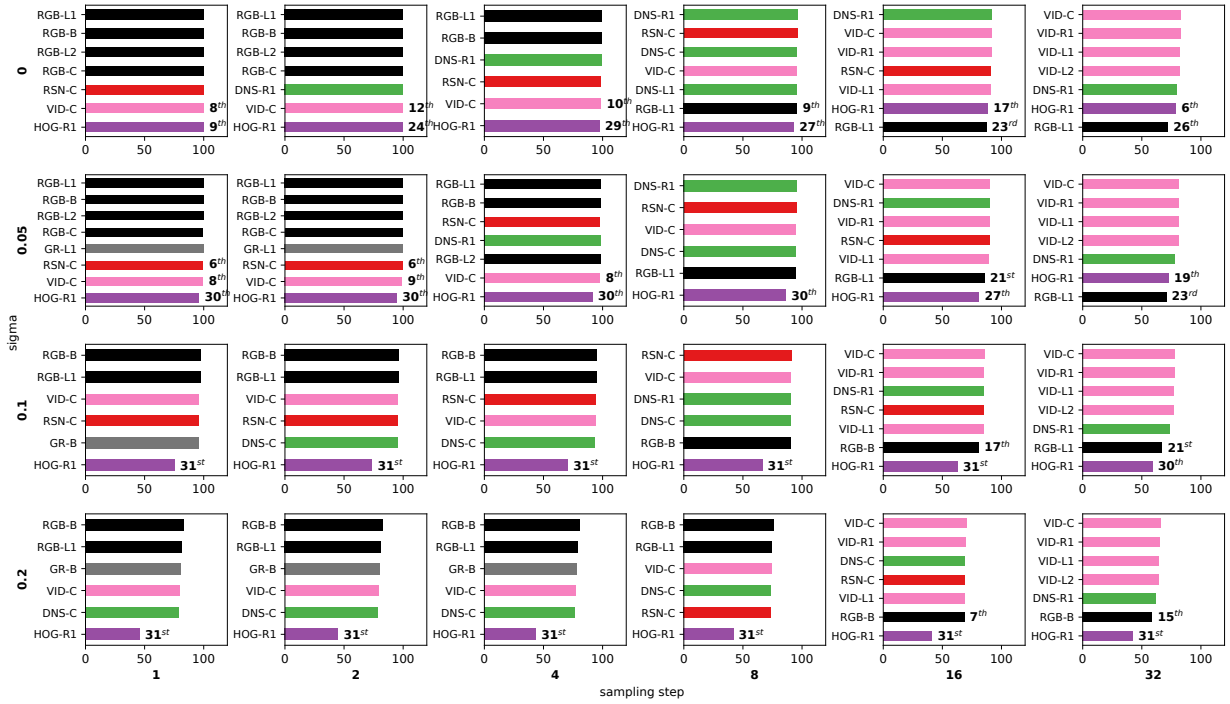


FIGURE 4.5 Mean average precision on DETRAC of the five best descriptor-affinity for each configuration σ-step (when one category of descriptors is not in the top-5, the best result is added). See Tables 4.2 and 4.3 for the color and hatching codes used in the figure. Best viewed in color.

Regardless of the dataset, if the BB correspond to the ground truth, the HOG descriptor is among the best models (when it is not in the top-5, the deviation in absolute value from the best model is small). But when σ increases, its performance falls on average to the 30^{th} position amongst 35 candidates. In the case of very imprecise BBs, the gap between this feature and others is significant. This is due to the construction of HOG : feature vectors are computed over cells of 8×8 pixels. So, a small shift in the BB makes this feature non robust.

CNN-based models

this category of models is competitive when the sampling step is high and σ moderate. When σ is less than 0.1 and the sampling step over 8, a CNN-based model is often in the top-5 ranking. Features computed from a VGG-19 descriptor are not competitive against the three other CNN-based models as these features never rank in the top-5. Moreover, cosine similarity is sometimes a good affinity measure but not consistently.

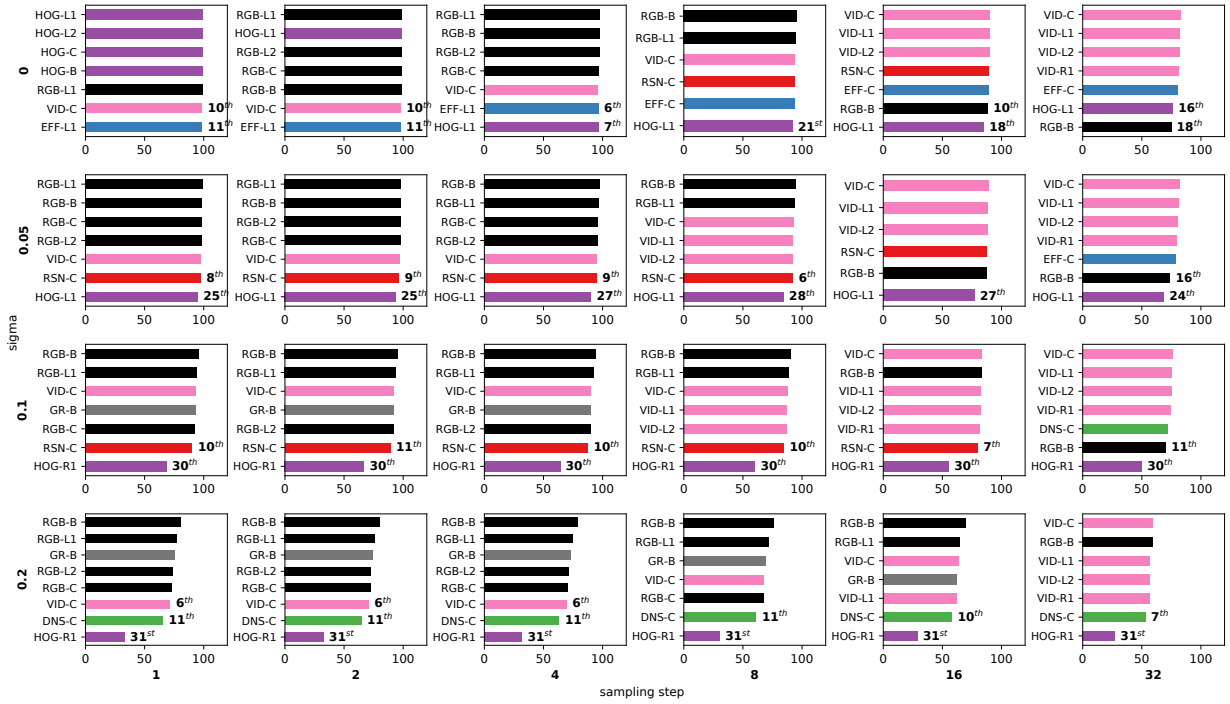


FIGURE 4.6 Mean average precision on UAVDT of the five best descriptor-affinity for each configuration σ-step (when one category of descriptors is not in the top-5, the best result is added). See Tables 4.2 and 4.3 for the color and hatching codes used in the figure. Best viewed in color.

ReID models

in the case of pedestrians tracking, OSNet-AIN is generally the best visual feature regardless of the performance of the detector. In almost all configurations, this model ranks first, with either L_1, L_2 distances or cosine similarity. Since this model is trained to discriminate pedestrians, it is made to extract meaningful instance-specific characteristics from images. So, even if the BB of the image is corrupted, it is able to discriminate persons. As for vehicles, the model from [113] ranks in top-5 when σ is over 0.1 or when the sampling step is over 4. For BBs more similar to the ground truth, the deviation in absolute value from the best model is low. Cosine similarity is systematically the best affinity measure for this descriptor.

4.5.2 Feature performance according to size of objects

In addition to two characteristics of the detector (its ability to predict correctly the coordinates of the BBs and to avoid missed detections), the size of objects might influence the choice of the visual feature descriptor. Smaller objects are commonly the hardest targets to track in MOT. But it is unclear how visual features are affected by the size of BBs.

Figure 4.7 gives the average precision with regard to the query object size, on the UAVDT dataset where there are few occlusions. The configuration σ-step selected correspond to the hardest one (0.2-32) where differences are more meaningful. For a fair comparison, only the L_2 distance is used.

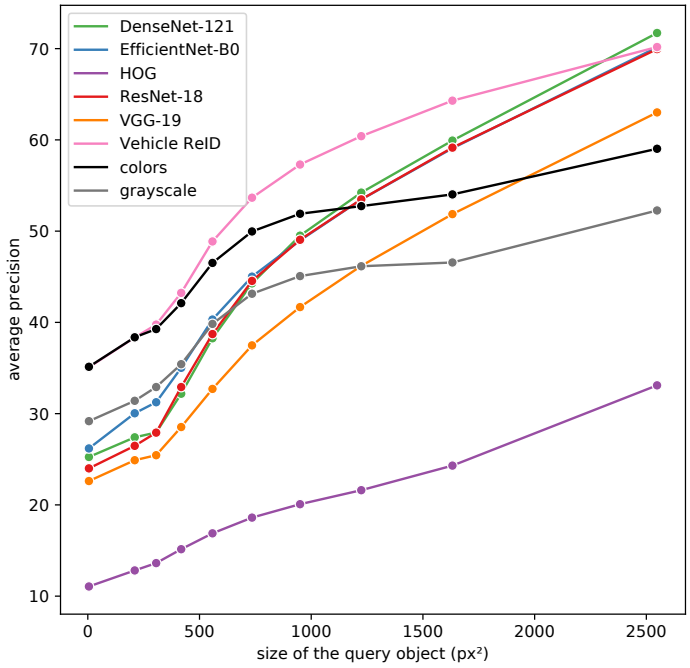


FIGURE 4.7 Average precision according to the query object size, with $\sigma = 0.2$, sampling step at 32 and the L_2 distance on UAVDT, computed at each decile. Best viewed in color.

Firstly, for any feature, the larger the query object is, the easier it is to get the correct match. RGB-histograms are among the best visual features for the smallest objects (approximately smaller than 250 pixels^2 of area), where it is difficult to extract semantics. But for larger objects, ReID features give the best performance. Then, the tested CNN-based models, except VGG-19 which performs more poorly, yield similar results, but lower than ReID which indicates that performing well on ImageNet does not necessarily produce better features for MOT. Similar conclusions can be drawn from other datasets, with the exception of Wildtrack because of its small scale in terms of available data.

4.6 Conclusion

In this paper, we compared several feature descriptors in the context of MOT in urban scenes. Our experiments show that features perform differently given the quality of bounding boxes. ReID features, combined with cosine similarity, are one of the best descriptors for pedestrians

and vehicles, regardless of the performance of the detector. If these models are not available, color histograms with the Bhattacharyya distance is competitive when the boxes are not too noisy. But, as soon as the bounding boxes get noisier, these methods are not able to compete against deep features. Moreover, the size of objects matter on the choice of visual features : in difficult cases, compared to RGB-histograms and modern deep features, ReID features particularly stand out on medium-sized objects.

Acknowledgment

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [CRDPJ 528786 - 18], [DG 2017-06115], and the support of Arcturus Networks.

CHAPITRE 5 ARTICLE 2 : MULTI-OBJECT TRACKING AND SEGMENTATION WITH A SPACE-TIME MEMORY NETWORK

Miah M., Bilodeau G.-A., Saunier N.
20th Conference on Robots and Vision (CRV), 2023, p. 184-193
Published on April 17, 2023
DOI : https://doi.org/10.1109/CRV60082.2023.00031

Nature de la contribution de Mehdi Naim Miah à l'article, telle que soumise à la conférence : conception, méthodologie, recherche bibliographique, codage informatique, analyse des résultats, rédaction initiale du manuscript, visualisation des données

Abstract

We propose a method for multi-object tracking and segmentation based on a novel memory-based mechanism to associate tracklets. The proposed tracker, MeNToS, addresses particularly the long-term data association problem, when objects are not observable for long time intervals. Indeed, the recently introduced HOTA metric (High Order Tracking Accuracy), which has a better alignment than the formerly established MOTA (Multiple Object Tracking Accuracy) with the human visual assessment of tracking, has shown that improvements are still needed for data association, despite the recent improvement in object detection. In MeNToS, after creating tracklets using instance segmentation and optical flow, the proposed method relies on a space-time memory network originally developed for one-shot video object segmentation to improve the association of sequence of detections (tracklets) with temporal gaps. We evaluate our tracker on KITTIMOTS and MOTSChallenge and we show the benefit of our data association strategy with the HOTA metric. Additional ablation studies demonstrate that our approach using a space-time memory network gives better and more robust long-term association than those based on a re-identification network. Our project page is at www.mehdimiah.com/mentos+.

5.1 Introduction

Object tracking is a common task in computer vision : given a video, the objective is to detect objects (for example, all road users such as vehicles, pedestrians and cyclists) and to attribute a unique identifier to each object. This task is fundamental for several applications such as for autonomous vehicles, city traffic management and road safety analysis [9, 127]. In

the last application, since crashes between road users are not always easily observable, the use of computer vision and surrogate risk estimation metrics let researchers collect more data about road safety without relying on the actual observation of incidents. From extracted user trajectories, several safety indicators are computed such as speed, acceleration, post-encroachment time or time-to-collision [128]. Besides, the widespread presence of cameras in cities enables large-scale studies.

However, collecting trajectories from raw video is particularly hard since the trajectories of interest are those involving at least two objects at a close distance. In such situations, it is common to observe occlusions, partial or complete, provoking missed detections and also identity switches (the identifiers of two objects are inverted). Objects trajectories are then incorrect, altering or mis-attributing the computed road safety indicators. That is why developing a tracking method robust to occlusions is particularly important for road safety indicators.

The most popular paradigm for tracking multiple object is "tracking-by-detection" : the first step, named the detection step, detects all objects of interest in all frames of the video and the second step, named the association step, aims to assign unique consistent identities to all the objects in a video. This paper mainly focuses on the second aspect. In particular, multi-object tracking and segmentation (MOTS) [76] consists in tracking several objects *at the pixel level* where the objects of interest are those belonging to a *category class* such as "cars". Until recently, MOTS was generally evaluated with measures heavily biased against the association step [11,129]. Luiten et al. [11] proved that some metrics such as the Multiple Object Tracking Accuracy (MOTA) [10] and its variants ignore largely the association quality. Hence, they introduced the High Order Tracking Accuracy (HOTA) metric that not only balances the detection and the association step but also has a better alignment with the human visual assessment. Therefore, evaluating MOTS with the HOTA metrics (HOTA, DetA and AssA) should lead to methods that are more robust and in turn that could improve the quality of the analysis of traffic videos.

Our proposed tracker, named MeNToS (**Me**mory **N**etwork-based **T**racker **o**f **S**egments), consists in using a propagation method originally developed for one-shot video object segmentation (OSVOS) to solve an association problem for MOTS. OSVOS [130–132] is a task where the segmentation masks of all objects of interest are *provided at the first frame*. It cannot rely on detections since the classes of objects of interest are unknown. Yet, we believe that its principle can be helpful for data association.

We solve the association problem hierarchically. First, given some instance segmentation masks, we associated masks between adjacent frames. As such, masks are spatially close and

visually similar and therefore a method based on a low-level information (meaning colors) may be sufficient. That is why our short-term association is based on the optical flow. After this first association, we obtain continuous tracklets of various lengths. Second, we use a space-time memory (STM) network [1], originally developed to solve OSVOS, to associate the tracklets. A STM network can be viewed as a spatio-temporal attention mechanism [50, 133] able to find a correspondence between a target and a set of queries. As opposed to OSVOS where memory networks [134, 135] are used to *propagate* masks to the next frames, here we use them to *associate* tracklets that are temporally apart.

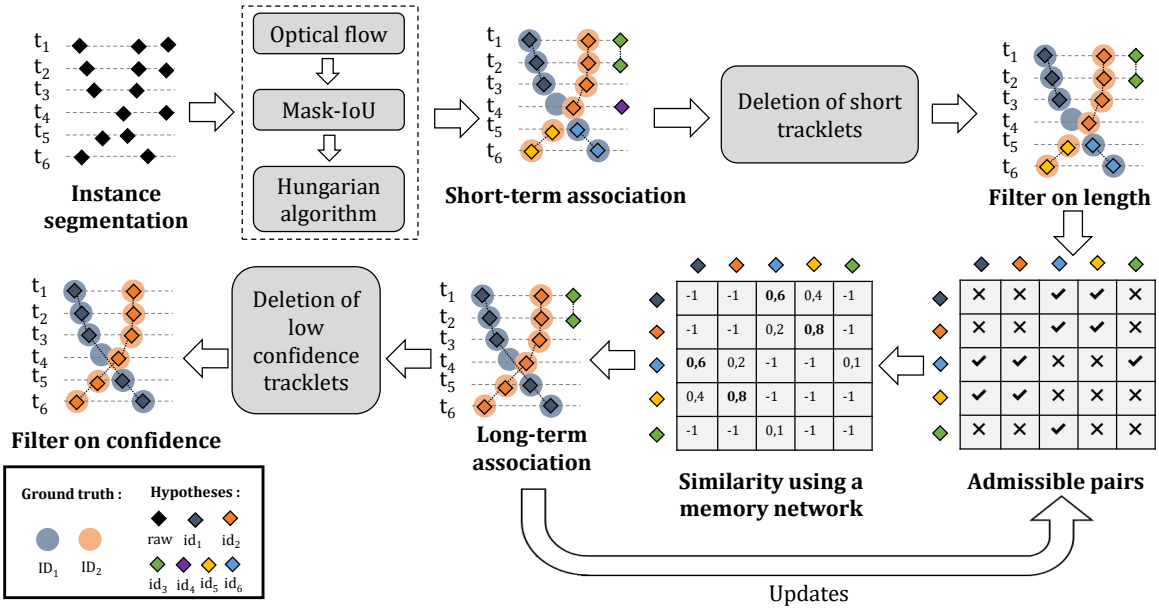


FIGURE 5.1 Illustration of our MeNToS method. Given an instance segmentation, binary masks are matched in adjacent frames to create tracklets. Very short tracklets are deleted. An appearance similarity, based on a memory network, is computed between two admissible tracklets. Then, tracklets are gradually merged starting with the pair having the highest similarity while respecting the updated constraints. Finally, low confidence tracks are deleted.

The main contributions of our work are as follows :

- We propose MeNToS, a tracker to solve MOTS based on a space-time memory network with a new similarity measure between tracklets ;
- We evaluate our tracker on KITTIMOTS and MOTSChallenge to prove that it is competitive, especially on the association part. We demonstrate that our approach for long-term association using a STM network is better than recent approaches based on re-identification networks ;
- We show the usefulness of the HOTA metric for MOTS to capture the improvement resulting from improved data association.

5.2 Related works

5.2.1 MOTS

Similarly to multi-object tracking where the "tracking-by-detection" paradigm is popular [136], MOTS is mainly solved by creating tracklets from segmentation masks and then building long-term tracks by merging the tracklets [12, 137, 138]. Usually, methods use an instance segmentation method to generate binary masks ; ReMOTS [12] used two advanced instance segmentation methods and self-supervision to refine masks. As for the association step, many methods require a re-identification (ReID) step. For example, Voigtlaender et al. [76] extended Mask R-CNN by an association head to return an embedding for each detection. Yang et al. [12] associated two tracklets if they were temporally close, without any temporal overlap with similar appearance features based on all their observations and a hierarchical clustering. Zhang et al. [137] used temporal attention to lower the weight of frames with occluded objects. More recently, Wei et al. [139] proposed to solve MOTS by improving the quality of detected masks with massive instance augmentation during training [140] and refining masks during inference [35], then by associating detections with an ensemble method.

5.2.2 OSVOS

Closely related to MOTS, OSVOS requires tracking objects whose segmentation masks are only provided at the first frame. OSVOS is mainly solved by propagation-based methods : a model learns the representation of the initial mask and tries to make some correspondences in the next frames. MAST [141] used a memory component to predict the future location of objects. STM [1] was proposed to solve OSVOS by storing some previous frames and masks in a memory that is later read by an attention mechanism to predict the new mask in a target image. Such a network was recently used [142] to solve video instance segmentation (VIS), a problem in which no prior knowledge is given about the objects to track. However, it is unclear how the STM network behaves when multiple instances from the same class appear in a video. We show in this work that space-time memory network performs well and can help to solve an association problem by taking advantage of the information at the pixel level and the presence of other objects.

5.2.3 Bridging the gap between MOTS and OSVOS

Despite some clear similarities between the two tasks (prediction of the future position of an object at the pixel level), there are some differences such as the evaluation (sMOTSA, HOTA measures which evaluate the quality of detection and association versus $\mathcal{J}\&\mathcal{F}$ which evaluates

the masks quality with a region similarity and a contour accuracy) and the specification at the first frame (predefined classes without any initial mask versus class-agnostic with initial mask). Recently, Wang et al. [143] addressed this issue by considering a common shared appearance model to solve several tracking problems including MOTS and OSVOS. They compared the performance of several pre-trained vision models such as an ImageNet [5] pretrained architecture, MoCo [86] or CRW [144] architectures to get visual features. Then the authors used these representations to either propagate instances like in OSVOS or associate them like in MOTS. Moreover, for the association step, they computed a similarity score between tracklets with an attention perspective. Their method consists in the reconstruction at the patch level of all features of existing tracklets and current detections. The similarity is then the average cosine similarity between the forward and backward reconstruction. Yan et al. [145] proposed UniTrack to solve both OSVOS and MOTS with a single framework consisting in a unified backbone for representations extraction, a unified embedding module and a unified head. In their network, the distinction between MOTS and OSVOS tasks is made with some "target priors" maps which provide a prior information about the objects to track.

5.2.4 Memory mechanism

The use of a memory mechanism enables to use information from multiple frames to associate detections, specially in case of long periods of occlusion. In this case, it is highly likely that the appearance of objects has changed. Instead on relying on only the last or first frame in which an object was detected, keeping multiple appearance information coming from several frames is a logical way of improving the performance of a tracker. But it is currently unknown which frames to keep in memory : the lastest, the whole appearance information, an average of appearance or a few of them. The tracker MeMOT [146] preserves a large spatio-temporal memory to keep the embeddings of the tracked objects over 24 frames. Conversely, Korbar and Zisserman [147] kept the appearance information of the first and the last five frames of each tracklet to overcome the occlusions, reducing the limitation due to the GPU memory. We show in this work that keeping in memory the appearance information from only two frames leads to competitive results.

5.3 Proposed method

As illustrated in Figure 5.1, our pipeline for tracking multiple objects is based on three main steps : detections via instance segmentation, a short-term association of segmentation masks in adjacent frames and a greedy long-term association of tracklets using a memory network.

5.3.1 Detections

Our method follows the "tracking-by-detection" paradigm. First, we use the non-overlapping instance segmentations from a pre-trained detector. Objects with a detection score higher than θ_d and a size (as the number of pixels of its binary mask) higher than θ_a are extracted. Detections with a low confidence score or with a small size are usually false positives.

5.3.2 Short-term association (STA)

During the short-term association, we associate temporally close segmentation masks between adjacent frames by computing the optical flow. Masks from the previous frames are warped and a mask IoU (mIoU) is computed between these warped masks and the masks from the next frame. As some object classes are visually similar (for example car and truck), only associating objects of the same class may lead to missing some objects due to classification errors. That is why this step is a class-agnostic association, letting the model match a car with a truck if the optical flow corresponds.

The Hungarian algorithm [66] is used to associate masks where the cost matrix is computed based on the negative mIoU. Matched detections with a mIoU above a threshold θ_s are connected to form a tracklet and the remaining detections form new tracklets.

Finally, the class of a tracklet is the one that accumulates the highest sum of confidence score of its detections, and tracklets with only one detection are deleted since they often correspond to false positives.

5.3.3 Long-term association (LTA)

Greedy long-term association and the use of a STM network for re-identification of tracklets are the novel contributions of our approach. Once tracklets have been created, it is necessary to link them in case of fragmentation caused, for example, by occlusion. In this long-term association, we use a space-time memory network as a similarity measure between tracklets by propagating some masks of a tracklet in the past and the future. Given a binary segmentation mask in a frame of reference and a query frame, the STM network outputs a heatmap indicating the probability of the location of the object of reference in the query frame. If a predicted heatmap of a tracklet sufficiently overlaps a mask of another tracklet, these two tracklets are linked together. Given the fact that this procedure is applied at the pixel-level on the whole image, this similarity is only computed on a selection of admissible tracklet pairs to reduce the computational cost. At this step, we point out that all tracklets have a length longer than or equal to two.

Measure of similarity between tracklets

Our similarity measure is based on the ability to match some parts of two different tracklets (say T^A and T^B) and can be interpreted as a pixel-level visual-spatial alignment rather than a patch-level visual alignment [12, 137]. For that, we propagate some masks of tracklet T^A to other frames where the tracklet T^B is present and then compare the masks of T^B and the propagated version of the mask heatmaps, computed before the binarization, for T^A. The more they are spatially aligned, the higher the similarity is. In details, let us consider two tracklets $T^A = (M_1^A, M_2^A, \cdots, M_N^A)$ and $T^B = (M_1^B, M_2^B, \cdots, M_P^B)$ of length N and P respectively, such that T^A appears first and where M_1^A denotes the first segmentation mask of the tracklet T^A. We use a pre-trained STM network [1] to store two binary masks as references (and their corresponding frames) : the closest ones (M_N^A for T^A and M_1^B for T^B) and a second mask a little farther (M_{N-n-1}^A for T^A and M_n^B for T^B). The farther masks are used because the first and last object masks of a tracklet are often incomplete due, for example, to occlusions. Then, the reference frames are used as queries to produce heatmaps with continuous values between 0 and 1 (H_N^A, H_{N-n-1}^A, H_1^B, H_n^B). Finally, the average cosine similarity between these four heatmaps and the four masks (M_N^A, M_{N-n-1}^A, M_1^B, M_n^B) is the final similarity between the two tracklets, denoted as $sim(T^A, T^B)$. Figure 5.2 illustrates a one-frame version of this similarity measure between tracklets.

Selection of pairs of tracklets

Instead of estimating a similarity measure between all pairs of tracklets, a selection is made to reduce the computational cost [12]. The selection is based on the following heuristic : two tracklets may belong to the same objects if they belong to the same object class, are temporally close, spatially close and with a small temporal overlap.

In details, let us denote $f(M)$ the frame where the mask M is present, $\bar{M}$ its center and fps, H and W respectively the number of frames per second, height and width of the video. The temporal ($C_t(T^A, T^B)$), spatial ($C_s(T^A, T^B)$) and temporal overlap ($C_o(T^A, T^B)$) costs between T^A and T^B are defined respectively as :

$$C_t(T^A, T^B) = \frac{|f(M_N^A) - f(M_1^B)|}{fps}, \tag{5.1}$$

$$C_s(T^A, T^B) = \frac{2}{H + W} \times \|\bar{M_N^A} - \bar{M_1^B}\|_1, \tag{5.2}$$

$$C_o(T^A, T^B) = |\bigcap_{T \in T^A, T^B} \{f(M), \forall M \in T\}| \tag{5.3}$$

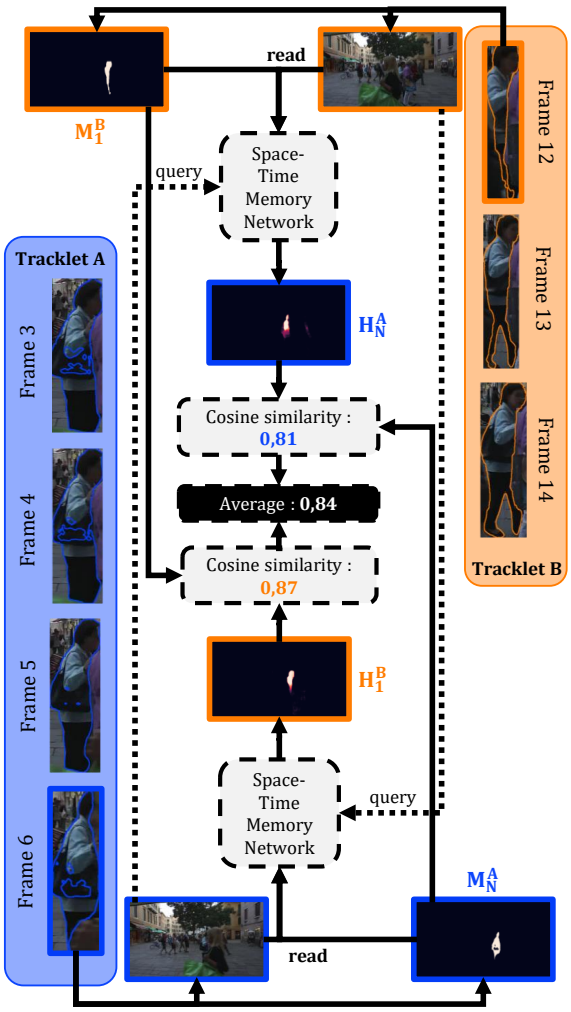


FIGURE 5.2 Similarity used at the long-term association step. For simplicity, only one mask and frame are used as reference and as target in the space-time memory network.

A pair (T^A, T^B) is admissible if the tracklets belong to the same object class, $C_t(T^A, T^B) \leq \tau_t$, $C_s(T^A, T^B) \leq \tau_s$ and $C_o(T^A, T^B) \leq \tau_o$.

Greedy association

We gradually merge the admissible pairs with the highest cosine similarity, $sim(T^A, T^B)$, if it is above a threshold θ_l, while continuously updating the admissible pairs using equation 5.3. A tracklet can therefore be repeatedly merged with other tracklets. Finally, tracks having their highest detection score lower than θ_f are deleted.

5.4 Experiments

5.4.1 Implementation details

At the detection step, the detection are provided by the RobMOTS challenge [148] obtained from a Mask R-CNN X-152 [31] and Box2Seg Network [149] for all 80 categories of COCO. The threshold of the detection confidence score θ_d is set to 50 % and small masks whose area is less than $\theta_a = 128$ pixels are removed.

For the short-term data association, the optical flow is computed with RAFT [58]. We select $\theta_s = 0.15$ for the threshold in the Hungarian algorithm.

For the long-term data association, the pre-selection is done with $(\tau_t, \tau_s, \tau_o) = (1.5, 0.2, 1)$. We took a space-time memory network pretrained on DAVIS [150]. To have a more generic tracker, STM was not fine-tuned on MOTSChallenge or KITTIMOTS. To measure similarity, the second frame is picked using $n = 5$. If that frame is not available, $n = 2$, is used instead. We select $\theta_l = 0.30$ in the greedy association. Then, the tracklets having at least one observation with a confidence score higher than $\theta_f = 90\%$ are kept. All these hyper-parameters were selected using the HOTA score on the validation sets and *remain fixed regardless of the dataset and object classes.*

5.4.2 Datasets and performance evaluations

We evaluated our method on KITTIMOTS and MOTSChallenge [76] two common datasets about MOTS. KITTIMOTS contains 21 training videos and 29 test videos on cars and pedestrians obtained from a camera mounted on the roof of a car. The scenes are captured at 10 Hz displaying real-world traffic situations. MOTSChallenge contains 4 training videos and 4 test videos only with pedestrians in cities. Tracking performance is measured with metrics including MOTA [10] (and its variant sMOTSA, MOTSA and sMOTSP), Identity F1 score (IDF1) which measures the quality of trajectory identity accuracy, Identity switches (IDSw) which counts the number of inversion of identity. We also use, when possible, the recently introduced HOTA metric [11] that fairly balances the quality of detections and associations. It can be decomposed into the DetA and AssA metrics to measure the quality of these two components. The higher the HOTA is, the more the tracker is aligned with the human visual assessment.

5.4.3 Results

Results [1] in tables 5.1 and 5.2 indicate that our method is competitive on the association performance. We ranked first and second respectively on pedestrians and cars on this criteria on KITTIMOTS and our identity switches are the lowest on MOTSChallenge with the second highest IDF1. *Contrarily to other methods, we point out that ours does not require any additional fine-tuning on the benchmarks or per benchmark hyper-parameters selection.* For example, we did not fine-tune STM on the benchmarks. Note also that the methods in the tables do not use all the same detection inputs. The ablation studies in the next section gives a better understanding of our contribution by fixing the detection inputs to assess only the data association component.

TABLEAU 5.1 Results on the test set of KITTIMOTS. **bold red** and *italic blue* indicate respectively the first and second best methods.

Method	Cars			Pedestrians		
	HOTA↑	DetA↑	AssA↑	HOTA↑	DetA↑	AssA↑
ViP-DeepLab [151]	**76.4**	**82.7**	70.9	*64.3*	**70.7**	*59.5*
EagerMOT [152]	74.7	76.1	*73.8*	57.7	60.3	56.2
MOTSFusion [138]	73.6	75.4	72.4	54.0	60.8	49.5
ReMOTS [12]	71.6	78.3	66.0	58.8	68.0	52.4
PointTrack [153]	62.0	*79.4*	48.8	54.4	62.3	48.1
MeNToS (ours)	*75.8*	77.1	**74.9**	**65.4**	*68.7*	**63.5**

TABLEAU 5.2 Results on the test set of MOTSChallenge. **bold red** and *italic blue* indicate respectively the first and second best methods.

Method	sMOTSA↑	IDF1↑	MOTSA↑	FP↓	FN↓	IDSw↓
ReMOTS [12]	**70.4**	**75.0**	**84.4**	*819*	*3999*	231
GMPHD [154]	*69.4*	66.4	*83.3*	935	**3985**	484
MPNTrack [155]	58.6	68.8	73.7	1059	7233	*202*
SORTS [156]	55.0	57.3	68.3	1076	8598	552
TrackRCNN [76]	40.6	42.4	55.2	1261	12641	567
MeNToS (ours)	64.8	*73.2*	76.9	**654**	6704	**110**

1. Full results at `https://www.cvlibs.net/datasets/kitti/eval_mots.php` for KITTIMOTS and `https://motchallenge.net/results/MOTS/` for MOTSChallenge

Some qualitative results are provided in Figure 5.3. We notice that our method can successfully retrieve objects even after occlusions, as happens to the green, yellow and purple cars in the first sequence and the blue pedestrian in the second sequence. The main limitations are contaminated segmentation masks and short tracklets since they provide little information and their appearances are unstable. For example, pedestrians usually walk in group provoking more occlusions and eventually lowering the quality of the masks, as in the third and fourth sequences. A large spatial displacement of the center of the masks may happen in the case of an occlusion, like in the fifth sequence. In that particular case, the spatial distance between the center of the two masks (the yellow one in the first frame and the green one in the second one) is higher than the threshold used for the constraint in equation 5.2.

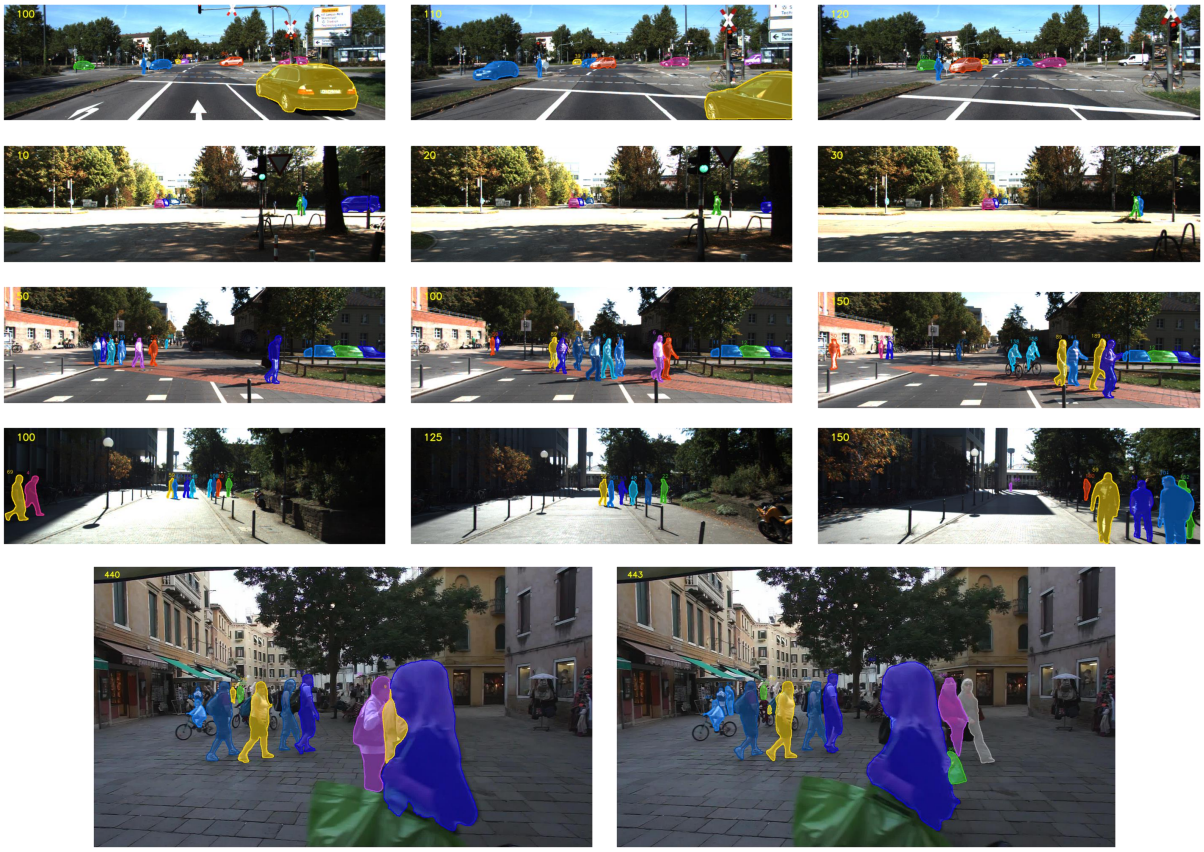


FIGURE 5.3 Qualitative results on KITTIMOTS and MOTSChallenge. Each row corresponds to a subsequence of a video clip.

5.5 Ablation studies

5.5.1 Contribution of each step

To examine the effects of each step, we evaluate the performance by successively adding a component at a time.

TABLEAU 5.3 ablation studies on the validation set of KITTIMOTS (KT) and the train set of MOTSChallenge. each step of our approach leads to an improvement in terms of HOTA and sMOTSA.

Step	KT-car	KT-ped	MOTSChallenge	
	HOTA	HOTA	HOTA	sMOTSA
STA	79.4	62.0	57.9	62.8
+ filter	79.8 (+0.4)	63.3 (+1.3)	58.2 (+0.3)	64.0 (+1.2)
+ LTA	84.1 (+4.3)	66.2 (+2.9)	64.5 (+6.3)	64.9 (+0.9)
+ filter	84.1 (+0.0)	67.6 (+1.4)	65.4 (+0.9)	67.7 (+2.8)

Results of Table 5.3 indicate that all steps improve the tracking performance (either in terms of HOTA or sMOTSA). More precisely, on KITTIMOTS and MOTSChallenge, the HOTA metric is highly sensitive to the quality of the association. From the tracking results obtained at the STA step, on average, two thirds of the total improvement in HOTA is made at the LTA step. As for sMOTSA, unsurprisingly, it shows a smaller improvement brought by the long-term association. Consequently, improving the LTA step leads to a boost in terms of HOTA : this improvement in the data association is not fully captured by the previous sMOTSA measure.

5.5.2 Upper bounds with oracles

Since the HOTA gives more importance to the data association part of MOTS compared to the sMOTSA, we also conducted some experiments to measure the current limitations of our method and estimate the steps which can lead to the biggest gain in HOTA. Using the ground truth annotations on KITTIMOTS, we build two methods that incorporate some ground truth knowledge :

1. **OracleLTA** that perfectly associates tracklets (perfect long-term association). Precisely, after associating masks using the optical flow with RAFT and filtering short tracklets, we compute the ground truth identity for each tracklet and associate them based on the same ground truth identity. This oracle ignores the step of the selection

of candidate pairs. Then tracklets that do not correspond to any ground truth one are deleted.

2. **OracleSLTA** that perfectly associates detections from all frames (perfect short-term and long-term association). The oracleSLTA is obtained with perfect short and long-term assignments of detections that match the ground truth and a perfect deletion of false positives. Precisely, this is the highest upper bound using the ground truth knowledge after the detection step. For all segmentation masks, if they correspond to a ground truth mask with a mIoU higher than 50%, they are kept and the ground truth identity is attributed, otherwise they are removed.

We evaluated the HOTA of these two oracle upper bounds on the validation set of KITTI-MOTS and compare them with our method. Results from Table 5.4 indicate that MeNToS is able to reach performance close to the oracles, although improvements are still possible. Potential gains of 3.3 and 5.1 points are still possible on the car and pedestrian classes of KITTIMOTS. Moreover, the short-term association step with RAFT is nearly perfect for cars : compared to the tracker with perfect association at both short and long-term association, 70% of the gap comes from the long-term association step. On the contrary, the short-term association is the limiting step for pedestrian : MeNToS is close to the performance of the tracker using oracle information at the long-term association step. We hypothesize that the difference between the car and pedestrian classes comes from the fact that there are more distractors (detected objects wrongly classified, such as mannequins recognized as persons), for pedestrians than for cars. Indeed, the model fully using the ground truth annotations is able to eliminate all distractors. Moreover, pedestrians are harder to track in KITTIMOTS because of their deformable nature and of their vertical shape which moves more on the horizontal axis between frames when the car capturing the videos is moving.

Tableau 5.4 HOTA for the oracle methods on the validation set of KITTIMOTS (KT).

Method	KT-car	KT-ped	Use of oracle at STA	Use of oracle at LTA
MeNToS	84.1	67.6	-	-
OracleLTA	86.4 (+2.3)	69.4 (+1.8)	-	✓
OracleSLTA	87.4 (+3.3)	72.7 (+5.1)	✓	✓

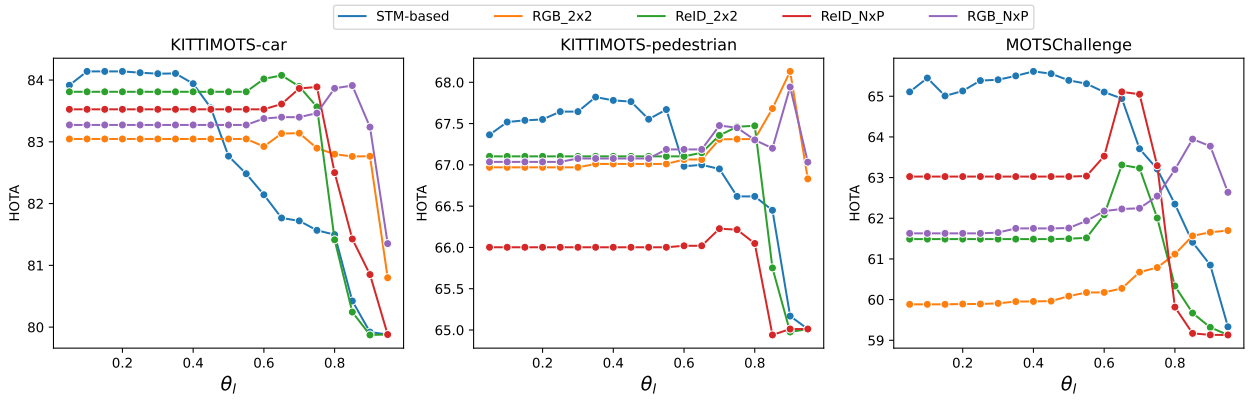


FIGURE 5.4 Comparison of some strategies of the long-term association for KITTIMOTS-car, KITTIMOTS-pedestrian and MOTSChallenge.

5.5.3 Comparison with other strategies of long-term association

In the proposed method, we use a STM network to compute a similarity measure between tracklets. Other methods traditionally used re-identification features [12] or color histograms [157]. Moreover, they also tend to use more reference frames whereas ours only takes into account two frames. In order to measure the contribution of the STM network combined with a cosine similarity, we replace them by one the following method :

1. `RGB_2x2` : for two tracklets, we extracted the histogram of color of the same two frames of references computed at the pixel level. The similarity measure is then the average of four Bhattacharyya coefficients ;
2. `RGB_NxP` : for two tracklets, we extracted the histogram of color of all frames of reference computed on the mask. The similarity measure is then the average Bhattacharyya coefficient over all possible combinations of histograms between the two tracklets ;
3. `ReID_2x2` : for two tracklets, we computed the ReID features of the same two frames of reference. Their similarity is the average of four cosine similarities ;
4. `ReID_NxP` : for two tracklets, we computed the ReID features of all frames of reference. Their similarity is the average cosine similarity over all possible combinations of ReID features between the two tracklets.

The ReID features are computed with the OsNet-AIN [104] network. This is motivated by some works [13, 37, 158] which indicate that ReID features are suitable for associating far apart detections and that color histograms are efficient specially for small objects.

Figure 5.4 illustrates the behavior of these four methods for LTA alongside the one based on

the STM network on HOTA with regard to the similarity threshold θ_l. First, our STM-based approach is the one with the highest HOTA and is less sensitive to the parameter θ_l : the HOTA is the highest on a large interval ($[0.1; 0.5]$). It is generally better than other methods requiring less frames in memory. Second, the four additional methods are sensitive to the parameter θ_l : the interval where the HOTA is the highest is narrow and there is no single method that is better on all datasets. Then, for color histogram-based methods, using all frames of a tracklet as reference leads to a higher HOTA than using only two frames of reference.

Generally STM-based association performs better, but it can face some issues when different parts of an object are occluded. For example, if one tracklet displays only the hood of a car and another only the car trunk, even if they belong to the same object, our STM-based approach could not associate such tracklets since no particular parts are in common. As for ReID features, they are trained to overlook such events but in the case of heavy occlusion, the features would also probably be too dissimilar. To reduce this effect, we performed the propagation at the pixel level and on the whole image for better precision to match tracklets. Working on the entire image is advantageous when a detection is missing. Moreover, this is the closest form of use of a STM network within the framework of the OSVOS.

5.5.4 Number of frames of reference

From the results of Figure 5.4, it seems that taking into account more reference frames leads to better results. Is this trend still relevant for our method based on STM ? Since it is not technically possible with our GPU to load all frames in memory and compute the attention, we compared the following approaches :

1. `Frame 1` : only the closest frames ($f(M_N^A)$ and $f(M_1^B)$ with the notation used in section 5.3.3) are used as references, as illustrated in Figure 5.2 ;
2. `Frames 1-2` : the two closest frames ($f(M_N^A)$, $f(M_{N-1}^A)$ and $f(M_1^B)$, $f(M_2^B)$) are used at reference ;
3. `Frames 1-5/2` : this is the method currently used in our approach. The closest frames are used and $f(M_{N-4}^A)$ (respectively $f(M_5^B)$) if it exists for T^A (respectively T^B), otherwise $f(M_{N-1}^A)$ (respectively $f(M_2^B)$) ;
4. `Frames 1-2-5` : the two closest frames are used and $f(M_{N-4}^A)$ (respectively $f(M_5^B)$) if it exists for T^A (respectively T^B).

Figure 5.5 illustrates the performance in terms of HOTA for different selections of reference frames in the STM with regard to the similarity threshold θ_l. On KITTIMOTS, using the

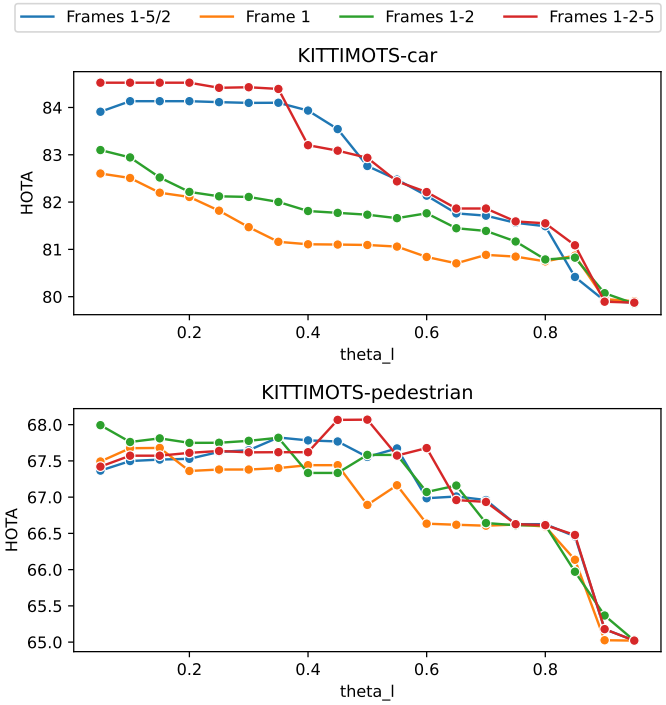


FIGURE 5.5 Ablation studies on the selection of the reference frames in the STM-based method on the validation set of KITTIMOTS.

closest frames as reference leads to the lowest HOTA. For pedestrians, using at least two reference frames provides a little improvement. With more frames, it seems that the performance is saturating. We hypothesize that this is due to some masks contaminated by other pedestrians, as people tend to walk in groups. As for cars, replacing the second closest frames by its fifth equivalent contributes to a boost in terms of HOTA. Considering three frames seems to saturate the HOTA. We hypothesize that using the fifth frame as a reference is a better choice than the second one because it is less similar to the first frame. Hence, more uncorrelated information is available for the STM. These results are similar to the conclusion drawn by Lai et al [141] on OSVOS : their Memory-Augmented Tracker used both short and long term memory to recover objects.

5.6 Conclusion

We propose a method to solve MOTS using a space-time memory network originally developed for solving OSVOS. It is mainly based on a hierarchical association where masks are first associated between adjacent frames to form tracklets. Then, these tracklets are associated using a STM network, leveraging the ability of the network to match similar parts. Experi-

ments on KITTIMOTS and MOTSChallenge show that our approach gives good performance with regard to the association quality. We demonstrate that our approach for long-term association using a STM network is better than the recent approach based on re-identification networks and is less sensitive to hyper-parameters.

Acknowledgment

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [DGDND-2020-04633 and RGPIN-2017-06115].

CHAPITRE 6 ARTICLE 3 : LEARNING DATA ASSOCIATION FOR MULTI-OBJECT TRACKING USING ONLY COORDINATES

Miah M., Bilodeau G.-A., Saunier N.

Article submitted to Pattern Recognition on May 3, 2024

Nature de la contribution de Mehdi Naim Miah à l'article, telle que soumise à la revue : conception, méthodologie, recherche bibliographique, codage informatique, analyse des résultats, rédaction initiale du manuscrit, visualisation des données, vérification de la reproductibilité

Abstract

We propose a novel Transformer-based module to address the data association problem for multi-object tracking. From detections obtained by a pretrained detector, this module uses only coordinates from bounding boxes to estimate an affinity score between pairs of tracks extracted from two distinct temporal windows. This module, named TWiX, is trained on sets of tracks with the objective of discriminating pairs of tracks coming from the same object from those which are not. Our module does not use the intersection over union measure, nor does it requires any motion priors or any camera motion compensation technique. By inserting TWiX within an online cascade matching pipeline, our tracker C-TWiX achieves state-of-the-art performance on the DanceTrack and KITTIMOT datasets, and gets competitive results on the MOT17 dataset. The code will be made available upon publication.

6.1 Introduction

Multi-object tracking (MOT) consists in detecting all objects of interest, such as cars or pedestrians, and assigning them a unique identity throughout a video. Common applications are road safety analysis, video-surveillance and environment awareness in self-driving cars. With the improvement of object detectors, a popular paradigm to solve MOT is tracking-by-detection, which consists of two steps : detecting objects in each frame of the video and associating detections that correspond to the same object. Under this paradigm, MOT is mainly solved as a data association problem : given two sets of detections, the objective is to find those that refer to the same object. This data association can either be done in an online setting or in an offline setting. In the former one, no information coming from the future can be exploited to track objects. This is the case for real-time applications, like self-driving cars. The offline setting is more suitable for applications such as road traffic and safety analysis.

In this setting, it is common to first associate detections between adjacent frames to create tracklets (continuous fragments of trajectories), which are later associated to form complete trajectories of objects.

Usually, given some detections, associations can be made using cues such as appearance (color and texture) and spatio-temporal information (position and motion). Several offline tracking algorithms [15, 37, 88] rely on appearance cues, abandoning the motion information. In contrast, many online tracking algorithms [20, 159] are only based on motion to make them efficient for real-time applications.

In the case of occlusions or missed detections, online trackers *generate* new boxes using probabilistic methods or by learning multi-modal distributions of trajectories [44, 160]. This extrapolation may provoke some drifts due to the autoregressive nature of the prediction, or create some false detections. Conversely, it is possible to associate tracklets after an occlusion by solving a simpler *discriminative* task : given a set of tracks and detections, find the ones that correspond to the same object.

These observations raise the following question : is it possible to learn to associate tracklets, without using any appearance information, any camera motion estimation, or any motion prior ? In this work, we investigate this question and propose a method to do exactly that. We present TWiX, a *Transformer-based neural network that returns an affinity score between all pairs of tracklets*. It is based on a *supervised contrastive learning task*, where the objective is to discriminate *pairs of tracklets* coming from the same objects from other pairs of tracklets. In TWiX, the representation of a pair of tracklets depends not only on its own characteristics but also on the *context*, represented by the characteristics of other pairs of tracklets. To the best of our knowledge, this is the first work to propose a Transformer-based network to associate tracklets in a discriminative fashion without using any appearance information, motion prior or camera motion compensation technique.

In summary, we make the following contributions :

- we propose a new association module, named TWiX, that returns a context-dependent affinity score for each pair of tracklets ;
- we propose a supervised contrastive framework to learn representations for tracklets given their spatio-temporal neighborhood (context) ;
- we obtain state-of-the-art (SOTA) results on DanceTrack and KITTIMOT and competitive results on MOT17, three popular datasets for MOT when using the module TWiX in the cascade matching pipeline ;
- we conducted several ablation studies and visualizations to show the importance of each component of our method.

6.2 Related works

We first introduce some previous tracking-by-detection approaches and their components for data association. Then, we discuss contrastive learning as an extension of re-identification and the use of Transformer networks to solve MOT.

6.2.1 Tracking-by-detection and association

Thanks to improvements in the field of object detection, tracking-by-detection is a very popular paradigm to solve MOT as an association problem, where the objective is to predict whether two detections belong to the same object. Cues such as appearance and motion are commonly used to compute a similarity score and the Hungarian algorithm is often used to associate detections in an online fashion.

First, tracking heavily relies on spatial information to associate objects. SORT [20] and ByteTrack [67] use the Kalman filter to predict the future position of an object, under a linear constant velocity model, and measure an affinity score with the Intersection over Union (IoU). C-BIoU [42] estimates the future position with a simple linear model and uses a modified version of IoU by proportionally increasing the size of the boxes when computing the affinity score. This allows the model to have a non-zero score even in case where an object moves quickly. IoU-Tracker [68] gets rid of any motion model by relying only on IoU to match objects between adjacent frames. However, relying only on spatial positions can provoke some incorrect associations especially in case of occlusions, where objects are naturally close to each other, or when the camera moves.

A strategy to deal with a moving camera is to use a camera motion compensation (CMC) technique [15]. It evaluates the camera motion between two adjacent frames through image registration to adjust the coordinates of bounding boxes. Techniques such as optical flow, enhanced correlation coefficient maximization are required, impacting the speed of the tracking. UCMCTracker proposed a novel camera compensation technique that is applied uniformly throughout each sequence, without computing any frame-by-frame registration [38]. However, this setting either requires to have access to camera parameters or must be done manually for each sequence.

Another solution is to use visual features in the data association task. The most common methods to extract visual features are based on convolutional neural networks (CNNs) pre-trained on a classification task or on a re-identification task [37]. Such methods extract a visual feature for each detection independently of other detections. However, they also struggle in case of crowded scenes due to total or partial occlusions between objects or in

case of different objects with the same appearance [8].

Conversely, a graph neural network (GNN) is a model where the representation of a node depends on those of its adjacent nodes [161]. For MOT, GNNs are used to model complex interactions between objects [70,162]. However, GNNs have a limited temporal view span due to their limited scalability. Besides, the interactions between pedestrians can also be learned using a social force model [163].

6.2.2 Contrastive learning

Contrastive learning is an approach to learn representations by making them agree between "similar instances" and disagree between "dissimilar instances". Re-identification learning with a triplet loss is a special case of contrastive learning where there is one anchor, one positive and one negative instance [61]. By considering multiple negative instances, this loss can be extended to the contrastive loss, as a logistic regression classifier, which learns to discriminate positive pairs from multiple negative pairs [81]. This contrastive loss is widely used in the field of unsupervised representation learning [85, 164] and recently for online MOT to learn the representation of a track [165]. Some representations are also trained with multiple positive pairs [88]. Data augmentation, large batch size and hard negative mining (a technique used to focus the learning on the most difficult examples) are commonly listed as critical components in contrastive learning to extract good representations [166]. In computer vision, such augmentations are for example image cropping, color dropping, color distortion and random Gaussian blur [85]. A large batch size ensures that the contrastive loss encounters enough negative samples, and the hard negative mining helps to learn more discriminative representations [164].

Intuitively, a contrastive framework aims at finding positive pairs among a large collection of negative pairs. In MOT, GNNs offer a natural way to create a few positive pairs (detections from the same object) and many negative pairs (detections from two different objects). GNNs were used to learn representations for all detections of a given object based on all detections from its previous and next frames [70]. Our framework differs from this previous work by learning representations at the level of a pair of tracklets instead of the detection level, and by formulating the problem as a contrastive task instead of a classification task, such as in the Deep Affinity Network [167].

6.2.3 Transformers for tracking

Transformer networks are based on an attention mechanism to extract relevant representations from sequences [50]. Even if this network was originally proposed for natural language processing in which sentences are the input sequences, a Transformer network can be exploited for image classification [168] or object detection if an image is reshaped as a sequence of patches.

The first use of a Transformer for solving MOT is the work of TransTrack [71]. It jointly detects and associates objects in an online fashion by using object features from previous frames as queries of the current frame. Similarly, Trackformer was proposed to solve multi-object tracking and segmentation [16]. This online algorithm introduced the track queries to refer to an object in a video. MOTR used a Transformer decoder to model a full track in an end-to-end manner [72]. It showed good ability to associate without using any re-identification method or track non-maximum suppression. MO3TR [169] proposed an end-to-end trainable Transformer-based tracker by learning the representations implicitly. Their algorithm combined two Transformers, one using temporal attention and another one using spatial attention.

Giuliari et al [51] proposed to forecast trajectories in a auto-regressive manner using a Transformer instead of a traditional recurrent neural network.

Our work distinguishes itself from these previous works by *explicitly* considering the association step as a *discriminative task, without generating any positions (e.g. with a Kalman filter or linear motion model), without using any appearance information and without using the IoU at the inference step.*

6.3 TWiX

We propose TWiX, a Transformer-based module that returns an affinity matrix between two sets of tracklets selected from two temporal windows. It first considers the interaction between observations within two tracklets, then the interaction between all the pairs of tracklets. TWiX is based on contrastive learning to extract representations taking into account the spatio-temporal coordinates of all the tracklets. For that, given some detections, we first create tracklets. Then using two temporal windows, each containing a set of tracklets, we create all pairs of tracklets from the two sets and we use a contrastive loss to learn whether a given pair of tracklets comes from the same object given all other pairs of tracklets to complete the MOT process.

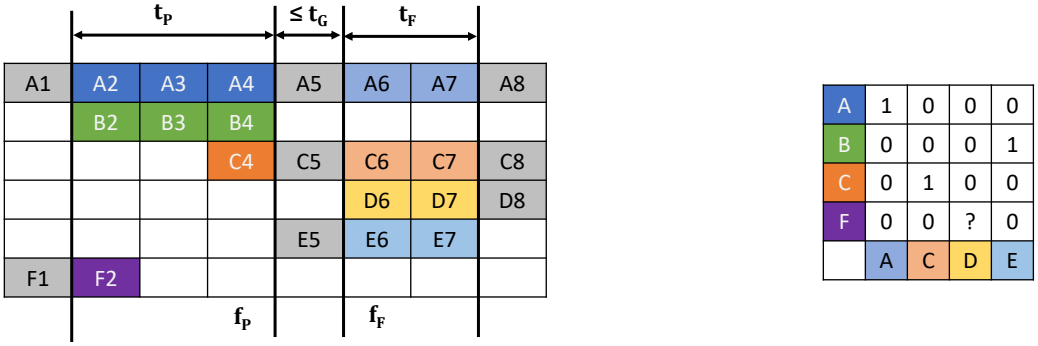
6.3.1 Creation of tracklets

The tracklets are created from detected bounding boxes and *not* from any ground-truth positions. Indeed, heavy data augmentations are a key aspect in the use of a contrastive framework. Detections are a natural way to augment the data : bounding box coordinates are noisier, some boxes are missing (false negative), others are extra (false positive).

Given a video and a set of detections, we create tracklets by associating bounding boxes between adjacent frames. Many approaches are possible like associating detections from frame t and $t+1$ that share the highest IoU or a buffered version of IoU [42]. This IoU can also be computed after warping the detections with the optical flow. The Kalman filter can also be used to leverage object motion. We use the IoU and Kuhn-Munkres algorithm on the cost matrix made of negative IoUs. During this short-term association between adjacent frames, an association is kept as long as the IoU is higher than a threshold θ_s.

6.3.2 Creation of a batch of tracklets

Once tracklets are created, as illustrated in Figure 6.1, we define a batch of tracklets as follows, without loss of generality :



(a) (b)

FIGURE 6.1 Creation of a batch of tracklets with two temporal windows used during the training. The two frames of reference are f_P=4 and f_F=6 and the temporal windows are of length t_P=3 and t_F=2 frames. a) The set of past and future tracklets contains each four tracklets. Gray detections are completely ignored in this batch of tracklets. b) The matrix $\mathbf{Y}$ indicates whether a pair is positive (1), negative (0) or ignored (?). Best viewed in color.

— two frames, f_P (reference frame of the past) and f_F (reference frame of the future), are selected such that $t_G \times fps \geq f_F - f_P - 1 \geq 0$, where t_G is the maximal temporal gap between tracklets and fps the number of frames per second in the video ;

— sub-tracklets of the video that appears between the frames $f_P - t_P \times fps$ and f_P form the set of past tracklets $\mathcal{S}_P$, and those between the frames f_F and $f_F + t_F \times fps$ form the set of future tracklets $\mathcal{S}_F$, where t_P (resp. t_F) is the maximal temporal duration on tracklets of the past (resp. future) ;
— from sets $\mathcal{S}_P$ and $\mathcal{S}_F$ containing respectively n_P and n_F tracklets, we create a matrix $\mathbf{Y} \in \mathbb{R}^{n_P \times n_F}$ that indicates if a pair of sub-tracklets $(T^i, T^j) \in \mathcal{S}_p \times \mathcal{S}_f$ corresponds to the same object.

Any pair of sub-tracklets is either positive, negative or ignored. A positive pair is obtained if the two sub-tracklets belong to the same ground-truth object, or are extracted from the same tracklet. A negative pair is obtained if the two sub-tracklets do not belong to the same ground-truth object or their full tracklets have some temporal overlap. The remaining pairs are labeled as ignored, for instance if one tracklet falls into some ignored regions or when the two sub-tracklets do not match with any ground-truth object.

6.3.3 Neural Network Architecture of TWiX

Figure 6.2 illustrates the TWiX model. The inputs of TWiX are two sets of tracklets : those of the past and those of the future. A tracklet T of length W is described by two sequences : a matrix of coordinates $\mathbf{C} \in \mathbb{R}^{W \times 4}$ and a vector of timestamps $\mathbf{T} \in \mathbb{R}^{W}$. We create *pairs* of tracklets by combining each tracklet from the past with those from the future. The motivation of using pairs of tracklets instead of tracklets is to get more discriminative features. Indeed, in the case of an occlusion, coordinates of bounding boxes of involved objects are very similar, so the representations of each tracklet may also be very similar if considered *independently*. By considering all pairs of tracklets, it is possible to get more discriminative representations for each pair by considering the *whole context*. We note that considering pairs is similar that what is done when computing IoU, which is very effective to associate detections.

After concatenating coordinates for each pair, they are normalized following a minmax scaler such that they are strictly between -1 and 1. We decided to use a minmax normalization instead of a normalization based on the full size of the image to generate more diversified training data where coordinates are changing depending on the position of other objects. These spatial coordinates are then linearly mapped to a D-dimensional vector, where D is the dimension of the representation.

Within the pair, temporal information is added using a fixed positional encoding based on sine and cosine functions [50]. This temporal information is the temporal distance of each observation with the first observation of the second track. Then, a `[CLS]` token is concatenated for each pair of tracklets [168]. A Transformer Encoder, named Intra-Pair,

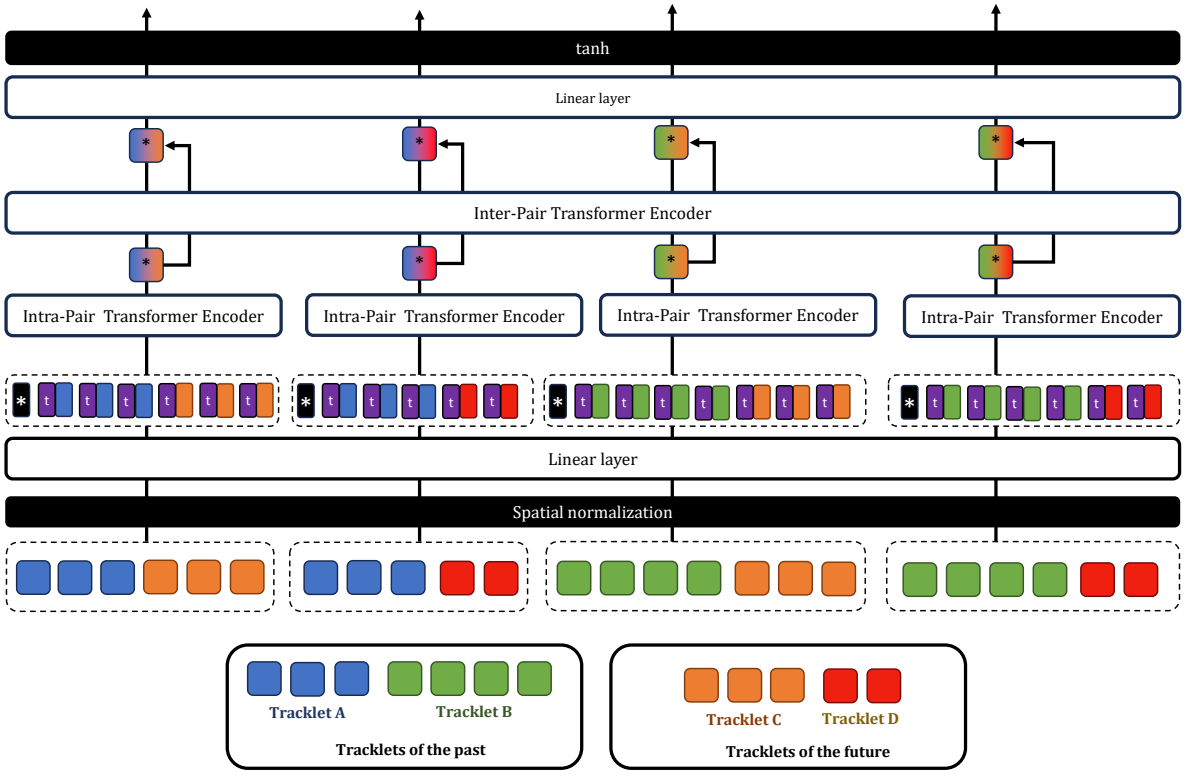


FIGURE 6.2 Architecture of TWiX (read from bottom to top). First, pairs of tracklets are normalized and linearly projected then encoded with a Transformer where attention is applied on the temporal dimension. Then, refined representations are obtained with a second Transformer which pays attention to all other pairs. Finally, a linear layer and a hyperbolic tangent function are used to compute an affinity score for each pair. Best viewed in color.

extracts a representation for each pair where the attention spans on the temporal dimension. Finally, a second Transformer Encoder, named Inter-Pair, extracts a representation for each pair taking into account the representations of all other pairs. These second representations are added to the first ones using a skip connection. At this step, we have a tensor of shape $n_P \times n_F \times D$. To have an affinity matrix, this tensor is sent to a linear layer and a hyperbolic tangent activation function to obtain a matrix of shape $n_P \times n_F$ with values between -1 and 1.

We note that for contrastive learning, the loss function is usually computed using the cosine similarity of embeddings [85]. Here, since the pairs are computed inside the model and not during the back-propagation step, we use a hyperbolic tangent function to obtain affinities between -1 and 1.

6.3.4 Contrastive loss

The output of TWiX is an affinity matrix between all tracklets of the past and of the future. The weights of TWiX, are learned by minimizing a bidirectional contrastive loss, $\mathcal{L}_{bdrC}$, defined as the sum of the forward, $\mathcal{L}_{fwdC}$, and the backward contrastive loss, $\mathcal{L}_{bwdC}$.

Formally, the contrastive loss function l_C for a single positive pair with score s^+ relatively to a set of negative pairs $\mathcal{N}$ of size N_{neg} is defined as follows,

$$l_C(s^+, \mathcal{N}) = \log\left(1 + \frac{B}{N_{neg}} \sum_{s^- \in \mathcal{N}} exp\left[-\frac{s^+ - s^-}{\tau}\right]\right), \tag{6.1}$$

where τ denotes a temperature parameter and B a batch size parameter.

If $\hat{\mathbf{Y}} = (\hat{y}_{ij})_{\substack{1 \le i \le n_P \\ 1 \le j \le n_F}}$ and $\mathbf{Y} = (y_{ij})_{\substack{1 \le i \le n_P \\ 1 \le j \le n_F}}$ are the predicted affinity matrix and the ground truth matrix, N^+ the number of positive pairs in $\mathbf{Y}$, the forward and backward contrastive losses are defined as follows,

$$\mathcal{L}_{fwdC}(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{N^+} \sum_{\substack{i,j \\ y_{ij}=1}} l_C(\hat{y}_{ij}, \{\hat{y}_{il} | 1 \le l \le n_F, y_{il} = 0\}), \tag{6.2}$$

$$\mathcal{L}_{bwdC}(\hat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{N^+} \sum_{\substack{i,j \\ y_{ij}=1}} l_C(\hat{y}_{ij}, \{\hat{y}_{kj} | 1 \le k \le n_P, y_{kj} = 0\}), \tag{6.3}$$

The bidirectional loss is finally defined as,

$$\mathcal{L}_{bdrC} = \mathcal{L}_{fwdC} + \mathcal{L}_{bwdC}. \tag{6.4}$$

In contrast to typical contrastive frameworks used for visual pretraining [85], here the number of tracklets inside the batch is not fixed, hence the shape of the ground truth matrix is also not fixed. So we scale the number of negative pairs with the batch size parameter B ($B \gg N_{neg}$) to simulate a large batch size.

The intuition behind the contrastive loss is to facilitate the learning of discriminative features. This ensures that the affinity between two tracklets referring to the same object is higher than the affinities between the first tracklet and all other tracklets. Introducing bidirectionality increases the robustness of the calculated affinities by incorporating both tracking and reverse tracking aspects.

6.3.5 Tracking with TWiX

A trained TWiX module outputs an affinity matrix between two sets of tracks. This module can replace any module to compute similarities, such as IoU measure, in any tracking pipeline. Hence, our TWiX module can create tracklets without using IoU.

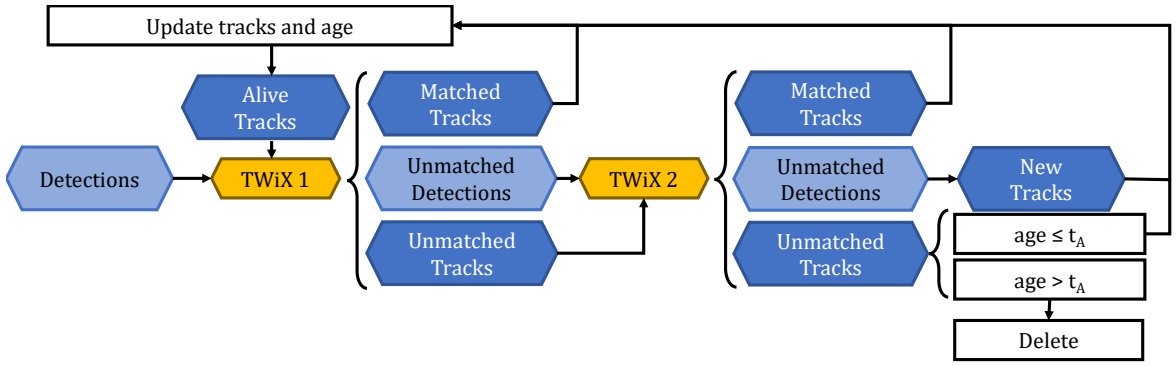


FIGURE 6.3 Our tracker C-TWiX use a cascade matching pipeline for tracking. The BIoU-computed matrix in C-BIoU is replaced by our TWiX module. Best viewed in color.

For online tracking, we use a MOT pipeline similar to cascade matching as described in C-BIoU [42] by only replacing the BIoU-computed matrix by the one obtained with the TWiX module. In this pipeline, as illustrated in Figure 6.3, the first TWiX module matches alive tracks (as the set of past tracklets $\mathcal{S}_P$) with detections (as the set of future tracklets $\mathcal{S}_F$) , then the second TWiX module matches unmatched tracks (as the set of past tracklets $\mathcal{S}_P$) with unmatched detections (as the set of future tracklets $\mathcal{S}_F$). The Hungarian algorithm is used for both matching, if the score is higher than a threshold θ_1 for the first match and θ_2 for the second one. We create new tracks from unmatched detections whose score are higher than θ_T. Tracks are killed when their age is higher than t_A.

We note that the TWIX module can also be applied for offline tracking. In that case, we need to adapt the association pipeline by first associating detections between adjacent frames to create tracklets. Then a second matching would associate tracklets as long as their affinity is high.

As the SOTA methods considered for comparison were developed only for online tracking, in order to have a fair comparison with them, we only applied the online tracking pipeline in the following experiments.

6.4 Experiments

We conducted experiments on three datasets focused on pedestrians and cars. MOT17 is a popular dataset with 7 training videos and 7 test videos [73]. They contain scenes with simple and linear movements of humans. The camera is either fixed or mounted on a car or carried by a pedestrian. DanceTrack dataset was recently proposed to be more challenging : targets have similar appearance, are subject to severe occlusions and have irregular motions [8]. It contains 40 videos for training, 25 for validation and 35 for test. Finally KITTIMOT contains videos of cars and pedestrians [21]. A camera is mounted on top of a car, driving through a city. This dataset is challenging because the videos are recorded at 10 FPS, whereas it is 20 for DanceTrack and between 14 and 30 for MOT17.

We follow the common practices in multi-object tracking on MOT17 and KITTIMOT where official validation sets are not provided. We split the full training sequences of MOT17 into two halves for training and for validation following Zhou et al [19]. And we use the splits from KITTIMOTS, the pixel-level variant of KITTIMOT, to create the training and the validation sets of KITTIMOT, following Luiten et al [11].

The quality of a tracker is measured on its ability to detect objects of interest and on its ability to keep the identity consistent throughout the video. We adopted the higher order metric for multi-object tracking (HOTA) to evaluate the quality of the tracking [11]. This metric was introduced to complement the MOTA [10] metric that takes into account the quality of the detections over the quality of association. The HOTA metric decomposes into the detection accuracy (DetA) and the association accuracy (AssA). Nevertheless, we provide also the MOTA and IDF1 [79] as additional metrics.

6.4.1 Implementation details

For a fair comparison, we used the detections from YOLOX [28] with weights provided by ByteTrack on the datasets DanceTrack and MOT17. Following Cao et al [48], we use the detections from PermaTrack [170] on the KITTIMOT dataset. For all datasets, only detections with a score higher than 50% and bigger than 128 pixels are kept. Data tracks are made using $\theta_s = 0.15$.

On MOT17 and KITTIMOT, after the selection of hyper-parameters on the validation set, we re-trained the two modules on the full training sets.

TWiX extracts representations for pairs of tracks taking into account their relative surroundings. However, these surroundings depend on the matching step. In the pipeline based on cascade matching, we noticed that associations at the first matching occurred mainly between

adjacent tracklets, when the temporal gap is zero. That is why we train the TWiX module with batches of tracklets with $t_G = 0$ for the first matching step and with a non-zero t_G for the second step. We note that to train the TWiX module for the second matching, we only consider batches where f_P (resp. f_F) corresponds to an end (resp. a beginning) of a tracklet. Indeed, during a partial occlusion, spatial coordinates experience a disturbance because the size of a bounding box decreases. Since no appearance information is used, such a signal is important to learn how to associate tracklets under occlusion. Other hyper-parameters are described in Table 6.1 for each dataset. Please note that only one frame is used for the tracklets of the future, as they are detections.

Tableau 6.1 Hyper-parameters for each dataset used during the training and inference steps.

Parameter	DanceTrack	MOT17	KITTIMOT
Maximal temporal gap t_G	1.6 sec	0.8 sec	0.8 sec
Past window size t_P	0.8 sec	0.4 sec	0.4 sec
Future window size t_F	$\frac{1}{fps}$ sec	$\frac{1}{fps}$ sec	$\frac{1}{fps}$ sec
Matching thresholds (θ_1, θ_2)	$(-0.5, -0.2)$	$(0.9, -0.5)$	$(0.4, -0.6)$
Maximum age t_A	1.6 sec	0.8 sec	0.8 sec
Minimal score θ_T	90%	70%	50%

As for the TWiX architecture, the hyperparameters are the same for the two Transformer encoders : the dimension size is 32, the number of heads is 16 and the dimension of the feedforward layer is 32. The batch size and temperature in the loss are respectively $B = 1024$ and $\tau = 0.1$. The model is trained during 30 epochs with the Adam optimizer and cosine learning rate decay. The only difference between the first and the second TWiX modules is that the first one is trained with single-layer Transformers and a learning rate of $1e-4$, and the second with Transformers with four layers and a learning rate of $1e-3$.

6.4.2 Main results

Tables 6.2, 6.3 and 6.4 contain respectively the performance measures of our tracker C-TWiX (TWiX in a cascade matching algorithm) on DanceTrack, MOT17 and KITTIMOT datasets. By analyzing the results, we can first notice that our tracker C-TWiX outperforms other appearance-free trackers on DanceTrack. This dataset is particularly challenging due to the irregular motions of persons. Previous methods employ linear filters, such as a Kalman filter or a linear model, to estimate the new position. Our tracker does not use such assumptions, which are not always correct, enabling it to improve the association step, increasing the AssA

score by 1.8 points and the HOTA by 1.5 points. Only UCMCTracker has a better HOTA. However, this tracker requires to manually select the compensation parameters on each sequence individually. Our tracker C-TWiX does not require such sequence-level adjustment.

TABLEAU 6.2 Performance on the test set of DanceTrack. Only trackers using the detections from ByteTrack and using only coordinates are shown. **Bold red** and *italic blue* indicate respectively the first and second best methods within each category.

Method	HOTA	DetA	AssA	MOTA	IDF1
METHODS USING MOTION COMPENSATION					
SparseTrack [171]	*55.5*	**78.9**	*39.1*	**91.3**	*58.3*
UCMCTrack [38]	**63.4**	NA	**51.1**	*88.8*	**65.0**
METHODS NOT USING ANY MOTION COMPENSATION					
SORT [20]	50.0	75.5	33.2	90.4	52.0
DeepSORT [37]	45.6	71.0	29.7	87.8	47.9
ByteTrack [67]	51.9	80.1	33.8	90.9	52.0
OC-SORT [48]	54.6	80.4	40.2	89.6	54.6
MotionTrack [172]	52.9	80.9	34.7	91.3	53.8
C-BIoU [42]	*60.6*	*81.3*	*45.4*	**91.6**	*61.6*
C-TWiX (ours)	**62.1**	**81.8**	**47.2**	*91.4*	**63.6**

On MOT17, the results of our tracker C-TWiX are on par with other trackers that do not rely on a camera motion compensation technique. Since this dataset contains many sequences with a moving camera, estimating the motion of the camera particularly helps the tracker.

Moreover, on KITTIMOT, where the objects of interest are cars and pedestrians, our tracker C-TWiX outperforms other trackers on cars, even those using motion compensation methods, and gets competitive results on pedestrians. Permatrack reaches excellent results on cars to the detriment of pedestrians, while OC-SORT shows the opposite trade-off. With C-TWiX, the average HOTA score increases by 0.6 point. Even if the framerate on this dataset is low, TWiX manages to correctly associate pedestrians, which is the hardest class. Indeed, they have a vertically elongated shape but move mainly on the horizontal axis that makes the overlap of bounding boxes between adjacent frames very low.

Finally, our tracker is fast running at 320 Hz on KITTIMOT, 300 Hz on DanceTrack and 50 Hz on MOT17 on a single GeForce RTX 2060 with 6 GB RAM (without considering the detection part).

TABLEAU 6.3 Performance on the test set of MOT17. Only trackers using the detections from ByteTrack and using only coordinates are shown. **Bold red** and *italic blue* indicate respectively the first and second best methods within each category.

Method	HOTA	DetA	AssA	MOTA	IDF1
METHODS USING MOTION COMPENSATION					
BoT-SORT [173]	*64.6*	NA	NA	*80.6*	*79.5*
SparseTrack [171]	**65.1**	**65.3**	**65.1**	**81.0**	**80.1**
UCMCTrack [38]	64.3	NA	*64.6*	79.0	79.0
METHODS NOT USING ANY MOTION COMPENSATION					
SORT [20]	63.0	64.2	62.2	80.1	*78.2*
ByteTrack [67]	63.1	*64.5*	62.0	*80.3*	77.3
OC-SORT [48]	*63.2*	63.2	*63.2*	78.0	77.5
MotionTrack [172]	60.9	NA	59.4	76.5	73.5
C-BIoU [42]	**64.1**	**64.8**	**63.7**	**81.1**	**79.7**
C-TWiX (ours)	63.1	64.1	62.5	78.1	76.3

6.4.3 Ablation study

In the following, we evaluate the effect of our choices in the design of the architecture of the TWiX module, measure the performance of the tracker C-TWiX on ground-truth detections, and visualize how the affinity between two bounding boxes evolves with regard to their relative position.

Oracle detections

Similarly to previous studies [8, 42], we conducted an experiment by replacing the detections by the ground-truth annotations, to evaluate only the association component of the tracking. Table 6.5 indicates that our C-TWiX tracker surpasses all other methods based on positions or motion on the DanceTrack dataset. The HOTA score is improved by 0.4 point compared to the previous best method and the DetA by 1 point, thanks to a better association. Surprisingly, even if TWiX is trained on data generated with IoU as pseudo-labels, it significantly beats IoU on HOTA showing the generalization ability of TWiX.

Inter-Pair Transformer Encoder

In the architecture of the TWiX module, the Inter-Pair Transformer Encoder aims to enhance the representation of the pair embeddings. To verify this assertion, we trained the TWiX modules with and without this second Transformer Encoder. Then, we evaluate the HOTA

TABLEAU 6.4 Performance on the test set of KITTIMOT. Only trackers using the detections from Permatrack and using only coordinates are shown. **Bold red** and *italic blue* indicate respectively the first and second best methods within each category.

	car			pedestrian		
Method	HOTA	AssA	MOTA	HOTA	AssA	MOTA
METHODS USING MOTION COMPENSATION						
Permatrack [44]	**77.4**	**77.7**	**90.9**	*47.4*	*43.7*	*65.1*
UCMCTrack [38]	*77.1*	*77.2*	*90.4*	**55.2**	**58.0**	**67.4**
METHODS NOT USING ANY MOTION COMPENSATION						
OC-SORT [48]	*74.6*	*74.5*	*87.8*	**53.0**	**57.8**	*62.0*
C-TWiX (ours)	**77.6**	**78.8**	**89.7**	*52.4*	*54.4*	**65.0**

TABLEAU 6.5 Performance on the validation set of DanceTrack using oracle detections. **Bold red** and *italic blue* indicate respectively the first and second best methods.

Loss function	HOTA	DetA	AssA	MOTA	IDF1
IoU [8]	72.8	**98.9**	53.6	98.7	63.5
IoU + Motion [8]	69.4	87.9	54.8	99.4	71.3
SORT [20]	67.6	86.6	52.8	98.1	69.6
OC-SORT [48]	79.1	97.7	*64.0*	*99.6*	76.1
C-BIoU [42]	*81.7*	97.6	**68.4**	99.3	**80.5**
C-TWiX (ours)	**82.1**	*98.6*	**68.4**	**99.7**	*78.1*

score at different thresholds (θ_1, θ_2) on the validation set of KITTIMOT. Figure 6.4 shows the heatmap of the HOTA score at different thresholds. Specifically, when $\theta_1 = 1$ (resp. $\theta_2 = 1$), only the second (resp. first) matching step is on, rejecting all possible matching at the first (resp. second) association step. Without the Inter-Pair Transformer Encoder, the HOTA score of cars drops by 6 points in the case of a pipeline with only the first matching step ($\theta_2 = 1$). This means that the quality of the first TWiX module, operating on adjacent tracklets, becomes poor at discriminating positive pairs from negative one. In that situation, since coordinates of bounding boxes are close, the absence of a layer interacting with all pairs degrades the quality of the module.

However, in the case where only the second matching step is on ($\theta_1 = 1$), the absence of the Inter-Pair Transformer Encoder does not provoke a drop in HOTA score. Since the second TWiX module was trained on harder batches ($t_G \neq 0$) and is deeper (4 layers instead of 1), its discriminative power is higher than the first one, at the cost of slower calculation time.

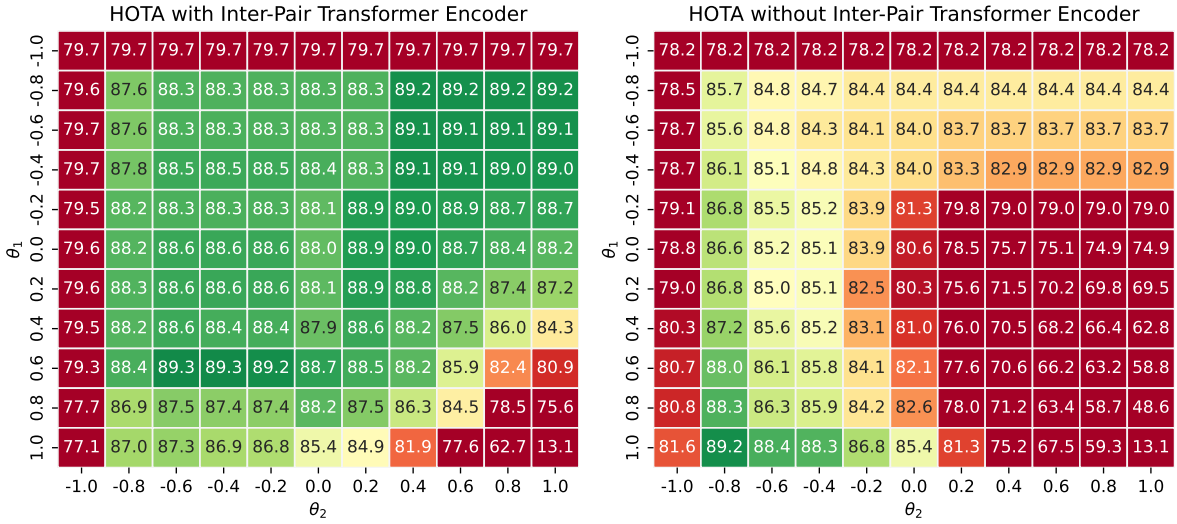


FIGURE 6.4 Comparison of HOTA on the validation set of KITTIMOT-car at different level of matching regarding the presence of the Inter-Pair Transformer Encoder (left) or not (right).

The Inter-Pair Transformer Encoder is therefore less essential in that case.

Nevertheless, adding the Inter-Pair Transformer Encoder increases the area in the (θ_1, θ_2) space with HOTA over 89. This results in a green zone which covers a larger part, whereas without this encoder, such a HOTA score is reachable only when θ_1 equals 1. In conclusion, the Inter-Pair Transformer Encoder makes the tracker more robust with respect to the selection of the hyper-parameters (θ_1 and θ_2), and also makes the tracking faster since the first TWiX module can rely on a single-layer Transformer.

Loss Function

The TWiX modules are trained with a bidirectional contrastive loss (Equation 6.4). The use of such a function is motivated to force a positive pair to have a higher affinity than any other negative pair within the same row or same column in the affinity matrix. In order to validate the choice of the bidirectional contrastive loss, we conducted an experiment with six other loss functions, keeping all the other hyper-parameters identical, except θ_1 and θ_2.

Experiments are conducted on the KITTIMOT and MOT17 datasets and reported in Figure 6.5. The parameters θ_1 and θ_2 are selected with a grid search for each dataset and object class.

First, since the affinity matrix returns values between -1 and 1 for each pairs of tracklets,

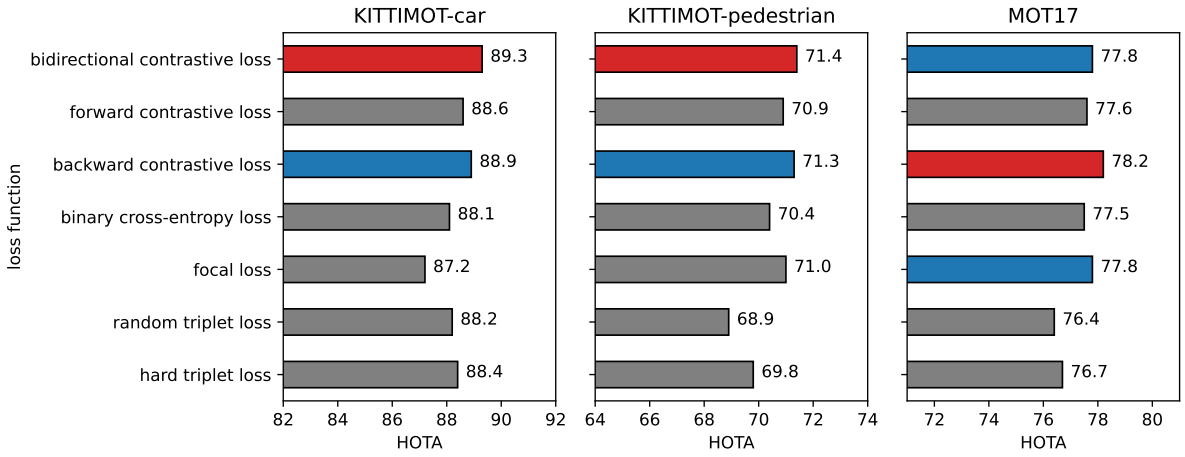


FIGURE 6.5 HOTA scores on KITTIMOT and MOT17 validation sets with regard to the loss function. Red and blue indicate respectively the first and second best methods.

we use the binary cross-entropy after linearly scaling the values to the range from 0 to 1. Despite the unbalanced ratio between the positive and negative pairs (1 :20), using the binary cross-entropy loss reaches decent results with a drop of respectively 1.2, 1.0 and 0.3 points in HOTA score on respectively KITTIMOT cars, pedestrians and on MOT17 compared to the use of our bidirectional contrastive loss.

The focal loss was introduced to overcome the unbalanced data ratio [174]. This improves the HOTA on pedestrians on KITTIMOT and MOT17 by 0.6 and 0.3, but decreases it on cars by 0.9 point compared to the use of the binary cross-entropy loss. However, these results are all lower than those obtained with our contrastive loss.

Both binary cross-entropy and focal losses compute the loss for each pair regardless of other pairs in the affinity matrix. Triplet loss aims at maximizing a positive pair affinity compared to a single negative pair, over a certain margin. For each positive pair in the matrix, we randomly select a negative pair on the same row or same column : this is the random triplet loss. To take into account harder negatives, as a critical component of contrastive learning [166], we select the negative pair on the same row or column which has the highest affinity : this is the hard triplet loss. We notice a drop in performance with both losses compared to our bidirectional contrastive loss. Using the hard negative instead of a random negative increases the HOTA between 0.2 and 0.9 : making the learning harder by forcing the TWiX module to discriminate harder cases improves the performance of the model. But considering all negatives, like in our contrastive loss, improves the quality of our association module.

Finally, since the bidirectional contrastive loss is the sum of the forward and the backward

contrastive losses, we compared it with its two components. Except on MOT17 for the backward version, our loss function beats the forward and the backward versions. This validates that a signal containing both negative from the same row (forward version) and from the same column (backward version) improves the learning by providing harder cases, since the model needs to discriminate a positive pair from all other negative pairs.

Visualization of TWiX Self-Affinity Maps

Contrarily to other IoU-based methods, TWiX takes into account the whole neighborhood of a bounding box to compute its affinities. That is why, in practice, it is hard to visualize the affinity matrix for TWiX between two sets of detections. Moreover, TWiX can leverage the temporal aspect, what other methods cannot do. Therefore, we propose, in a matter a simplicity, to measure the self-affinity of a bounding box : given a box, we compute the affinity between itself and a translated version. The translation is added on the box vertically and horizontally. The obtained map indicates the locations where the affinity is the highest.

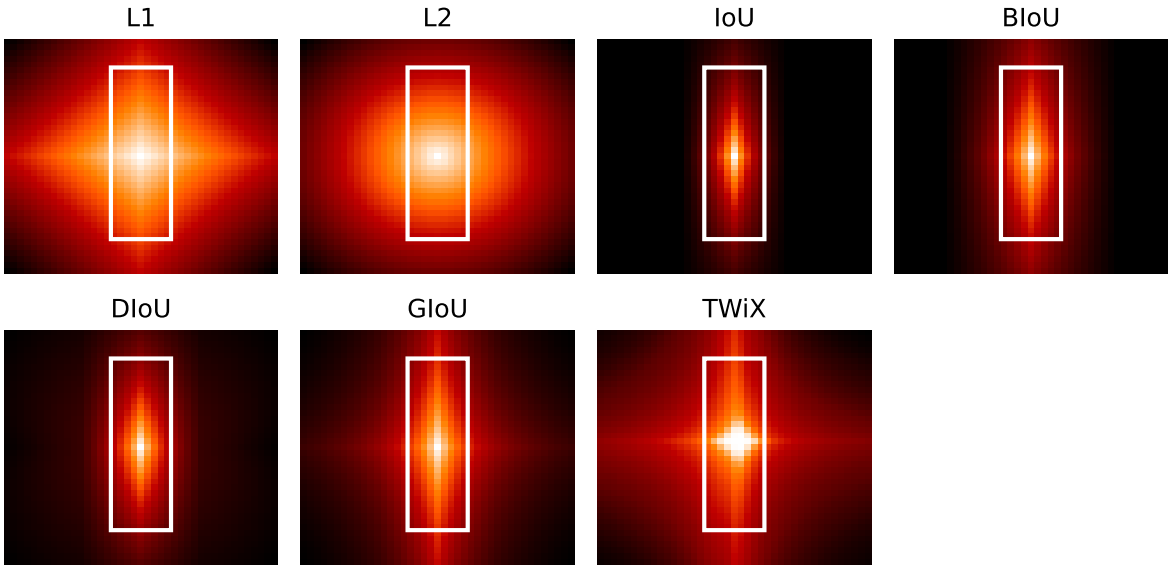


FIGURE 6.6 Self-affinity maps of several model-based methods and TWiX. The affinity between the white box of reference and its translated version is indicating by the color at its translated center position. The whiter, the higher the affinity is.

Figure 6.6 illustrates these maps for six model-based methods and TWiX, which is data-based. First, for the distance L_1 and L_2, symmetrical diamond-shaped and circular maps are obtained respectively, ignoring the shape of the original box. For IoU and the buffered version BIoU [42], we notice that the map extends in the predominant direction of the box,

here the vertical axis. This behaviour is not desired because the motion of an object is not related to its shape. Moreover, the map reaches zero at an intermediate distance, where the overlap between the boxes is empty. Methods such as DIoU [40] and GIoU [39] solve this last issue, but a preferential direction is still present based on the shape of the object. Finally, for TWiX, not only the map exhibits two preferential directions but also, it does not reach zero at an intermediate distance. This behaviour is desired because it means that TWiX learns that objects mainly move vertically or horizontally, the actual direction of motion.

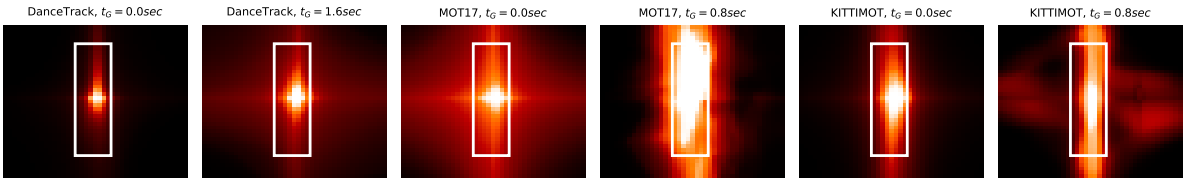


FIGURE 6.7 Self-affinity maps of TWiX on different datasets and with different maximal temporal gaps t_G

Moreover, TWiX adapts to the dataset (framerate, object motion, camera motion, object size, etc.) and to the maximal temporal gap allowed. Figure 6.7 illustrates this. In the case of a dataset with high framerate, the region with a high affinity will be smaller indicating small motion, such as in DanceTrack ($fps = 30$), contrarily to KITTIMOT ($fps = 10$) where a bigger region is observed. And allowing a higher temporal gap t_G enlarges the region with a high affinity, expanding the search area for a possible match.

6.5 Conclusion

In this work, we proposed a contrastive framework to learn representations on pairs of tracklets for MOT. Contrastive learning has shown promising results to learn representations in textual data, images and online MOT. To the best of our knowledge, this is the first work exploiting contrastive learning on the association step of a tracker solely based on coordinates. Our framework creates batches of tracklets that are later encoded with two Transformer Encoders. The first one extracts representation for a pair of tracklets and the second Encoder learns to enhance these representations by paying attention to every pair of tracklets. Experiments on multiple datasets show that our tracker C-TWiX outperforms previous methods on DanceTrack and KITTIMOT and is on par on MOT17. Contrarily to other IoU-based approaches, our module TWiX is able to learn motion from tracklets adjusting the search area for each object.

Even if our module TWiX requires to create all pairs of tracklets, resulting in a bi-quadratical

computation during the attention mechanism, our tracker C-TWiX is able to track objects in real time.

Acknowledgments

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC) [funding reference number RGPIN-2020-04633].

CHAPITRE 7 DISCUSSION

Dans cette section, seront discutés différents sujets relatifs à la tâche de suivi multi-objets, y compris les difficultés rencontrées lors de la thèse et quelques solutions.

7.1 Mesures de performance MOTA versus HOTA

L'une des difficultés rencontrées est la mesure de la qualité d'un algorithme de suivi. La mesure MOTA a été pendant très longtemps la mesure de référence. Leichter et Krupta [175] avaient, dès 2013, fait part de leur remarque sur la non-monotonicité des mesures de performance CLEAR (dont MOTA) : il existe des cas où la suppression d'une erreur sans introduction d'une nouvelle erreur entraîne une baisse de la mesure MOTA. Ils avaient alors proposé cinq mesures de performance pour tenir compte de la variété des erreurs, sans toutefois les combiner en une unique mesure. Au contraire, la mesure HOTA respecte la propriété de monotonicité et peut se décomposer en sous-mesures capables d'informer sur des aspects de l'algorithme de suivi.

Lors de la conception de l'algorithme MeNToS, la non-monotonicité de la mesure MOTA rendait difficile la justification de certaines étapes qui amélioraient le suivi visuellement sans entraîner un progrès de la mesure MOTA. La figure 7.1 illustre un cas simple où une bonne association n'améliore pas la mesure MOTA.

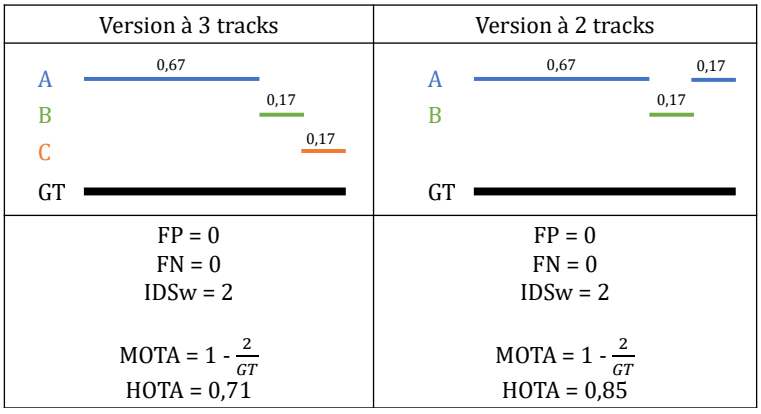


FIGURE 7.1 Illustration d'un cas où une bonne association n'améliore pas le MOTA. Un seul objet (GT) est présent. À gauche, l'algorithme de suivi lui associe trois tracklets tandis qu'à droite, il lui associe les tracks A et B. Avoir fusionné les tracks A et C ne change pas le nombre d'IDSw, conservant la même valeur pour le MOTA. La valeur du HOTA au contraire augmente. Inspirée de Luiten et al [11].

7.2 Comparaison juste entre les algorithmes de suivi

Dans la galaxie des tâches de vision par ordinateur, celle du suivi multi-objets se distingue par plusieurs caractéristiques. Elle dispose de nombreuses bases de données annotées (MOT17, MOT20, DanceTrack, KITTIMOT, UA-DETRAC, UAVDT, etc), de multiples mesures de performance (HOTA, MOTA, IDF_1, etc) et de serveurs dédiés à la soumission de résultats de suivi sur des bases de test, ce qui limite les résultats biaisés. On pourrait ainsi croire que la comparaison entre les algorithmes de suivi est juste.

Toutefois, tous les algorithmes ne sont pas comparables entre eux. Cela dépend de leurs caractéristiques, comme décrit dans la revue de la littérature. Il s'agit des capteurs utilisés, du caractère en ligne ou non de l'algorithme, de la rapidité d'exécution, des informations utilisées (position et/ou apparence), des bases de données externes utilisées en pré-entraînement ou encore des détections lors d'un suivi par détection. Pour ce dernier cas, l'accomplissement d'un nouvel état de l'art s'accompagne souvent de l'utilisation de nouvelles détections plus performantes. Cela fut le cas de SORT (utilisation des détections de Faster R-CNN), IoU-Tracker (détections d'Evolving Boxes) et ByteTrack (détections de YOLOX). C'est pourquoi les algorithmes de suivi par détection devraient employer les mêmes détections pour une comparaison juste et équitable.

Une solution serait de fournir, par exemple tous les ans ou deux ans, de nouvelles détections. Ainsi, la tâche de suivi multi-objets serait davantage centrée sur la sous-tâche d'association et évoluerait de pair avec les progrès dans la tâche de détection. En effet, MOT17 fournit des détections publiques, mais elles ne sont plus le reflet de l'état actuel des détections et ne peut que biaiser les pistes de recherche.

Un autre risque est "l'empoisonnement du puits" : un algorithme peut obtenir injustement d'excellents résultats sur une base de données. En effet, ByteTrack utilise des paramètres différents sur les vidéos de **test** de MOT17 et MOT20, ce qui viole les règles définies implicitement et l'objectif même de la base de test, c'est-à-dire la mesure non biaisée des performances d'une méthode. Ainsi, quiconque souhaitant battre ce record devra soit trouver une nouvelle approche tout en respectant une méthodologie intègre [162] ou bien, reproduire les mêmes biais méthodologiques. Le second choix risque à terme d'empoisonner totalement les résultats publiés sur cette base de données.

7.3 Discussion sur TWiX

Cette section traite uniquement de discussions relatives à la conception de l'algorithme de suivi C-TWiX et de son module TWiX présentés au chapitre 6.

7.3.1 Mesure de la performance durant l'apprentissage de TWiX

Lors de l'apprentissage des modules TWiX, une difficulté fut la mesure de la qualité de l'association. En effet, ces modules retournaient une matrice d'affinité entre deux ensembles de tracks. Pour sélectionner les hyper-paramètres du modèle et identifier les cas de surapprentissage (*overfitting*), une mesure de performance est nécessaire. Tout d'abord, les mesures HOTA ou MOTA ne peuvent pas servir de référence, car elles dépendent directement du pipeline de l'algorithme de suivi, requièrent une sélection des valeurs de seuils d'association et nécessitent la prédiction sur l'ensemble de la base de données, ce qui est chronophage. De plus, dans la conception préliminaire de C-TWiX, le pipeline n'était pas encore fixé.

Une bonne mesure de similarité entre une matrice binaire déséquilibrée et une matrice de prédiction devrait avoir les propriétés suivantes :

1. ne pas dépendre d'un seuil d'association, car expérimentalement, ce seuil dépend des bases de données ;
2. être définie même en l'absence de paires positives ;
3. tenir compte du fait que les données sont déséquilibrées : il y a plus de paires négatives que de paires positives ;
4. permettre l'estimation d'intervalles de confiance ;
5. permettre l'estimation de la robustesse quant au choix du seuil d'association.

Le premier critère disqualifie de nombreuses mesures telles que l'exactitude, la précision et le score F_1 et le deuxième la perte par contraste employée en tant que fonction de perte à chaque batch. Le troisième critère privilégie la mesure d'aire sous la courbe PR (*Precision-Recall*) à celle sous la courbe ROC (*Receiver Operating Characteristic*), car cette dernière retourne une valeur trop optimiste quand les données sont déséquilibrées. La mesure d'aire sous la courbe PR donne la mesure AP (*average precision*). De plus, Boyd et al [176] ont montré qu'il était possible d'estimer empiriquement un intervalle de confiance à 95% pour une valeur de AP de $\hat{\theta}$ sur un échantillon de taille n de la manière suivante :

$$\hat{\theta} \pm 1.96\sqrt{\frac{\hat{\theta}(1-\hat{\theta})}{n}} \tag{7.1}$$

Étant donné que l'aire sous la courbe est comprise entre 0 et 1, $\hat{\theta}$ peut être interprétée comme la probabilité θ associée à une variable suivant la loi de Bernoulli de probabilité θ. Une faiblesse de cet estimateur d'intervalle de confiance est qu'il peut retourner des bornes en dehors de l'intervalle $[0, 1]$.

Ainsi, la mesure AP ne requiert pas de seuil, est adaptée au cas déséquilibré et son incertitude peut être estimée. C'est pourquoi, elle a été employée durant l'apprentissage des modules TWiX pour identifier les cas de sur-apprentissage. Une limitation reste le manque d'information sur la robustesse du choix du seuil d'association lors du suivi. C'est pour cela que la sélection des seuils d'association a été faite à l'aide d'une recherche en grille (*grid search*), comme illustrée dans la figure 6.4.

Plus précisément, pour le premier module TWiX entraîné sur les données d'association à court terme, la valeur AP est calculée sur toutes les valeurs non ignorées de la matrice d'affinité. Nous rappelons que des FP peuvent être présents, car TWiX repose sur les détections d'un détecteur. Ainsi, l'association entre deux FP étant incertaine, la vérité terrain correspondante n'est ni "positive", ni "négative" mais "ignorée". Quant au second module TWiX, cette valeur est calculée sur les paires de tracks (t_P, t_F) où t_P est un track du passé qui se termine et t_F un track du futur qui débute, et non sur toutes les paires.

7.3.2 Corrélation entre les mesures AP et HOTA

Toutefois, la mesure AP est-elle corrélée à la mesure HOTA ? En effet, la mesure AP mesure la qualité de l'association alors qu'HOTA mesure la qualité du suivi. Pour évaluer cela, lors de l'entraînement du module TWIX, la valeur AP est calculée à différentes époques (*epochs*). Pour le suivi, afin de mesurer l'effet à la fois à court terme et à long terme, un pipeline plus simple à une seule étape d'association fut employé, au lieu de l'association par cascade. Les seuils d'acceptation sont identiques à chaque mesure. La figure 7.2 indique les mesures AP et HOTA sur la base de validation de KITTIMOT-ped pour les deux modules TWIX, entraînés à court terme et à long terme.

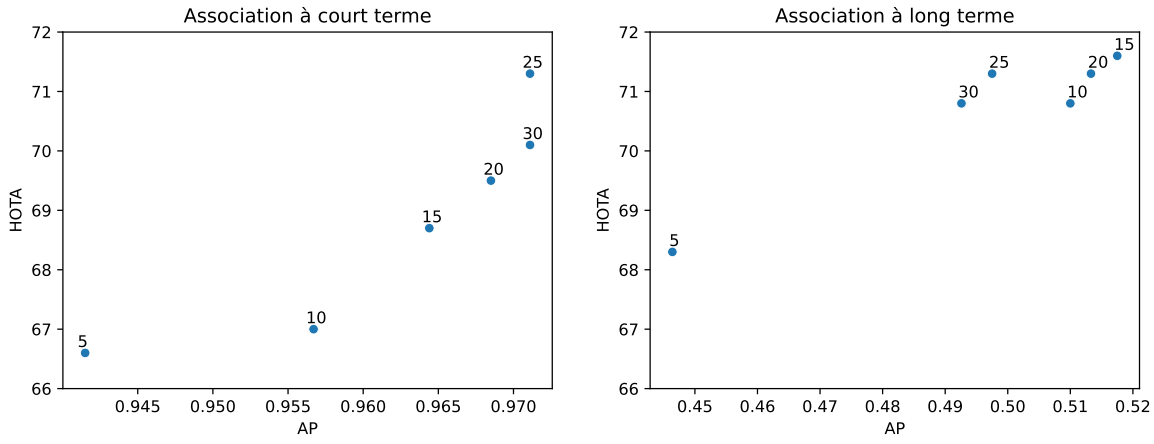


FIGURE 7.2 Relation entre HOTA et AP sur les piétons de KITTIMOT à différentes époques

Il s'avère que les mesures AP semblent bien linéairement corrélées aux mesures HOTA pour les deux modules TWiX. Ensuite, les deux modules peuvent permettre d'atteindre un HOTA de plus de 71 tandis que les valeurs AP sont très différentes : elles sont supérieures à 0.9 pour le premier module et inférieures à 0.6 pour le second. Cela montre que l'apprentissage du second module est plus difficile, mais que les performances atteignables en suivi sont très similaires entre les deux modules.

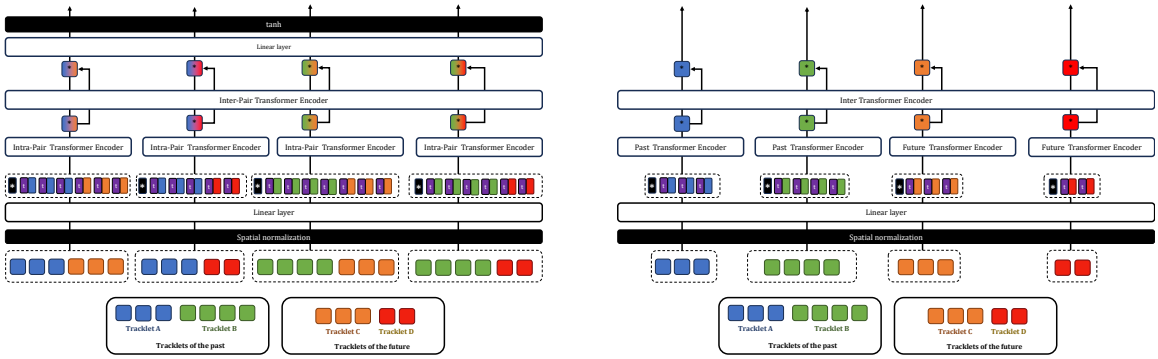
7.3.3 Utilisation de paires

Le module TWiX combine d'abord chaque track du passé avec chaque détection en une *paire* pour en calculer une représentation dans le *Intra-Pair Transformer Encoder*. Cette représentation dépend des coordonnées spatio-temporelles de ces deux éléments, indépendamment des autres objets présents dans la scène. Ces représentations sont ensuite raffinées dans le *Inter-Pair Transformer Encoder* pour tenir compte des positions des autres objets. Il est à noter que cela diffère des autres approches par contraste où une représentation est calculée pour chaque objet *indépendamment* des autres objets. La perte y est ensuite calculée à partir des produits scalaires des représentations, sans passer par une estimation avec une couche tanh. Pourquoi ne pas apprendre les représentations des tracks et des détections sans la création de paires ? Cette approche est plus légère, car elle a une complexité en $O(n^2)$ au lieu de $O(n^4)$ dans le mécanisme d'attention du second Transformer.

Pour répondre à cette question, une seconde version de TWiX a été implémentée. Cette version, nommée TWiX-NoPair, considère ainsi chaque objet individuellement dans un premier Transformer pour en calculer une représentation, qui est ensuite raffinée dans un second Transformer en tenant compte de la position des autres objets. La similarité y est calculée avec le produit scalaire entre les représentations raffinées. La figure 7.3 illustre les architectures de TWiX et de cette variante.

Le tableau 7.1 contient les résultats en AP des versions de TWiX sur la base de KITTI-MOTS évalués sur l'association à court terme et à long terme. Pour les deux associations, la version sans les paires TWiX-NoPair a plus de difficulté à apprendre des représentations discriminantes avec une mesure AP plus basse.

À court terme, les cas les plus difficiles sont les cas où les objets sont spatialement proches. Si les objets sont considérés indépendamment les uns des autres, les coordonnées spatiales étant proches, les représentations obtenues après le premier Transformer seront également assez proches. Le raffinage entrepris par le second Transformer sera rendu difficile par ce manque de discrimination initiale.



(a) Architecture de TWiX (b) Architecture de TWiX-NoPair

FIGURE 7.3 Illustration des architectures des variantes de TWiX : a) TWiX, reprise de la figure 6.2 du chapitre 6 ; b) TWiX-noPair ne crée jamais de paires

TABLEAU 7.1 Performance d'association sur KITTIMOTS sur les données d'association à court terme (STA) et long terme (LTA) en apprentissage et en validation. Les résultats **en gras** indiquent les meilleurs résultats.

Modèle	Données	Perte par contraste ↓	AP-train ↑	AP-val ↑
TWiX-NoPair	STA	1.57	0.83	0.82
TWiX	STA	**0.32**	**0.96**	**0.96**
TWiX-NoPair	LTA	1.29	0.79	0.42
TWiX	LTA	**0.68**	**0.84**	**0.73**

De plus, à long terme, lorsque les paires ne sont pas créées, il n'est pas possible de concevoir des codages de position temporelle (*temporal positional encodings*) qui soient relatifs, car les durées d'occlusion entre les paires ne sont pas constantes. Ainsi, les positions temporelles sont décrites de manière absolue, ce qui réduit la capacité du modèle à généraliser. Cela explique pourquoi la version sans paires TWiX-NoPair souffre davantage d'un sur-apprentissage que la version TWiX à long terme.

7.3.4 Stratégies de création des données d'association

L'apprentissage des modules TWiX repose sur des batches de paires de tracklets. Les tracklets ont été créés en associant successivement les boîtes de deux trames consécutives dès lors que leur IoU dépassait un seuil de 15%, sans track rebirth. Puis, deux modules TWiX ont été entraînés : le premier, moins profond, mais plus rapide, sur les cas où l'écart temporel entre les tracklets du passé et ceux du futur est faible, et le second, plus profond, sur les autres

cas. La figure 7.4 donne des exemples de batches pour ces deux cas.

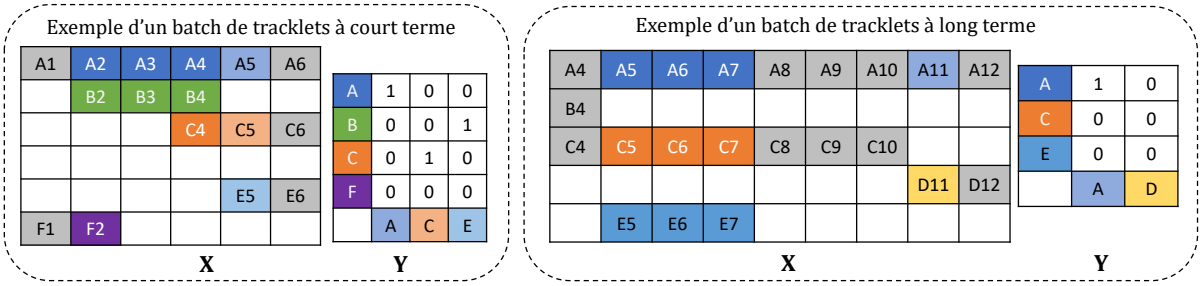


FIGURE 7.4 Illustration de création de batches pour l'association à court et long terme. La notation "C4" désigne la détection de l'objet C à la 4e trame, les couleurs identifient chaque tracklet et la couleur grise indique les détections en dehors des batches. À long terme, la fenêtre du passé se termine à la trame 7 qui correspond à la disparition du tracklet E, et la fenêtre du futur commence à la trame 11 qui correspond à l'apparition du tracklet D.

Dès lors, comment créer les tracklets ? Pour la sous-tâche de détection, deux choix sont possibles : les annotations du jeu de données ou les détections issues d'un détecteur. Pour la sous-tâche d'association, trois choix sont possibles : utiliser les annotations du jeu de données (si les détections proviennent d'un détecteur, il est nécessaire de faire une association trame par trame pour connaître la véritable identité de chaque observation), utiliser une heuristique comme l'association avec IoU pour créer des tracklets (comme IoU-Tracker) ou utiliser cette heuristique en combinaison des annotations pour associer les tracklets entre eux et obtenir ainsi des tracks. Le tableau 7.2 présente les limites des différentes stratégies de création de données.

TABLEAU 7.2 Limitations des stratégies de création de batches de données. Elles portent sur l'absence de données d'association en LTA, un biais de ces données, l'absence d'augmentation de données, des erreurs d'association et la non gestion des faux positifs. Le symbole "✓" indique l'absence de limite, "✗" la présence d'une limite et "-" un cas indéfini.

Détection	Association	Données LTA	Non biaisé	Aug. données	Ass. correctes	FP
GT	GT	✗	-	✗	✓	✗
GT	IoU	✗	-	✗	✗	✗
GT	IoU + GT	✓	✗	✗	✗	✗
Détecteur	GT	✓	✓	✓	✓	✗
Détecteur	IoU	✗	-	✓	✗	✓
Détecteur	IoU + GT	✓	✓	✓	✗	✓
Détecteur	GT + IoU	✓	✓	✓	✓	✓

Tout d'abord, utiliser les annotations des bases de données en tant que tracklet (ligne 1) est problématique, car les occlusions sont absentes sur certaines bases de données. En effet, par exemple dans MOT17, lors d'une occlusion totale, les boîtes englobantes sont tout de même annotées. Dans ce cas, il n'est pas possible de créer des batches de données pour l'entraînement du second module TWiX, c'est-à-dire sur les données à long terme (LTA). De plus, les données souffrent d'un manque de diversité, car il n'y a pas d'augmentation de données avec des FP et des FN.

L'utilisation d'une heuristique pour l'association (ligne 2) ne supprime aucune de ces limitations. Au contraire, cette fois-ci, les associations peuvent parfois être incorrectes. De plus, à l'instar de IoU-Tracker, seuls des tracklets sont générés. Dans ce cas, aucune association n'est faite entre les tracklets ce qui rend impossible la création de données en LTA.

Si les tracklets créés précédemment sont associés à l'aide des annotations (ligne 3), il est possible de créer des batches de données en LTA, mais ceux-ci seront biaisés. En pratique, au moment d'une occlusion, les coordonnées des boîtes englobantes sont perturbées graduellement. En mesurant les variations de tailles des boîtes englobantes d'objets détectés, il s'avère qu'il est possible de distinguer les tracks subissant une occlusion des autres tracks. La figure 7.5 illustre les rapports de hauteur et de largeur vis-à-vis de la boîte englobante la plus proche du moment d'occlusion (donc la plus à même de subir une forte perturbation). Pour les tracks en dehors d'une situation d'occlusion, ce rapport de taille reste aux environs de 1 : les boîtes conservent les mêmes dimensions. Mais, pour les tracks en situation d'occlusion, une variation croissante apparaît : à deux trames d'une occlusion, les boîtes sont 2% plus grandes et à cinq trames, elles sont plus grandes de 5%. Pour un modèle qui ne repose que sur des coordonnées, un tel signal risque d'entraîner un fossé de domaine entre les données d'apprentissage issues des annotations (donc non soumis à des problèmes d'occlusion partielle) et les données en inférence qui reposent sur des détecteurs.

Pour obtenir à la fois des données non biaisés pour le LTA et une augmentation de données, la meilleure stratégie consiste à utiliser un détecteur. Si les véritables associations sont combinées à un détecteur (ligne 4), il n'est pas possible de constituer des tracklets pour les FP, car aucune véritable observation ne coïncide avec eux. Ainsi, cela biaise les données de tracklets.

Pour palier ce problème, l'utilisation d'une heuristique, comme IoU (ligne 5), permet de créer des tracklets pour tous les objets, y compris les FP. Mais, comme la stratégie de la ligne 2, seuls des tracklets sont obtenus. Cela empêche l'entraînement du second module TWiX sur des données d'association à long terme.

Au final, en créant des tracklets avec IoU puis en les combinant avec les annotations (ligne

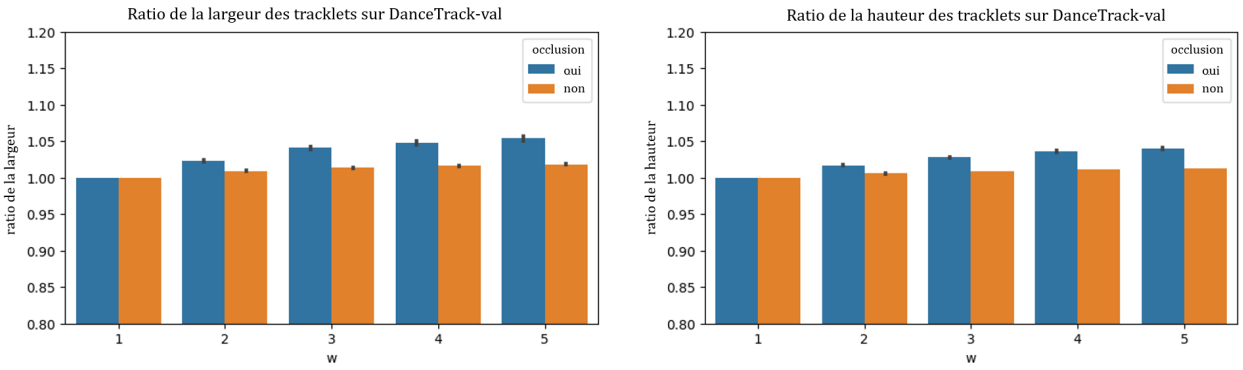


FIGURE 7.5 Illustration de rapports de taille dans les cas d'occlusion (bleu) ou non (orange), selon la distance temporelle à l'occlusion (en trames). Les tailles sont normalisées par rapport à la taille de la boîte la plus proche de l'occlusion ($w = 1$), ce qui explique que les ratios valent 1.00 pour $w = 1$. Plus les boîtes sont proches de l'occlusion, plus elles sont petites.

6), il est possible de corriger ce problème. Dans ce cas, la seule limitation reste la présence de possibles erreurs d'association en raison de IoU. Une manière de corriger cela consiste à utiliser d'abord les annotations, puis IoU (ligne 7) pour les objets restants, à savoir les FP.

Toutefois, la stratégie de la ligne 6 a été privilégiée à celle de la ligne 7. En effet, l'apprentissage des modules TWiX capture les biais internes de chaque base de données : taille des objets, densité, vitesse de déplacement, etc. Or, ils ne sont pas transférables entre jeux de données. Donc, il est nécessaire d'entraîner les modules sur chaque base de données, ce qui nécessite des annotations. La stratégie de la ligne 6 ne requiert qu'une annotation entre *tracklets* alors que celle de la ligne 7 une annotation entre *détections adjacentes*, ce qui est plus laborieux. Ainsi, la stratégie de la ligne 6 sera plus simple à mettre en pratique sur de nouvelles séquences. De plus, il s'est avéré par la suite que l'ajout de mauvaises associations n'avait pas fortement impacté les résultats de suivi, comme l'avait montré le tableau 6.5 du chapitre 6.

Cas pratique : comparaison en HOTA entre les lignes 3 et 6. Les stratégies des lignes 3 et 6 diffèrent par la nature des détections : elles sont issues des annotations pour la première et d'un détecteur pour la seconde. Deux modules TWiX ont été entraînés sur les tracklets générés par ces deux approches sur des données à court terme. Puis, l'inférence a été faite à partir des détections de ByteTrack. Le tableau 7.3 indique les performances en HOTA d'un pipeline avec une seule étape d'association sur plusieurs seuils d'association θ_s. Ainsi, pour l'utilisation des détections GT, l'absence d'augmentation de données et la présence d'un biais dégrade fortement les capacités du module TWIX à associer correctement les tracks, entraînant une baisse en HOTA de 3 points et en rendant l'algorithme moins robuste quant

au choix du seuil d'association.

TABLEAU 7.3 HOTA selon la nature des détections sur MOT17-val, calculé selon plusieurs valeurs de seuils d'acceptation θ_s. Le résultat **en gras** indique le meilleur résultat.

Détection	Association	$\theta_s = -0.8$	$\theta_s = -0.6$	$\theta_s = -0.4$	$\theta_s = -0.2$
GT	IoU + GT	74.1	72.9	70.8	69.3
ByteTrack	IoU + GT	77.0	77.1	**77.6**	77.0

7.3.5 Choix du détecteur pour le suivi

Pour obtenir de bonnes performances de suivi, il faut que les détections soient de bonne qualité. Comment privilégier un détecteur par rapport à un autre ? Pour faire ce choix sur une base de données, une estimation de la borne haute en HOTA peut être faite en associant chaque boîte englobante à sa vérité terrain puis en attribuant la véritable identité à toutes les détections correspondant au même objet. Ceci s'interprète par une suppression de tous les FP et association parfaite des TP à la bonne identité. Une telle étude oracle a été menée partiellement dans la section 5.5.2 du chapitre 5.

Cet algorithme simple a été appliqué sur plusieurs bases de données, plusieurs détecteurs et seuils de détection θ_d. Le tableau 7.4 indique les bornes supérieures en HOTA d'un algorithme associant correctement les identités.

Lorsque les détections sont la vérité terrain, le HOTA obtenu est de 100, la valeur maximale, sauf sur KITTIMOT en raison des arrondis sur les petites boîtes englobantes. De plus, l'utilisation de seuil de détection θ_d distinct permet de mesurer l'écart irrattrapable en fixant 50% au lieu de 10%. Sur MOT17, plus le détecteur est de bonne qualité, plus cet écart est petit. Finalement, les détecteurs (hors GT) utilisés dans les algorithmes de suivi sont ceux qui fournissent le HOTA le plus élevé.

7.3.6 Comparaison entre les algorithmes MeNToS et C-TWiX

Cette thèse contient deux approches pour résoudre le suivi multi-objets en suivant le paradigme de suivi par détection : MeNToS dans le chapitre 5 et C-TWiX dans le chapitre 6.

Tout d'abord, les résultats de ces deux approches ne peuvent pas être comparés, car MeNToS résout la tâche de suivi MOTS alors que C-TWiX s'intéresse à la tâche de suivi MOT. Ensuite, MeNToS se base sur une approche hors ligne alors que C-TWiX associe les détections en ligne, sans aucun post-traitement. Les informations sont également différentes : MeNToS exploite

les réseaux à mémoire spatio-temporelle qui utilisent l'apparence et la position des objets alors que TWiX n'exploite que les informations de position. De plus, MeNToS se base sur des réseaux pré-entrainés (RAFT et STM) alors que les modules TWiX n'ont été entraînés que sur les bases de données de suivi. Puis, l'information temporelle est encodée de manière différente : l'association à long terme de MeNToS, qui est construite sur les réseaux STM, se base directement sur les numéros de trames (l'association considère les trames $n-1$, $n-2$ et/ou $n-5$) alors que TWiX convertit les numéros de trames en secondes. Cette dernière approche permet une meilleure généralisation sur des vidéos ayant des taux d'images par seconde différents. C'est pourquoi une comparaison juste des résultats des deux algorithmes de suivi n'est pas possible.

Toutefois, une comparaison analytique des phases d'association est possible. Les deux algorithmes de suivi se fondent sur deux étapes d'association : une à court terme entre les trames adjacentes et une à long terme entre des trames distantes. Dans MeNToS, la première phase utilise un réseau RAFT qui calcule le flux optique entre deux images et la seconde phase utilise les réseaux STM en exploitant plusieurs images. Dans les modules TWiX, les deux phases n'exploitent que les coordonnées spatio-temporelles des objets. Ainsi, les deux algorithmes exploitent de l'information provenant du contexte dans le calcul de la similarité entre deux objets sans l'utilisation de descripteurs visuels. Dans l'algorithme MeNToS, le contexte désigne toutes les images utilisées dans les réseaux RAFT et STM (c'est-à-dire y compris l'apparence de l'arrière-plan, l'apparence des autres objets et leurs positions), alors que dans le module TWiX, le contexte est la position et le mouvement des autres objets.

En enfin, la complexité algorithmique est différente lors de l'inférence. Considérons une vidéo de résolution $H \times W$, contenant N objets dans T trames. Tout d'abord dans MeNToS, la complexité du calcul du flux optique entre deux trames avec RAFT est en $O(HW)$ et celle du réseau STM entre deux tracklets est aussi en $O(HW)$. Ainsi, la complexité de calcul de MeNToS est en $O(HWT+HWN^2)$, car le flux optique est calculé entre les trames adjacentes et les réseaux STM sont exploités entre deux tracklets. Dans C-TWiX, chaque module TWiX a une complexité en $O(N^2)$, mais une utilisation en mémoire en $O(N^4)$ dans le mécanisme d'attention, car elle calcule explicitement la similarité entre des paires d'objets. Ainsi, la complexité totale de C-TWiX est en $O(TN^2)$. La résolution de la vidéo n'intervient pas car aucune information visuelle n'est utilisée dans les modules TWiX. Ainsi, MeNToS éprouve des difficultés pour les vidéos à haute résolution et C-TWiX pour les vidéos à forte densité. Le tableau 7.5 décrit ces deux algorithmes de suivi à partir des composantes décrites dans la revue de littérature. Ainsi, l'algorithme de suivi MeNToS a été construit à partir de modules pré-entraînés et d'heuristiques tandis que C-TWiX présente une approche plus élégante pour l'association de données.

TABLEAU 7.4 Mesure HOTA lorsque les associations sont parfaites selon le détecteur et le seuil de détection θ_d. Les meilleures performances (hors GT) sont **en gras**.

Base de données	Détection	$\theta_d = 10\%$	$\theta_d = 50\%$
MOT17-train	GT	100	100
MOT17-train	ByteTrack	**85.5**	**83.4**
MOT17-train	Faster R-CNN R-50	56.1	51.9
MOT17-train	Faster R-CNN R-101	56.0	51.8
MOT17-train	YOLOX	53.6	45.0
MOTSChallenge-train	GT	100	100
MOTSChallenge-train	Mask R-CNN X-152 + Box2Seg	**77.2**	**75.4**
MOTSChallenge-train	MaskFormer 2 SwinL	-	75.2
MOTSChallenge-train	Cascade Mask R-CNN X-152	-	73.4
KITTIMOT-val-car	GT	97.1	97.1
KITTIMOT-val-car	Permatrack	**90.7**	**90.5**
KITTIMOT-val-car	boîtes de Mask R-CNN X-152 + Box2Seg	-	71.8
KITTIMOT-val-ped	GT	97.5	97.5
KITTIMOT-val-ped	Permatrack	**74.7**	**73.9**
KITTIMOT-val-ped	boîtes de Mask R-CNN X-152 + Box2Seg	-	55.6
KITTIMOTS-val-car	GT	100	100
KITTIMOTS-val-car	Mask R-CNN X-152 + Box2Seg	**88.1**	**87.4**
KITTIMOTS-val-car	MaskFormer 2 SwinL	-	84.8
KITTIMOTS-val-car	Cascade Mask R-CNN X-152	-	84.3
KITTIMOTS-val-ped	GT	100	100
KITTIMOTS-val-ped	Mask R-CNN X-152 + Box2Seg	**74.2**	**72.7**
KITTIMOTS-val-ped	MaskFormer 2 SwinL	-	72.0
KITTIMOTS-val-ped	Cascade Mask R-CNN X-152	-	69.5
DanceTrack-val	GT	100	100
DanceTrack-val	ByteTrack	**85.8**	**84.3**

Tableau 7.5 Descriptions des algorithmes MeNToS et C-TWiX

Caractéristique	MeNToS	C-TWiX
Tâche	MOTS	MOT
En ligne	non	oui
Localisation	masques et cartes de chaleur	boîtes englobantes
STA	flux optique RAFT pré-entraîné	TWiX entraîné sur des données de STA
LTA	réseau STM pré-entraîné	TWiX entraîné sur des données de LTA
Assignation	algorithme hongrois à STA et glouton à LTA	algorithme hongrois
Autres	interpolation linéaire, filtrage des tracks à faible score, filtrage des tracklets à LTA par une heuristique	association par cascade

CHAPITRE 8 CONCLUSION

Ce dernier chapitre présente une synthèse des travaux réalisés lors de cette thèse et présente les limitations des différentes solutions. Des recommandations pour des travaux futurs sont discutées en dernier lieu.

8.1 Synthèse des travaux

Dans le cadre de cette thèse, nous avons développé des algorithmes de suivi multi-objets MOT et MOTS. Ceux-ci reposent sur le paradigme de suivi par détection où l'ensemble des détections est fixé en amont. Nos travaux se sont donc naturellement portés sur la sous-tâche d'association. Celle-ci peut être résolue par une approche considérant la similarité visuelle et/ou la proximité spatiale.

Le premier article s'est concentré sur une méthodologie pour comparer différents descripteurs visuels associés à des mesures d'affinité sur la tâche de suivi multi-objets. L'évaluation proposée mesure la capacité à associer correctement deux observations du même objet à deux instants, à partir d'une représentation vectorielle. Elle a montré que les vecteurs de réidentification associés à la similarité cosinus sont les descripteurs les plus performants pour cette tâche. Elles sont également robustes à l'écart temporel entre les deux instants et robustes à la qualité des boîtes englobantes. Comparativement aux autres descripteurs, les vecteurs de reID sont particulièrement efficaces pour décrire des objets de taille moyenne.

Deux approches sans vecteurs de réidentification ont été proposées dans les deux articles restants. La première est basée sur des masques et des cartes de chaleur tandis que la seconde exploite la position relative entre tous les objets pour le calcul des similarités. Ces deux approches ont été motivées pour surmonter deux limitations des vecteurs de reID.

Premièrement, les vecteurs de réidentification sont sensibles lorsqu'un objet occulte l'objet d'intérêt. Pour tenir compte de ces cas d'occlusion, nous avons proposé une première méthode d'association entre masques binaires dans le deuxième article. Un réseau à mémoire spatio-temporelle pré-entraîné est utilisé pour retrouver la position d'un masque dans une nouvelle image. Une stratégie est ainsi proposée pour mesurer la similarité entre deux tracklets dans le cadre d'un suivi MOTS. Cette méthode s'est révélée plus performante sur deux bases de données et le module d'association s'est montré plus robuste au choix d'un seuil que les descripteurs visuels classiques.

Deuxièmement, les vecteurs de réidentification sont calculés sur chaque objet, indépendam-

ment des autres objets, limitant leur capacité de discrimination lorsque les objets se ressemblent visuellement. Pour tenir compte de la présence du contexte, nous avons proposé une seconde méthode à base d'encodeurs de Transformer pouvant à la fois fonctionner sur des tracklets, en tant que séquence de positions, et sur des ensembles de tracklets. Une stratégie à base de paires de tracklets a été proposée pour mesurer la similarité entre un tracklet et une détection dans le cadre d'un suivi MOT. Cette méthode s'est montrée particulièrement efficace lorsque les mouvements ne sont pas linéaires.

La figure 8.1 présente un résumé de la thèse.

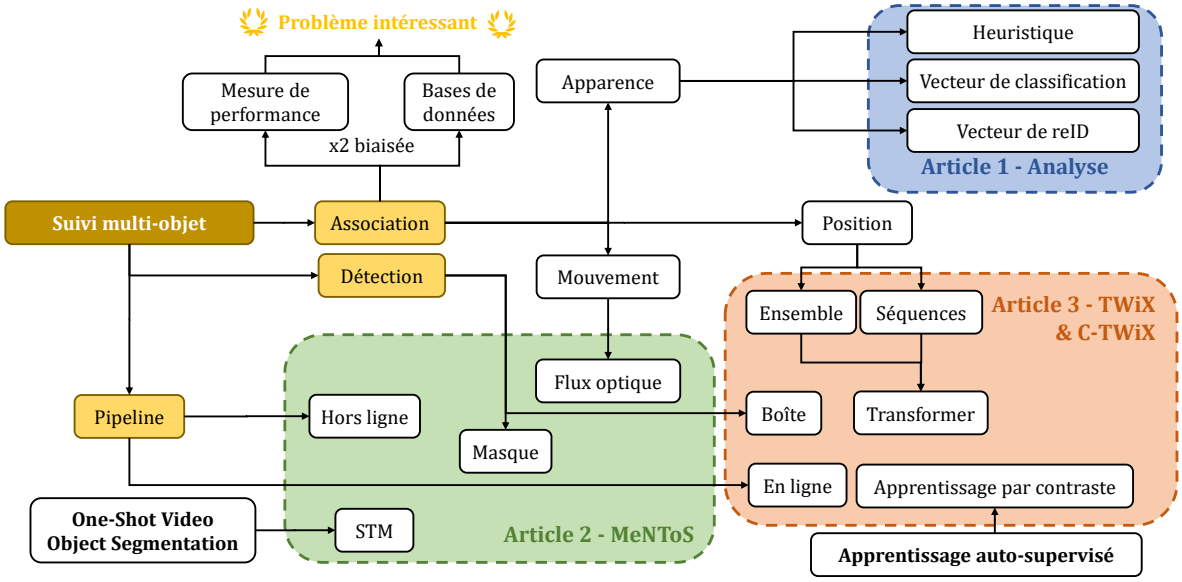


FIGURE 8.1 Représentation graphique de la thèse

8.2 Limitations des solutions proposées

L'étude comparative du premier article porte uniquement sur les descripteurs visuels. Ces représentations peuvent être consolidées en considérant un mécanisme de mémoire qui conserve un historique des précédents descripteurs. Pour des raisons de simplicité, seule la version sans mémoire a été évaluée. Une approche comparative entre des séquences d'apparences serait pertinente.

Ensuite, la méthode proposée avec le réseau à mémoire spatio-temporelle s'est révélée assez lente, car elle fonctionnait à l'échelle du pixel et considérait l'entièreté de l'image. Malgré

les performances de l'algorithme MeNToS, son application sur des images à haute résolution n'est pas envisageable.

Et enfin, la méthode proposée avec TWiX requiert la création de paires de tracklets. Cela entraîne un coût en mémoire en $O(n^4)$ dans le mécanisme d'attention. C'est pourquoi son application sur des bases de données avec une grande densité de personnes, comme MOT20, n'est pas possible en l'état. De plus, la sélection des seuils d'association θ_s et θ_l nécessite une recherche en grille laborieuse.

8.3 Améliorations futures

Les deux algorithmes de suivi MeNToS et C-TWiX reposent tous les deux sur la création de paires de tracklets. Le premier ne les crée qu'après une première étape à court terme, ce qui en limite le nombre, tandis que le second les crée à chaque nouvelle trame. La complexité dans le mécanisme d'attention y est alors en $O(n^4)$ ce qui rend son application inenvisageable lorsque la densité d'objets est élevée. Des travaux portant sur le mécanisme d'attention dans l'estimation de matrices permettraient d'accélérer les calculs en réduisant le nombre de paires considérées.

Ensuite, C-TWiX n'exploite pas l'apparence visuelle dans le calcul des affinités. Or, il est tout à fait possible de calculer de manière distincte plusieurs matrices d'affinité basées sur des informations différentes (apparence, position/mouvement, score, etc) et de les combiner. Actuellement, les stratégies pour combiner des signaux multimodaux consistent en une simple moyenne entre les matrices. Une approche pondérant plus fortement les matrices les plus discriminantes permettrait de s'adapter automatiquement à des cas où l'apparence des objets est similaire par exemple.

Et enfin, un entraînement des modules TWiX de manière dynamique en créant les tracklets au fur et à mesure d'un suivi, et non hors ligne comme actuellement, simplifierait le choix des seuils d'association θ_s et θ_l (par exemple fixés à 0). Cela ne devrait pas trop affecter les données à court terme (qui restent sur des trames consécutives), mais affecterait celles à long terme en réduisant le nombre d'exemples de batches. Cela peut toutefois être compensé par une meilleure représentativité des données de tracklets.

RÉFÉRENCES

[1] S. W. Oh, J.-Y. Lee, N. Xu et S. J. Kim, "Video Object Segmentation Using Space-Time Memory Networks," dans *ICCV*, 2019.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, L. Kai et L. Fei-Fei, "ImageNet : A large-scale hierarchical image database," dans *CVPR*, 2009.

[3] A. Krizhevsky, I. Sutskever et G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," dans *NIPS*, 2012.

[4] Q. V. Le, M. Ranzato, R. Monga, M. Devin, G. Corrado, K. Chen, J. Dean et A. Y. Ng, "Building high-level features using large scale unsupervised learning," dans *ICML*, 2012.

[5] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg et L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, n°. 3, p. 211–252, déc. 2015.

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár et C. L. Zitnick, "Microsoft COCO : Common Objects in Context," dans *ECCV*, 2014.

[7] L. Leal-Taixé, A. Milan, I. Reid, S. Roth et K. Schindler, "MOTChallenge 2015 : Towards a Benchmark for Multi-Target Tracking," *arXiv : 1504.01942*, avr. 2015.

[8] P. Sun, J. Cao, Y. Jiang, Z. Yuan, S. Bai, K. Kitani et P. Luo, "DanceTrack : Multi-Object Tracking in Uniform Appearance and Diverse Motion," dans *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, nov. 2021.

[9] S. Zangenehpour, J. Strauss, L. F. Miranda-Moreno et N. Saunier, "Are signalized intersections with cycle tracks safer ? A case–control study based on automated surrogate safety analysis using video data," *Accident Analysis & Prevention*, vol. 86, p. 161–172, janv. 2016.

[10] R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofolo, D. Mostefa et P. Soundararajan, "The CLEAR 2006 Evaluation," dans *Multimodal Technologies for Perception of Humans*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg : Springer, 2007.

[11] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixe et B. Leibe, "HOTA : A Higher Order Metric for Evaluating Multi-Object Tracking," *International Journal of Computer Vision (IJCV)*, oct. 2020.

[12] F. Yang, X. Chang, C. Dang, Z. Zheng, S. Sakti, S. Nakamura et Y. Wu, “ReMOTS : Self-Supervised Refining Multi-Object Tracking and Segmentation,” dans *CVPR - Workshops*, 2020.

[13] M. Miah, J. Pepin, N. Saunier et G.-A. Bilodeau, “An Empirical Analysis of Visual Features for Multiple Object Tracking in Urban Scenes,” dans *International Conference on Pattern Recognition (ICPR)*, 2021.

[14] M. Miah, G.-A. Bilodeau et N. Saunier, “Multi-Object Tracking and Segmentation with a Space-Time Memory Network,” dans *2023 20th Conference on Robots and Vision (CRV)*, juin 2023, p. 184–193.

[15] P. Bergmann, T. Meinhardt et L. Leal-Taixe, “Tracking Without Bells and Whistles,” dans *ICCV*, 2019.

[16] T. Meinhardt, A. Kirillov, L. Leal-Taixe et C. Feichtenhofer, “TrackFormer : Multi-Object Tracking with Transformers,” dans *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[17] B. X. Chen et J. K. Tsotsos, “Fast Visual Object Tracking with Rotated Bounding Boxes,” dans *ICCV - Workshops*, 2019.

[18] H. Perreault, G.-A. Bilodeau, N. Saunier et M. Héritier, “CenterPoly : real-time instance segmentation using bounding polygons,” dans *ICCV - Workshops*, 2021.

[19] X. Zhou, V. Koltun et P. Krähenbühl, “Tracking Objects as Points,” dans *ECCV*, 2020.

[20] A. Bewley, Z. Ge, L. Ott, F. Ramos et B. Upcroft, “Simple online and realtime tracking,” dans *ICIP*, 2016.

[21] A. Geiger, P. Lenz et R. Urtasun, “Are we ready for autonomous driving ? The KITTI vision benchmark suite,” dans *CVPR*, 2012.

[22] G. Nam, M. Heo, S. W. Oh, J.-Y. Lee et S. J. Kim, “Polygonal Point Set Tracking,” dans *CVPR*, 2021.

[23] B. Xiao, H. Wu et Y. Wei, “Simple Baselines for Human Pose Estimation and Tracking,” dans *ECCV*, 2018.

[24] S. Ren, K. He, R. Girshick et J. Sun, “Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks,” dans *NIPS*, 2015.

[25] J. Redmon, S. Divvala, R. Girshick et A. Farhadi, “You Only Look Once : Unified, Real-Time Object Detection,” dans *CVPR*, 2016.

[26] J. Redmon et A. Farhadi, “YOLO9000 : Better, Faster, Stronger,” dans *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, p. 7263–7271.

[27] ——, “YOLOv3 : An Incremental Improvement,” avr. 2018.

[28] Z. Ge, S. Liu, F. Wang, Z. Li et J. Sun, “YOLOX : Exceeding YOLO Series in 2021,” août 2021.

[29] A. Kirillov, K. He, R. Girshick, C. Rother et P. Dollar, “Panoptic Segmentation,” dans *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, p. 9404–9413.

[30] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár et R. Girshick, “Segment Anything,” avr. 2023.

[31] K. He, G. Gkioxari, P. Dollar et R. Girshick, “Mask R-CNN,” dans *ICCV*, 2017.

[32] X. Wang, T. Kong, C. Shen, Y. Jiang et L. Li, “SOLO : Segmenting Objects by Locations,” dans *ECCV*, 2020.

[33] R. Hu, P. Dollár, K. He, T. Darrell et R. Girshick, “Learning to Segment Every Thing,” dans *CVPR*, 2018.

[34] P. Voigtlaender, J. Luiten et B. Leibe, “BoLTVOS : Box-Level Tracking for Video Object Segmentation,” *arXiv :1904.04552 [cs]*, déc. 2019.

[35] C. Tang, H. Chen, X. Li, J. Li, Z. Zhang et X. Hu, “Look Closer To Segment Better : Boundary Patch Refinement for Instance Segmentation,” dans *CVPR*, 2021.

[36] P. Mahalanobis, “On the Generalised Distance in Statistics,” *Journal of Genetics*, vol. 41, p. 159–193, 1936.

[37] N. Wojke, A. Bewley et D. Paulus, “Simple online and realtime tracking with a deep association metric,” dans *ICIP*, 2017.

[38] K. Yi, K. Luo, X. Luo, J. Huang, H. Wu, R. Hu et W. Hao, “UCMCTrack : Multi-Object Tracking with Uniform Camera Motion Compensation,” janv. 2024.

[39] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid et S. Savarese, “Generalized Intersection Over Union : A Metric and a Loss for Bounding Box Regression,” dans *CVPR*, 2019.

[40] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye et D. Ren, “Distance-IoU Loss : Faster and Better Learning for Bounding Box Regression,” *AAAI*, avr. 2020.

[41] A. Simonelli, S. R. R. Bulò, L. Porzi, M. López-Antequera et P. Kontschieder, “Disentangling Monocular 3D Object Detection,” mai 2019.

[42] F. Yang, S. Odashima, S. Masui et S. Jiang, “Hard To Track Objects With Irregular Motions and Similar Appearances ? Make It Easier by Buffering the Matching Space,” dans *WACV*, 2023.

[43] G. Bertasius et L. Torresani, “Classifying, Segmenting, and Tracking Object Instances in Video with Mask Propagation,” dans *CVPR*, 2020.

[44] P. Tokmakov, J. Li, W. Burgard et A. Gaidon, “Learning to Track with Object Permanence,” dans *ICCV*, 2021.

[45] E. Rublee, V. Rabaud, K. Konolige et G. Bradski, “ORB : An efficient alternative to SIFT or SURF,” dans *ICCV*, 2011.

[46] M. A. Fischler et R. C. Bolles, “Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, n°. 6, p. 381–395, 1981.

[47] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, vol. 82, n°. 1, p. 35–45, mars 1960.

[48] J. Cao, X. Weng, R. Khirodkar, J. Pang et K. Kitani, “Observation-Centric SORT : Rethinking SORT for Robust Multi-Object Tracking,” dans *CVPR*, 2023.

[49] S. Hochreiter et J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, n°. 8, p. 1735–1780, nov. 1997, conference Name : Neural Computation.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser et I. Polosukhin, “Attention is All you Need,” dans *NIPS*, 2017.

[51] F. Giuliari, I. Hasan, M. Cristani et F. Galasso, “Transformer Networks for Trajectory Forecasting,” dans *ICPR*, 2020.

[52] S. Pellegrini, A. Ess, K. Schindler et L. van Gool, “You’ll never walk alone : Modeling social behavior for multi-target tracking,” dans *ICCV*, 2009.

[53] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei et S. Savarese, “Social LSTM : Human Trajectory Prediction in Crowded Spaces,” dans *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, p. 961–971.

[54] A. Vemula, K. Muelling et J. Oh, “Social Attention : Modeling Attention in Human Crowds,” dans *2018 IEEE International Conference on Robotics and Automation (ICRA)*, mai 2018, p. 4601–4607, iSSN : 2577-087X.

[55] B. D. Lucas et T. Kanade, “An iterative image registration technique with an application to stereo vision,” dans *International Joint Conference on Artificial Intelligence (IJCAI)*, 1981.

[56] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers et T. Brox, “FlowNet : Learning Optical Flow With Convolutional Networks,” dans *ICCV*, 2015.

[57] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy et T. Brox, “FlowNet 2.0 : Evolution of Optical Flow Estimation With Deep Networks,” dans *CVPR*, 2017.

[58] Z. Teed et J. Deng, “RAFT : Recurrent All-Pairs Field Transforms for Optical Flow,” dans *ECCV*, 2020.

[59] P. Dendorfer, V. Yugay, A. Ošep et L. Leal-Taixé, “Quo Vadis : Is Trajectory Forecasting the Key Towards Long-Term Multi-Object Tracking ?” oct. 2022, arXiv :2210.07681 [cs]. [En ligne]. Disponible : http://arxiv.org/abs/2210.07681

[60] S. Ding, L. Lin, G. Wang et H. Chao, “Deep feature learning with relative distance comparison for person re-identification,” *Pattern Recognition*, vol. 48, n°. 10, p. 2993–3003, oct. 2015.

[61] A. Hermans, L. Beyer et B. Leibe, “In Defense of the Triplet Loss for Person Re-Identification,” nov. 2017.

[62] W. Chen, X. Chen, J. Zhang et K. Huang, “Beyond Triplet Loss : A Deep Quadruplet Network for Person Re-Identification,” dans *CVPR*, 2017.

[63] C. Song, Y. Huang, W. Ouyang et L. Wang, “Mask-Guided Contrastive Attention Model for Person Re-Identification,” dans *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, p. 1179–1188.

[64] E. Ning, C. Wang, H. Zhang, X. Ning et P. Tiwari, “Occluded person re-identification with deep learning : A survey and perspectives,” *Expert Systems with Applications*, vol. 239, p. 122419, avr. 2024.

[65] J. Seidenschwarz, G. Brasó, V. C. Serrano, I. Elezi et L. Leal-Taixé, “Simple Cues Lead to a Strong Multi-Object Tracker,” déc. 2022.

[66] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, 1955.

[67] Y. Zhang, P. Sun, Y. Jiang, D. Yu, Z. Yuan, P. Luo, W. Liu et X. Wang, “ByteTrack : Multi-Object Tracking by Associating Every Detection Box,” dans *European conference on computer vision*, 2022.

[68] E. Bochinski, V. Eiselein et T. Sikora, “High-Speed tracking-by-detection without using image information,” dans *AVSS*, 2017.

[69] E. Bochinski, T. Senst et T. Sikora, “Extending IOU Based Multi-Object Tracking by Visual Information,” dans *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2018, iSSN : null.

[70] G. Brasó et L. Leal-Taixé, “Learning a Neural Solver for Multiple Object Tracking,” dans *CVPR*, 2020.

[71] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang et P. Luo, “TransTrack : Multiple Object Tracking with Transformer,” mai 2021.

[72] F. Zeng, B. Dong, T. Wang, X. Zhang et Y. Wei, “MOTR : End-to-End Multiple-Object Tracking with TRansformer,” dans *European Conference on Computer Vision*, 2022.

[73] A. Milan, L. Leal-Taixe, I. Reid, S. Roth et K. Schindler, “MOT16 : A Benchmark for Multi-Object Tracking,” mai 2016.

[74] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler et L. Leal-Taixé, “MOT20 : A benchmark for multi object tracking in crowded scenes,” mars 2020.

[75] M. Fabbri, G. Braso, G. Maugeri, O. Cetintas, R. Gasparini, A. Osep, S. Calderara, L. Leal-Taixe et R. Cucchiara, “MOTSynth : How Can Synthetic Data Help Pedestrian Detection and Tracking ?” dans *ICCV*, 2021.

[76] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger et B. Leibe, “MOTS : Multi-Object Tracking and Segmentation,” dans *CVPR*, 2019.

[77] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang et S. Lyu, “UA-DETRAC : A new benchmark and protocol for multi-object detection and tracking,” *Computer Vision and Image Understanding*, vol. 193, p. 102907, avr. 2020.

[78] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang et Q. Tian, “The Unmanned Aerial Vehicle Benchmark : Object Detection and Tracking,” dans *ECCV*, 2018.

[79] E. Ristani, F. Solera, R. Zou, R. Cucchiara et C. Tomasi, “Performance Measures and a Data Set for Multi-target, Multi-camera Tracking,” dans *ECCV Workshops*, G. Hua et H. Jégou, édit., 2016.

[80] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger et R. Shah, “Signature Verification using a "Siamese" Time Delay Neural Network,” dans *Advances in Neural Information Processing Systems*, vol. 6. Morgan-Kaufmann, 1993.

[81] M. Gutmann et A. Hyvärinen, “Noise-contrastive estimation : A new estimation principle for unnormalized statistical models,” dans *AISTATS*, 2010.

[82] C. Doersch, A. Gupta et A. A. Efros, “Unsupervised Visual Representation Learning by Context Prediction,” dans *ICCV*, 2015.

[83] R. Zhang, P. Isola et A. A. Efros, “Colorful Image Colorization,” dans *ECCV*, 2016.

[84] S. Gidaris, P. Singh et N. Komodakis, “Unsupervised Representation Learning by Predicting Image Rotations,” dans *ICLR*, 2018.

[85] T. Chen, S. Kornblith, M. Norouzi et G. Hinton, “A Simple Framework for Contrastive Learning of Visual Representations,” dans *ICML*, 2020.

[86] K. He, H. Fan, Y. Wu, S. Xie et R. Girshick, “Momentum Contrast for Unsupervised Visual Representation Learning,” dans *CVPR*, 2020.

[87] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger et I. Sutskever, “Learning Transferable Visual Models From Natural Language Supervision,” dans *ICML*. PMLR, 2021, p. 8748–8763.

[88] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell et F. Yu, “Quasi-Dense Similarity Learning for Multiple Object Tracking,” dans *CVPR*, 2021.

[89] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao et T.-K. Kim, “Multiple Object Tracking : A Literature Review,” *arXiv :1409.7618*, mai 2017.

[90] S. Gladh, M. Danelljan, F. S. Khan et M. Felsberg, “Deep Motion Features for Visual Tracking,” dans *International Conference on Pattern Recognition (ICPR)*, 2016.

[91] H.-L. Ooi, G.-A. Bilodeau, N. Saunier et D.-A. Beaupré, “Multiple Object Tracking in Urban Traffic Scenes with a Multiclass Object Detector,” dans *International Symposium on Visual Computing (ISVC)*, 2018.

[92] J.-P. Jodoin, G.-A. Bilodeau et N. Saunier, “Tracking All Road Users at Multimodal Urban Traffic Intersections,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, n°. 11, p. 3241–3251, nov. 2016.

[93] Y. Yang et G.-A. Bilodeau, “Multiple Object Tracking with Kernelized Correlation Filters in Urban Mixed Traffic,” dans *Conference on Computer and Robot Vision (CRV)*, mai 2017.

[94] S. Kornblith, J. Shlens et Q. V. Le, “Do Better ImageNet Models Transfer Better ?” dans *CVPR*, 2019.

[95] D. Riahi et G.-A. Bilodeau, “Multiple object tracking based on sparse generative appearance modeling,” dans *IEEE International Conference on Image Processing (ICIP)*, 2015.

[96] D. Zhu, H. Sun et N. Yang, “A real-time and robust approach for short-term multiple objects tracking,” dans *International Conference on Computer Science and Information Processing (CSIP)*, 2012.

[97] L. Sun, G. Liu et Y. Liu, “Multiple pedestrians tracking algorithm by incorporating histogram of oriented gradient detections,” *IET Image Processing*, vol. 7, n°. 7, p. 653–659, oct. 2013.

[98] M. Heimbach, K. Ebadi et S. Wood, “Resolving occlusion ambiguity by combining Kalman tracking with feature tracking for image sequences,” dans *2017 51st Asilomar Conference on Signals, Systems, and Computers*, oct. 2017, p. 144–147.

[99] L. Wang, N. T. Pham, T.-T. Ng, G. Wang, K. L. Chan et K. Leman, “Learning deep features for multiple object tracking by using a multi-task learning strategy,” dans *IEEE International Conference on Image Processing (ICIP)*, 2014.

[100] A. Sadeghian, A. Alahi et S. Savarese, “Tracking the Untrackable : Learning to Track Multiple Cues With Long-Term Dependencies,” dans *ICCV*, 2017.

[101] S. Tang, M. Andriluka, B. Andres et B. Schiele, “Multiple People Tracking by Lifted Multicut and Person Re-Identification,” dans *CVPR*, 2017.

[102] C. Ma, J.-B. Huang, X. Yang et M.-H. Yang, “Hierarchical Convolutional Features for Visual Tracking,” dans *ICCV*, 2015.

[103] M. Danelljan, A. Robinson, F. Shahbaz Khan et M. Felsberg, “Beyond Correlation Filters : Learning Continuous Convolution Operators for Visual Tracking,” dans *ECCV*, 2016.

[104] K. Zhou, Y. Yang, A. Cavallaro et T. Xiang, “Omni-Scale Feature Learning for Person Re-Identification,” dans *ICCV*, 2019.

[105] ——, “Learning Generalisable Omni-Scale Representations for Person Re-Identification,” *arXiv :1910.06827*, oct. 2019.

[106] S. Gil, R. Milanese et T. Pun, “Comparing features for target tracking in traffic scenes,” *Pattern Recognition*, vol. 29, n°. 8, p. 1285–1296, août 1996.

[107] N. Dalal et B. Triggs, “Histograms of oriented gradients for human detection,” dans *CVPR*, 2005.

[108] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng et T. Darrell, “DeCAF : A Deep Convolutional Activation Feature for Generic Visual Recognition,” dans *ICML*, 2014.

[109] K. Simonyan et A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” dans *ICLR*, 2015.

[110] K. He, X. Zhang, S. Ren et J. Sun, “Deep Residual Learning for Image Recognition,” dans *CVPR*, 2016.

[111] G. Huang, Z. Liu, L. van der Maaten et K. Q. Weinberger, “Densely Connected Convolutional Networks,” dans *CVPR*, 2017.

[112] M. Tan et Q. V. Le, “EfficientNet : Rethinking Model Scaling for Convolutional Neural Networks,” dans *ICML*, 2019.

[113] C.-W. Wu, C.-T. Liu, C.-E. Chiang, W.-C. Tu et S.-Y. Chien, "Vehicle Re-Identification With the Space-Time Prior," dans *CVPR - Workshops*, 2018.

[114] S. Jin, H. Su, C. Stauffer et E. Learned-Miller, "End-To-End Face Detection and Cast Grouping in Movies Using Erdos-Renyi Clustering," dans *ICCV*, 2017.

[115] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bulletin of the Calcutta Mathematical Society*, vol. 35, p. 99–109, 1943.

[116] T. Chavdarova, P. Baqué, S. Bouquet, A. Maksai, C. Jose, T. Bagautdinov, L. Lettry, P. Fua, L. Van Gool et F. Fleuret, "WILDTRACK : A Multi-Camera HD Dataset for Dense Unscripted Pedestrian Detection," dans *CVPR*, 2018.

[117] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000, tex.citeulike-article-id : 2236121 tex.posted-at : 2008-01-15 19 :21 :54 tex.priority : 4.

[118] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai et S. Chintala, "PyTorch : An Imperative Style, High-Performance Deep Learning Library," dans *NeurIPS*, 2019.

[119] L. Melas-Kyriazi, "lukemelas/EfficientNet-PyTorch," mars 2020. [En ligne]. Disponible : https://github.com/lukemelas/EfficientNet-PyTorch

[120] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang et Q. Tian, "Scalable Person Re-Identification : A Benchmark," dans *ICCV*, 2015.

[121] K. Zhou et T. Xiang, "Torchreid : A library for deep learning person re-identification in pytorch," *arXiv :1910.10093*, 2019.

[122] X. Liu, W. Liu, H. Ma et H. Fu, "Large-scale vehicle re-identification in urban surveillance videos," dans *International Conference on Multimedia and Expo (ICME)*, 2016, iSSN : 1945-788X.

[123] X. Liu, W. Liu, T. Mei et H. Ma, "A Deep Learning-Based Approach to Progressive Vehicle Re-identification for Urban Surveillance," dans *ECCV*, 2016.

[124] L. Yang, P. Luo, C. Change Loy et X. Tang, "A Large-Scale Car Dataset for Fine-Grained Categorization and Verification," dans *CVPR*, 2015.

[125] J. Sochor, J. Špaňhel et A. Herout, "BoxCars : Improving Fine-Grained Recognition of Vehicles Using 3-D Bounding Boxes in Traffic Surveillance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, n°. 1, p. 97–108, 2018, conference Name : IEEE Transactions on Intelligent Transportation Systems.

[126] M. Naphade, M.-C. Chang, A. Sharma, D. C. Anastasiu, V. Jagarlamudi, P. Chakraborty, T. Huang, S. Wang, M.-Y. Liu, R. Chellappa, J.-N. Hwang et S. Lyu, "The 2018 NVIDIA AI City Challenge," dans *CVPR - Workshops*, 2018.

[127] T. Fu, W. Hu, L. Miranda-Moreno et N. Saunier, "Investigating secondary pedestrian-vehicle interactions at non-signalized intersections using vision-based trajectory data," *Transportation Research Part C : Emerging Technologies*, vol. 105, p. 222–240, août 2019.

[128] E. Beauchamp, N. Saunier et M.-S. Cloutier, "Study of automated shuttle interactions in city traffic using surrogate measures of safety," *Transportation Research Part C : Emerging Technologies*, vol. 135, p. 103465, févr. 2022.

[129] J. Valmadre, A. Bewley, J. Huang, C. Sun, C. Sminchisescu et C. Schmid, "Local Metrics for Multi-Object Tracking," *arXiv :2104.02631*, avr. 2021.

[130] S. Caelles, A. Montes, K.-K. Maninis, Y. Chen, L. Van Gool, F. Perazzi et J. Pont-Tuset, "The 2018 DAVIS Challenge on Video Object Segmentation," *arXiv :1803.00557*, mars 2018.

[131] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers et L. Van Gool, "One-Shot Video Object Segmentation," dans *CVPR*, 2017.

[132] S. Caelles, J. Pont-Tuset, F. Perazzi, A. Montes, K.-K. Maninis et L. Van Gool, "The 2019 DAVIS Challenge on VOS : Unsupervised Multi-Object Segmentation," *arXiv :1905.00737*, mai 2019.

[133] D. Bahdanau, K. Cho et Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," dans *ICLR*, 2015.

[134] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus et R. Socher, "Ask me anything : Dynamic memory networks for natural language processing," dans *ICML*, 2016.

[135] S. Sukhbaatar, A. Szlam, J. Weston et R. Fergus, "End-To-End Memory Networks," dans *NIPS*, 2015.

[136] P. Dendorfer, A. Osep, A. Milan, K. Schindler, D. Cremers, I. Reid, S. Roth et L. Leal-Taixé, "MOTChallenge : A Benchmark for Single-Camera Multiple Target Tracking," *International Journal of Computer Vision (IJCV)*, vol. 129, n°. 4, p. 845–881, avr. 2021.

[137] H. Zhang, Y. Wang, J. Cai, H.-M. Hsu, H. Ji et J.-N. Hwang, "LIFTS : Lidar and monocular image fusion for multi-object tracking and segmentation," dans *CVPR - Workshops*, 2020.

[138] J. Luiten, T. Fischer et B. Leibe, “Track to Reconstruct and Reconstruct to Track,” *IEEE Robotics and Automation Letters*, vol. 5, n°. 2, p. 1803–1810, avr. 2020.

[139] D. Wei, J. Hua, H. Wang, B. Lai, K. Huang, C. Zhou, J. Huang et X. Hua, “RobTrack : A Robust Tracker Baseline towards Real-World Robustness in Multi-Object Tracking and Segmentation,” dans *CVPR RVSU Workshop*, 2021.

[140] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le et B. Zoph, “Simple Copy-Paste Is a Strong Data Augmentation Method for Instance Segmentation,” dans *CVPR*, 2021.

[141] Z. Lai, E. Lu et W. Xie, “MAST : A Memory-Augmented Self-supervised Tracker,” dans *CVPR*, 2020.

[142] S. Garg et V. Goel, “Mask Selection and Propagation for Unsupervised Video Object Segmentation,” dans *WACV*, 2021.

[143] Z. Wang, H. Zhao, Y.-L. Li, S. Wang, P. H. S. Torr et L. Bertinetto, “Do Different Tracking Tasks Require Different Appearance Models ?” dans *NeurIPS*, 2021.

[144] A. Jabri, A. Owens et A. A. Efros, “Space-Time Correspondence as a Contrastive Random Walk,” dans *NeurIPS*, 2020.

[145] B. Yan, Y. Jiang, P. Sun, D. Wang, Z. Yuan, P. Luo et H. Lu, “Towards Grand Unification of Object Tracking,” dans *ECCV*, 2022.

[146] J. Cai, M. Xu, W. Li, Y. Xiong, W. Xia, Z. Tu et S. Soatto, “MeMOT : Multi-Object Tracking with Memory,” dans *CVPR*, 2022.

[147] B. Korbar et A. Zisserman, “End-to-end Tracking with a Multi-query Transformer,” oct. 2022.

[148] J. Luiten, A. Hoffhues, B. Beqa, P. Voigtlaender, I. Sárándi, P. Dendorfer, A. Osep, A. Dave, T. Khurana, T. Fischer, X. Li, Y. Fan, P. Tokmakov, S. Bai, L. Yang, F. Perazzi, N. Xu, A. Bewley, J. Valmadre, S. Caelles, J. Pont-Tuset, X. Wang, A. Geiger, F. Yu, D. Ramanan, L. Leal-Taixé et B. Leibe, “RobMOTS : A Benchmark and Simple Baselines for Robust Multi-Object Tracking and Segmentation,” dans *CVPR RVSU Workshop*, 2021.

[149] J. Luiten, P. Voigtlaender et B. Leibe, “PReMVOS : Proposal-Generation, Refinement and Merging for Video Object Segmentation,” dans *ACCV*, 2018.

[150] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung et L. Van Gool, “The 2017 DAVIS Challenge on Video Object Segmentation,” *arXiv :1704.00675*, mars 2018.

[151] S. Qiao, Y. Zhu, H. Adam, A. Yuille et L.-C. Chen, "ViP-DeepLab : Learning Visual Perception with Depth-aware Video Panoptic Segmentation," dans *CVPR*, 2021.

[152] A. Kim, A. Ošep et L. Leal-Taixé, "EagerMOT : Real-time 3D multi-object tracking and segmentation via sensor fusion," dans *CVPR - Workshops*, 2020.

[153] Z. Xu, W. Zhang, X. Tan, W. Yang, H. Huang, S. Wen, E. Ding et L. Huang, "Segment as Points for Efficient Online Multi-Object Tracking and Segmentation," dans *ECCV*, 2020.

[154] Y.-m. Song et M. Jeon, "Online Multi-Object Tracking and Segmentation with GMPHD Filter and Simple Affinity Fusion," dans *CVPR - Workshops*, 2020.

[155] G. Braso, O. Cetintas et L. Leal-Taixe, "Multi-Object Tracking and Segmentation via Neural Message Passing," juill. 2022.

[156] M. Ahrnbom, M. Nilsson et H. Ardö, "Real-time and online segmentation multi-target tracking with track revival re-identification," dans *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2021.

[157] J. Xing, H. Ai et S. Lao, "Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses," dans *CVPR*, 2009, p. 1200–1207.

[158] E. Ristani et C. Tomasi, "Features for Multi-Target Multi-Camera Tracking and Re-Identification," dans *CVPR*, 2018.

[159] G. Wang, R. Gu, Z. Liu, W. Hu, M. Song et J.-N. Hwang, "Track Without Appearance : Learn Box and Tracklet Embedding With Local and Global Motion Patterns for Vehicle Tracking," dans *ICCV*, 2021.

[160] F. Saleh, S. Aliakbarian, H. Rezatofighi, M. Salzmann et S. Gould, "Probabilistic Tracklet Scoring and Inpainting for Multiple Object Tracking," dans *CVPR*, 2021.

[161] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner et G. Monfardini, "The Graph Neural Network Model," *Transactions on Neural Networks*, janv. 2009.

[162] O. Cetintas, G. Brasó et L. Leal-Taixé, "Unifying Short and Long-Term Tracking with Graph Hierarchies," dans *CVPR*, 2023.

[163] Y. Liu, Q. Yan et A. Alahi, "Social NCE : Contrastive Learning of Socially-aware Motion Representations," dans *ICCV*, 2021.

[164] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos et M. Valko, "Bootstrap Your Own Latent : A New Approach to Self-Supervised Learning," dans *NeurIPS*, 2020.

[165] W. Li, Y. Xiong, S. Yang, M. Xu, Y. Wang et W. Xia, “Semi-TCL : Semi-Supervised Track Contrastive Representation Learning,” juill. 2021.

[166] S. Appalaraju, Y. Zhu, Y. Xie et I. Fehérvári, “Towards Good Practices in Self-supervised Representation Learning,” dans *NeurIPS Workshops*, 2020.

[167] S. Sun, N. Akhtar, H. Song, A. Mian et M. Shah, “Deep Affinity Network for Multiple Object Tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, n°. 1, p. 104–119, janv. 2021, conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence.

[168] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit et N. Houlsby, “An Image is Worth 16x16 Words : Transformers for Image Recognition at Scale,” dans *ICLR*, 2021.

[169] T. Zhu, M. Hiller, M. Ehsanpour, R. Ma, T. Drummond, I. Reid et H. Rezatofighi, “Looking Beyond Two Frames : End-to-End Multi-Object Tracking Using Spatial and Temporal Transformers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, n°. 11, p. 12 783–12 797, nov. 2023, conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence.

[170] P. Tokmakov, A. Jabri, J. Li et A. Gaidon, “Object Permanence Emerges in a Random Walk along Memory,” dans *International Conference on Machine Learning*, 2022.

[171] Z. Liu, X. Wang, C. Wang, W. Liu et X. Bai, “SparseTrack : Multi-Object Tracking by Performing Scene Decomposition based on Pseudo-Depth,” nov. 2023.

[172] C. Xiao, Q. Cao, Y. Zhong, L. Lan, X. Zhang, H. Cai, Z. Luo et D. Tao, “MotionTrack : Learning Motion Predictor for Multiple Object Tracking,” juin 2023.

[173] N. Aharon, R. Orfaig et B.-Z. Bobrovsky, “BoT-SORT : Robust Associations Multi-Pedestrian Tracking,” juill. 2022.

[174] T.-Y. Lin, P. Goyal, R. Girshick, K. He et P. Dollar, “Focal Loss for Dense Object Detection,” dans *ICCV*, 2017, p. 2980–2988.

[175] I. Leichter et E. Krupka, “Monotonicity and Error Type Differentiability in Performance Measures for Target Detection and Tracking in Video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, n°. 10, p. 2553–2560, oct. 2013, conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence.

[176] K. Boyd, K. H. Eng et C. D. Page, “Area under the Precision-Recall Curve : Point Estimates and Confidence Intervals,” dans *Machine Learning and Knowledge Discovery in Databases*, ser. Lecture Notes in Computer Science, H. Blockeel, K. Kersting, S. Nijssen et F. Železný, édit. Berlin, Heidelberg : Springer, 2013, p. 451–466.

ANNEXE A STATISTIQUES SUR LES BASES DE DONNÉES DE MOT ET MOTS LORSQUE LES ANNOTATIONS SONT DISPONIBLES

Pour les bases de données DanceTrack, MOTSChallenge et KITTIMOTS, les partitionnement des données sont ceux fournis par les auteurs des jeux de données. Pour les bases de données MOT17 et MOT20, les partitionnements ont été obtenus selon la stratégie de Zhou et al [19] en divisant chaque séquence équitablement entre données d'apprentissage et données de validation. Pour la base de données KITTIMOT, le partitionnement a été obtenu en reprenant celui de KITTIMOTS, en suivant la stratégie de Luiten et al [11].

Le tableau A.1 comprend des statistiques des bases de données en MOT et MOTS.

Tableau A.1 Statistiques des bases de données

Données	partition	séq.	trames	objets	obs.	objets/séq.	obs./trame
MOT17	train	7	5316	546	112297	78.0	21.1
MOT17	train_half	7	2657	359	58255	51.3	21.9
MOT17	val_half	7	2659	339	54042	48.4	20.3
MOT20	train	4	8931	2215	1134614	553.8	127.0
MOT20	train_half	4	4464	1240	519033	310.0	116.3
MOT20	val_half	4	4467	1418	615581	354.5	137.8
MOTSChallenge	train	4	2862	228	26894	57.0	9.4
MOTSChallenge	train_half	4	1430	129	13548	32.2	9.5
MOTSChallenge	val_half	4	1432	144	13346	36.0	9.3
KITTIMOT	fulltrain	21	8008	746	38770	35.5	4.8
KITTIMOT	train	12	5027	527	27226	43.9	5.4
KITTIMOT	val	9	2981	219	11544	24.3	3.9
KITTIMOTS	fulltrain	21	8008	749	38275	35.7	4.8
KITTIMOTS	train	12	5027	530	26899	44.2	5.4
KITTIMOTS	val	9	2981	219	11376	24.3	3.8
DanceTrack	train	40	41796	419	348930	10.5	8.3
DanceTrack	val	25	25508	273	225148	10.9	8.8