MLVOT - Rapport Finale

Baptiste Bellamy

January 6, 2025

Abstract

Ce projet est le résultat de plusieurs TPs. Nous avons du implémenter des algorithmes de segmentation de personnes. Nous allons pour chaque étapes, décrire en quoi elle consiste, les résultats ainsi que les problèmes rencontrés. Pour chaque étapes, des vidéos sont fournies.

Contents

1 Single Object Tracking avec Kalman Filter (Centroid-Tracker)

L'objectif de ce TP était d'implémenter un système de suivi d'objet simple (SOT - Single Object Tracker) dans un espace 2D. Le projet consistait à intégrer un filtre de Kalman afin de suivre un objet représenté par son centroïde à travers une séquence vidéo.

1.1 Code

Pour mener à bien ce projet, plusieurs contributions clés ont été apportées, à commencer par l'initialisation et l'implémentation du filtre de Kalman. Un fichier dédié, KalmanFilter.py, a été créé pour contenir une classe intitulée KalmanFilter. Cette classe regroupe les fonctions essentielles nécessaires au suivi d'objet. La fonction __init__() a permis d'initialiser les matrices fondamentales (A, B, H, Q, R, P) ainsi que les différents paramètres tels que l'intervalle de temps, les accélérations, et les écarts-types liés au bruit de processus et aux mesures. Ensuite, la fonction predict() a été développée pour estimer l'état actuel de l'objet et prédire sa position future. Enfin, la fonction update() a permis d'ajuster les prédictions en fonction des données mesurées fournies par la détection.

L'étape suivante a consisté à intégrer la détection dans le suivi de l'objet. Pour cela, la fonction detect(), issue du fichier Detector.py, a été utilisée. Cette fonction permet de la détection des centroïdes des objets présents dans chaque images qui servent d'entrée pour le filtre de Kalman.

La création du fichier principal objTracking.py a permis de rassembler l'ensemble des composants pour réaliser le suivi complet. Ce fichier assure la capture et la lecture des vidéos, tout en utilisant le filtre de Kalman pour prédire et corriger les positions des objets détectés. Une attention particulière a été portée à la visualisation des résultats. Ainsi, le centroïde détecté est représenté par un cercle vert, tandis que la position prédite par le filtre est affichée sous forme d'un rectangle bleu. De plus, la position estimée corrigée, obtenue après l'ajustement des mesures, est indiquée par un rectangle rouge. Enfin, la trajectoire de l'objet suivi est tracée sous forme d'une ligne jaune pour donner une vue d'ensemble du chemin parcouru.

1.2 Résultats

Les positions de l'objet ont été suivies avec précision tout au long de la séquence, et la trajectoire a été représentée dans la vidéo. Grâce à l'utilisation du filtre de Kalman, les variations brusques dans les détections ont pu être efficacement atténuées, offrant ainsi un suivi fluide et cohérent.

1.3 Défis et Solutions

Je n'ai pas rencontrée de problème particulier dans cette partie.

2 IOU-based Tracking (Bounding-Box Tracker)

Dans le cadre du TP2, intitulé "IOU-based Tracking (Bounding-Box Tracker)", l'objectif était de concevoir un système simple de suivi multi-objets (MOT) basé sur l'Intersection over Union (IoU). Ce traqueur repose sur une représentation des objets sous forme de boîtes englobantes (bounding boxes). Les données fournies comprenaient des séquences d'images, des fichiers de détection et des annotations de vérité terrain.

Les vidéos ont été exécuté sur YOLOv5l.

2.1 Code

2.1.1 Chargement et ptétraitement des données

Les détections ont été chargées à partir des fichiers CSV fournis, chaque ligne décrivant une boîte englobante avec les informations suivantes :

- Le numéro de la frame.
- Les coordonnées et dimensions de la boîte englobante.
- Le score de confiance de la détection.
- Les coordonnées 3D, ignorées pour ce défi 2D.

Ces données ont été structurées dans des tableaux pour permettre leur exploitation dans les étapes suivantes du processus.

2.1.2 Détection des trajectoires

Afin de détecter les trajectoire, nous utilisons une matrice de similarité. La similarité entre les boîtes englobantes détectées et les objets suivis a été mesurée en calculant l'Intersection over Union (IoU). Une matrice de similarité a été construite, où chaque entrée correspond au score d'IoU entre une trajectoire existante et une nouvelle détection. Ces scores ont permis d'évaluer le degré de correspondance entre les objets détectés et ceux suivis. L'association des détections aux trajectoires existantes a été réalisée en appliquant l'algorithme hongrois, implémenté à l'aide de la fonction linear_sum_assignment de la bibliothèque scipy. Cet algorithme a permis d'optimiser l'appariement en maximisant les scores d'IoU. Les appariements ayant un score inférieur à un seuil prédéfini ont été ignorés, permettant ainsi de réduire les erreurs de correspondance.

Un système de gestion des trajectoires a été implémenté pour suivre les objets tout au long de la séquence vidéo. Les principales étapes incluent la mise à jour des trajectoris existantes, la gestion des trahectoires disparue (si au bout de X frames, la box à disparue, alors la trahectoires est supprimer) mais aussi la créatio de nouvelles trajectoires.

2.1.3 Visualisation des résultats

Pour vérifier la performance du traqueur, une interface de visualisation a été développée. Chaque objet suivi était représenté par une boîte englobante superposée à la vidéo, avec un ID unique affiché au-dessus. Les résultats finaux ont été sauvegardés dans un fichier texte et la video annoté en format mp4.

2.1.4 Dificultés

Le développement de ce traqueur basé sur l'IoU a présenté plusieurs défis techniques. L'un des principaux obstacles a été la mise en œuvre correcte de la matrice de similarité, qui repose sur le calcul

de l'Intersection over Union (IoU) entre chaque boîte englobante détectée et les objets suivis. Initialement, les résultats n'étaient pas concluants, car des erreurs dans le calcul de l'IoU ou la gestion des indices dans la matrice entraînaient des associations incorrectes. Cela affectait la cohérence du suivi, avec des trajectoires erronées ou des objets non appariés de manière adéquate. De plus la gestion des trajectoires existantes ou non, a été un défis, nottament la suppréssion des trajectoires disparues.

3 Kalman-Guided IoU Tracking (Bounding-Box Tracker)

L'objectif était d'améliorer le système de suivi multi-objets développé dans le TP2 en y intégrant un filtre de Kalman. Ce filtre a permis de prédire l'état des objets suivis, représentés par leurs centroïdes, tout en utilisant l'association basée sur l'Intersection over Union (IoU) et l'algorithme hongrois.

3.1 Code

Le filtre de Kalman est intégré au système de suivi en tant que module prédictif. À chaque nouvelle frame, les centroïdes des objets suivis sont d'abord mis à jour grâce aux prédictions du filtre, en fonction de leur état précédent. Ces prédictions sont ensuite utilisées pour construire une matrice de similarité avec les nouvelles détections, basée sur les scores d'IoU. Une fois cette matrice calculée, l'algorithme hongrois associe de manière optimale les nouvelles détections aux trajectoires existantes.

Lorsqu'une détection est appariée à une trajectoire, les prédictions du filtre de Kalman sont corrigées en intégrant la nouvelle position mesurée. En revanche, pour les trajectoires non appariées, le filtre de Kalman continue de prédire leurs positions, ce qui permet de gérer efficacement les pertes temporaires de détection. Les trajectoires non mises à jour pendant un certain nombre de frames consécutives sont ensuite supprimées, tandis que de nouvelles trajectoires sont créées pour les nouvelles détections.

3.2 Résultat

L'intégration du filtre de Kalman au suivi multi-objets basé sur l'IoU permet d'améliorer la robustesse et la précision du suivi. En prédisant les positions futures des objets suivis, le filtre de Kalman lisse les trajectoires et compense les pertes temporaires de détection.

La vidéo finale, "Yolov5l_KelmannFilter_video.mp4", présente le Kalman-Guided IoU Tracking. Les résultats sont très satisfaisants, avec un suivi précis des individus et un nombre minimal d'artefacts.

3.3 Difficultés

L'une des principales difficultés rencontrées a été l'intégration du calcul des centroïdes à l'algorithme basé sur l'IoU. Pour utiliser le filtre de Kalman, il était nécessaire de représenter les boîtes englobantes détectées par leurs centroïdes, ce qui a exigé une conversion précise entre les coordonnées des boîtes et celles des centroïdes. Cette étape a pendant un temps, introduit des erreurs dans les calculs. Une autre difficulté majeure concernait la gestion des dimensions des boîtes englobantes. En effet, le filtre de Kalman prédit les positions des centroïdes, mais les dimensions des boîtes doivent également être ajustées pour refléter les variations réelles des objets suivis.

4 Appearance-Aware IoU-Kalman Object Tracker

Cette partie a pour objectif d'étendre le traqueur IoU-Kalman en intégrant un mécanisme de ré-identification d'objets (ReID). Ce nouveau composant permettait d'utiliser des informations d'apparence, en plus des données géométriques (IoU), pour améliorer la robustesse du suivi d'objets dans des séquences vidéo complexes. L'intégration de la ReID reposait sur l'utilisation de caractéristiques extraites d'objets détectés grâce à un modèle de deep learning préentraîné.

4.1 Code

4.1.1 Chargement et initialisation

Le programme commence par charger les données nécessaires:

- Les inputs précedemment utilisées (det.txt et images)
- Un modèle de ReID pré-entraîné, chargé à l'aide de la bibliothèque onnxruntime.

4.1.2 Prétraitement des données

Les patches des objets détectés, extraits des images selon les boîtes englobantes, sont traités par redimensionnement à une taille standard (64x128), normalisation des valeurs RGB selon les statistiques d'ImageNet, puis convertis en tenseurs adaptés au modèle de ReID.

4.1.3 Calcul des caractéristiques ReID

Chaque patch d'image est passé dans le modèle de ReID pour obtenir un vecteur de caractéristiques unique représentant l'apparence de l'objet. Ce vecteur est ensuite utilisé pour comparer les objets détectés à ceux suivis afin de suivre le déplacement et de retrouver l'objet en cas de perte de l'objet.

4.1.4 Matrice de similarité

Pour chaque frame, une matrice de similarité est construite en combinant deux éléments principaux :

- **L'IoU (Intersection over Union)** : Calculé entre les boîtes englobantes des objets détectés et les trajectoires prédites par le filtre de Kalman.
- **La similarité d'apparence** : Distance cosinus entre les vecteurs de caractéristiques des objets, extraits par le modèle ReID à partir des détections actuelles et précédentes.).

La formule utilisée pour calculer le score de similarité combiné S est donnée par :

$$S = \alpha \cdot \text{IoU} + \beta \cdot \text{similarité d'apparence}$$

où α et β sont des poids qui équilibrent l'importance de l'IoU et de la similarité d'apparence (arbitrairement choisi, ici $\alpha = 0.7$ et $\beta = 0.3$.)

Une fois la matrice de similarité construite, l'algorithme hongrois est appliqué pour associer de manière optimale les détections aux trajectoires existantes.

Enfin, les trajectoires sont alors mises à jour de la même manière que précedemment, cependant, Les caractéristiques ReID (du model) sont elles aussi mises à jour.

4.2 Résultat et dfficultées

La gestion des pertes temporaires est assurée par le filtre de Kalman, qui permet de suivre les objets même en cas d'absence temporaire de détection, tandis que la ré-identification (ReID) garantit la cohérence des identifiants malgré les occlusions ou les variations d'apparence..

La vidéo "output/Yolov5l_Reid_video.mp4" illustre les résultats obtenus, où l'on peut observer que la ré-identification fonctionne correctement lors de courtes pertes des trackers. Cependant, du bruit apparaît, et le problème à l'origine n'a pas pu être identifié. Les principales difficultés rencontrées incluent la mise à jour et la sauvegarde des caractéristiques ReID, qui ont engendré de nombreux bugs, ainsi que des artefacts et autres anomalies non résolus.

5 Conclusion

À travers ce TP, j'ai pu implémenter et comprendre différents algorithmes de segmentation. Parmi eux, le Kalman-Guided IoU Tracking s'est avéré être la meilleure implémentation, comme le montrent les résultats visibles dans la vidéo associée. Bien que l'implémentation de la ré-identification (ReID) ait été réalisée, les résultats obtenus ne correspondent pas aux attentes initiales, laissant place à des améliorations futures.