POLYTECHNIQUE MONTRÉAL

UNIVERSITÉ D'INGÉNIERIE

INF6804 – Vision par ordinateur

Hiver 2020

Rapport TP No. 3 :

Détection et suivi d'un objet d'intérêt

2035719 – Matthias RAMOS

2035967 – Yoann Heitz

Dans le cadre de ce troisième travail pratique nous proposons et évaluons une méthode de suivi d'objet d'intérêt dans une séquence vidéo. Cette méthode est de type modèle discriminatif basé sur le tracking CSRT (Channel and Spatial Reliability Tracker). Nous utilisons une implémentation inclue dans OpenCV et le code fourni avec le travail pratique.

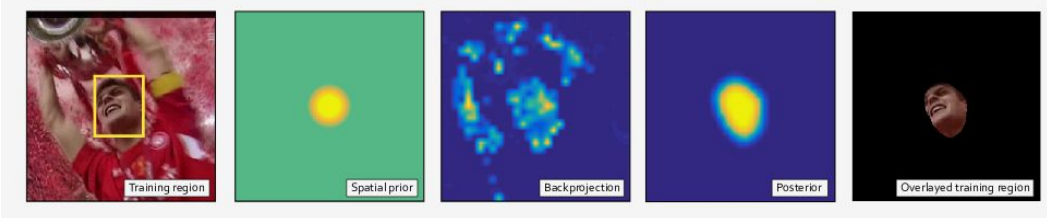
I) Présentation de la méthode :

1) Description générale :

Afin de suivre l'objet, ce dernier est tout d'abord renseigné au début de la séquence vidéo. Il n'y a donc pas de détection initiale à effectuer. Nous utilisons ensuite notre tracker pour suivre ce même objet. Ce modèle est un modèle par région. Plus précisément il est basé sur une méthode par filtre de corrélation :

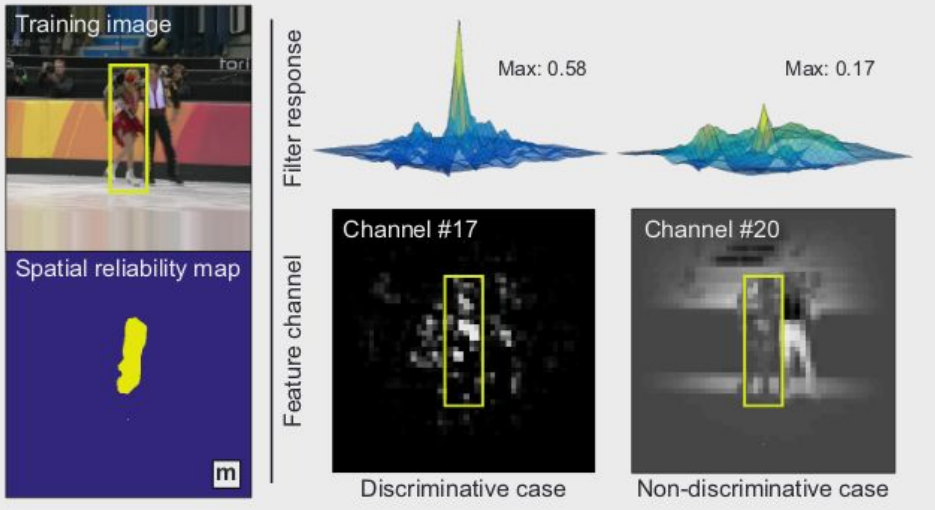
- un filtre optimal est trouvé (lors d'une phase d'entraînement). Ce filtre est ensuite passé sur une région autour de l'endroit où l'objet à suivre a été détecté la dernière fois. L'objectif de ce filtre est de donner une valeur positive à tous les pixels appartenant à l'objet à suivre et une valeur négative à tous les autres (qui font partie de l'arrière plan). On a donc une réponse maximale lorsque le filtre est centré sur l'objet, et c'est alors cette réponse maximale que l'on recherche pour localiser l'objet dans la nouvelle image. Par ailleurs, chaque pixel a été encodé au préalable sur plusieurs canaux. Ces canaux contiennent alors des informations représentant la région et des caractéristiques de chaque pixel et ce sont ces informations qui seront filtrées. De telles caractéristiques peuvent être la description en histogramme de gradients, les différentes intensités de couleur du pixel ou son intensité en niveaux de gris. Dans le cas de CSRT ces canaux sont les canaux associés aux features de HOG, aux noms de couleur et aux niveaux de gris. Cette étape est l'étape de base lors d'un suivi grâce à un filtre de corrélation. Le tracker CSRT ajoute 2 traitements supplémentaires pour améliorer la détection et le suivi de l'objet.

- une "carte spatiale de fiabilité" (spatial reliability map) est construite et un entraînement est effectué dessus lorsque l'objet a été détecté. Cette étape permet, une fois l'objet détecté (et une région rectangulaire l'englobant produite), de déterminer quels pixels de la région calculée font partie de l'arrière plan et lesquels font partie de l'objet. Cela permet alors de déterminer plus en détail la forme de l'objet et donc d'avoir plus d'informations sur celui-ci afin d'améliorer le suivi. En effet, le filtre de corrélation est à chaque image entraîné sur les pixels appartenants à l'objet seulement et non plus sur tous les pixels contenus dans une région rectangulaire centrée sur l'objet et pouvant contenir des pixels d'arrière plan. L'entraînement du filtre sur les pixels de l'arrière plan est donc minimisé au maximum grâce à cette étape.



Reconnaissance des pixels appartenant à l'objet pour la construction de la carte de fiabilité spatiale

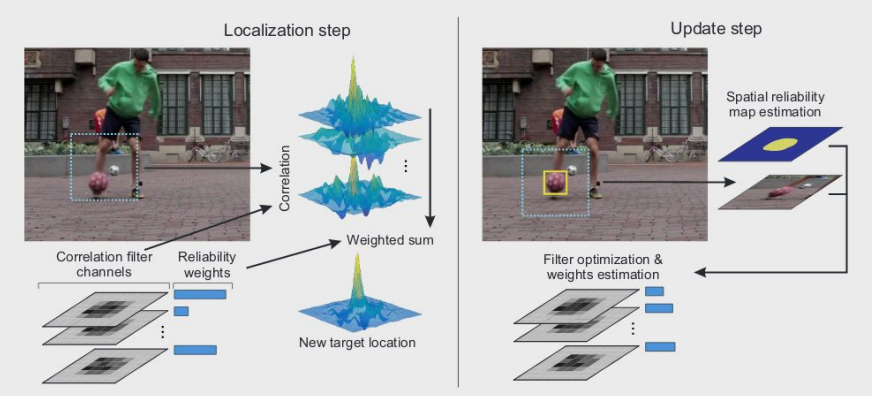
- enfin une “fiabilité des canaux” est estimée (channel reliability estimation). Cela permet d’estimer à quel point chaque canal vote pour la localisation de l’objet. Cette estimation permet alors de donner moins d’importance aux canaux avec moins de fiabilité (car pouvant être bruités et donc produire une bonne réponse pour des pixels d’arrière plan).



Si des canaux sont discriminés les résultats de localisation sont moins bruités que dans le cas où tous les canaux ont le même poids

Le filtre est à chaque image entraîné sur la dernière localisation de l’objet en prenant en compte sa forme déterminée à l’aide de la carte de fiabilité spatiale et à l’aide de l’estimation de fiabilité des canaux (l’impact de certains canaux est réduit, l’aspect discriminatif de cette méthode est donc renforcé). La taille de la bounding box localisant l’objet peut alors aussi changer en fonction de l’estimation des pixels appartenants à l’objet.

CSRT est basé sur un tracker nommé DCF (Discriminative Correlation Filter) qui n’opère que sur l’étape de corrélation avec le filtre. Ce dernier finit donc souvent par entraîner le filtre de corrélation sur des pixels d’arrière plan et ne peut pas déterminer une forme/taille précise de l’objet pour améliorer l’entraînement.



Fonctionnement global du tracker avec étape de localisation et étape d’apprentissage (en prenant en compte les différents poids des canaux et les pixels appartenant probablement à l’objet su

2) Avantages et inconvénients :

Le tracker CSRT a ses propres forces et faiblesses qu'il faut mettre en comparaison avec d'autres trackers existants donc le tracker DCF.

Tout d'abord CSRT effectue deux étapes supplémentaires lors de son entraînement sur l'objet détecté par rapport à d'autres trackers n'effectuant que l'étape de corrélation. On en déduit deux conséquences sur les performances de CSRT par rapport ces trackers tels que DCF et KCF (Kernelized Correlation Filter). Tout d'abord effectuer ces étapes consomme du temps de calcul. CSRT est donc plus lent à effectuer le suivi que ces trackers (cependant CSRT est quand même apte à faire de la détection en temps réel, même sur un CPU). La contrepartie est que CSRT est plus performant et robuste que KCF et DCF.

CSRT n'est pas basé sur un modèle par mouvement. Il sera donc robuste face à des mouvements imprédictibles. Par contre si l'objectif est de faire une estimation et étude de mouvement, CSRT ne sera pas capable de produire des résultats directement, il faudra extrapoler les caractéristiques du mouvement de l'objet à partir des caractéristiques de bounding boxes prédites.

CSRT n'a besoin que d'une région initiale pour effectuer son entraînement. Une seule bounding box (et donc une seule image initiale prétraitée) est nécessaire pour amorcer le processus de suivi. Le reste de l'entraînement se fait ensuite d'une image à l'autre à partir de la détection précédente.

Lors du suivi d'un objet, selon le tracker utilisé, ce dernier est capable de déterminer sur une image en particulier si il a été capable de localiser l'objet ou si même sa meilleure prédiction est fausse et donc que l'objet n'a pas pu être localisé. CSRT a une bonne capacité d'évaluation de succès de prédiction. Dans le cas où l'objet n'a pas été localisé avec succès CSRT ne mettra donc pas à jour son filtre de corrélation (d'ailleurs CSRT est capable d'annoncer si une perte de localisation sera définitive ou non). Ce comportement entraîne alors plusieurs conséquences :

- si un objet n'a pas pu être localisé sur quelques images et qu'il ne s'est pas trop déplacé, CSRT sera en mesure de pouvoir le relocaliser par la suite si ce dernier n'a pas changé d'apparence durant le temps où il n'a pas pu être localisé.
- si l'objet ne peut plus être localisé à cause d'une occlusion et qu'il ne se déplace et ne change pas trop il pourra donc être relocalisé. Cependant si il change de forme/couleurs pendant l'occlusion il sera perdu de vue du tracker et ne pourra plus être localisé par la suite.
- si le flot d'image est bruité et/ou si des images sont perdues de temps en temps, la détection pourra ne pas être compromise

Le fait que CSRT met à jour son filtre de corrélation en fonction des pixels appartenant à l'objet entraîne que des rotations, changements de formes/tailles et couleurs graduels de l'objet n'impacteront pas la précision du tracker. Cependant si ces changements sont brusques alors le tracker aura du mal à s'adapter.

Sur des occlusions partielles, CSRT commencera à mettre à jour son filtre de corrélation sur des pixels hors objets. Le modèle finira donc par diverger et perdre en précision sur le temps.

Sur de longues occlusions totales ou pertes de localisation, de manière générale l'objet se déplacera beaucoup ou changera et CSRT sera moins apte à se remettre de cette perte de localisation.

II) Description des expériences

1) Précision

Nous évaluerons d'abord la précision du tracker vis à vis du groundtruth. Après avoir généré les bounding boxes sur chaque image, une comparaison sera faites avec celles du groundtruth grâce à la métrique IoU (Intersection over Union). Pour deux ensembles S1 et S2, cette métrique mesure le rapport de la taille de l'intersection de S1 et S2 sur la taille de la réunion de S1 et S2. Dans notre cas S1 et S2 seront l'ensemble des pixels contenus dans les bounding boxes issues du groundtruth et de la prédiction du tracker pour chaque image étudiée.

En fonction d'un certain "cutoff" (30%, 50% ou 70%) chaque encadrement sera jugé correct ou non. Ce "cutoff" correspond à un seuil pour la métrique IoU à partir duquel une prédiction est jugée valide. En dessous de ce seuil d'IoU, la prédiction est considérée mauvaise et rejetée. Ainsi le nombre total d'encadrements corrects donnera la précision du modèle sur la séquence. Cette précision (accuracy) correspond à la proportion de prédictions correctes (rapport du nombre de prédictions correctes et du nombre total d'images de la séquence).

Ensuite la moyenne des IoU des encadrements corrects donnera la "robustesse" de ce modèle, donnant ainsi une idée encore plus précise de l'efficacité de notre tracker. Il est à noter que l'on s'attend donc à chaque fois à une robustesse plus élevée que le seuil de validité. Il faudra donc comparer cette robustesse relativement au seuil de validité d'une prédiction (cutoff).

D'après les caractéristiques de CSRT décrites plus haut, on s'attend à de très bon résultats au niveau de la précision et robustesse de cette méthode, notamment une très bonne résilience pour les séquences avec mouvements brusques. Un redimensionnement important des bounding boxes ne devrait pas représenter un problème non plus.

Par ailleurs nous rapportons aussi la proportion d'images pour lesquelles le tracker estime qu'il n'a pas réussi à localiser l'objet.

2) Vitesse de traitement

La vitesse de traitement, en frame par seconde (fps), sera aussi évaluée pour déterminer les performances "temps réel" de cette méthode. Très simplement, le temps de calcul total divisé par le nombre de frame de la séquence nous donnera cette valeur.

Le temps de calcul pour cette méthode risque d'être assez long par rapport à d'autres techniques de suivi, qu'elles soient génératives ou discriminatives, puisque qu'un entraînement est nécessaire à chaque étape. Cependant on s'attend quand même à avoir un temps de traitement raisonnable même sur un CPU, qui permettrait quasiment de faire un traitement en temps réel (bien que d'autres méthodes plus rapides telles que KCF assurent avec encore une marge de temps un traitement en temps réel).

3) Cas d'utilisation

Pour tester cette méthode, nous l'utiliserons sur des exemples présentant des conditions différentes et spécifiques pour faire apparaître les forces et faiblesses citées précédemment.

Le dataset utilisé sera le dataset VOT2013 qui contient 16 séquences vidéos différentes. Chacune des vidéos est accompagnée d'un fichier texte groundtruth. Ce fichier indique pour chaque image les caractéristiques de la bounding box contenant l'objet à suivre. Une ligne est utilisée par image et ces caractéristiques sont au nombre de 4 et correspondent aux coordonnées du coin supérieur gauche de la bounding box et ses dimensions horizontales et verticales. C'est donc ce fichier qui sera utilisé pour évaluer les performances de notre modèle.

Ce dataset comprend des vidéos dans lesquelles les objets sont filmés dans des conditions variées qui permettent donc d'évaluer le modèle dans beaucoup de situations non optimales pour le suivi.

Pour tester les performances du modèle nous avons décidé de sélectionner 4 vidéos filmées dans des conditions très différentes les unes des autres.

La première vidéo sera une vidéo filmée dans des conditions optimales. L'objet aura un mouvement fluide et ne changera pas de forme. Cette vidéo servira de cas de base pour évaluer les performances pouvant être attendues au mieu du modèle.

Les 3 vidéos suivantes comportent des caractéristiques pouvant mettre à mal un suivi d'objet.

Nous aurons une vidéo où l'objet aura des mouvements brusques et subira des changements de formes, des brusque translations et rotations.

La seconde vidéo sera une vidéo où l'objet subira plusieurs occlusions.

Enfin la dernière vidéo sera une vidéo où l'arrière plan change énormément. On pourra ainsi vérifier que le modèle n'apprend pas le filtre de corrélation sur des pixels appartenant à l'arrière-plan. Dans le cas contraire, sur une telle vidéo, les performances seraient fortement amoindries.

Les 4 vidéos du dataset retenues sur ces critères sont les suivantes:

- cas "baseline" servant d'indicateur de performance de base : la séquence "cup" est retenue. On y voit une main tenir une tasse bleue. Cette tasse a une couleur qui se distingue très facilement de l'arrière-plan qui est blanc et noir. L'objet n'a aucun mouvement brusque, il a forme plutôt carrée à l'image et ne change jamais de silhouette ni ne subit de rotation, et sans aucune occlusion. Il n'y a donc aucun obstacle à sa détection. La vidéo comporte 303 images.
- cas avec des mouvements brusques et changement de forme : la séquence "iceskater" est retenue. C'est une séquence de patinage artistique de haut niveau assez nette, avec beaucoup de mouvements rapides et une silhouette en constante déformation. Il n'y a qu'une seule patineuse et donc un seul objet du même type à suivre sur l'image. Cette séquence permettra de tester la robustesse du modèle aux mouvements brusques et changements rapides de forme (sans occlusion). L'arrière-plan de la vidéo est pour la majeure partie de la séquence le sol de la patinoire en glace qui est blanc. Il n'y a donc aucun obstacle extérieur à l'objet à suivre qui puisse impacter les performances du suivi. Cette vidéo comporte 500 images.
- cas avec occlusion de l'objet : c'est la vidéo "woman" qui est sélectionnée. Elle montre une femme qui se déplace dans la rue. Au cours de la vidéo, la personne passe en partie derrière des voitures et le bas de son corps est donc recouvert par un élément considéré comme d'arrière-plan. Une autre vidéo aurait pu être choisie, la vidéo "face" dans laquelle une personne fait passer un objet devant son visage qui est l'objet
  d'intérêt. Cependant son visage est statique et une modification que nous avons apporté au tracker (et que nous expliciterons dans la partie suivante) dans le cas où l'objet n'est pas détecté pourrait introduire du biais dans les performances relevées. Nous avons donc choisie la séquence "woman" car l'objet d'intérêt subit plusieurs occlusions différentes (de plus les voitures qui recouvrent la personne changent de

couleurs) et en plus la région d'intérêt est statique. Cette séquence comporte 597 images.

- cas avec changement permanent d'arrière-plan : c'est la vidéo "diving" qui est sélectionnée. Elle représente un athlète dans une piscine qui s'apprête à faire un plongeon depuis une planche. Lors de sa préparation il fait des mouvements de grandes amplitudes et se déplace donc beaucoup par rapport à l'arrière-plan (constitué des gradins et des installations de la piscine) qui change donc nettement dans la région encadrée par la bounding box. Avec cette séquence nous pourrons donc voir si le modèle tend à apprendre sur des pixels appartenant à l'arrière plan, dans quel cas les performances seront amoindries. Cette séquence comporte 231 images.

III) Description de l'implémentation

1) Implémentation :

Pour notre implémentation nous nous sommes principalement aidé du code fourni avec l'énoncé de ce travail que nous avons complété avec une librairie de OpenCV pour inclure le tracker. Notre implémentation du tracker est largement inspirée de l'article d'Adrian Rosebrock sur le site pyimagesearch.com, où après tests et consultations d'avis d'utilisateurs avérés, le choix de CSRT comme modèle discriminatif s'est avéré le plus efficace pour l'ensemble des séquences à traiter.

Nous avons cependant apporté certaines modifications à ce tracker. Lors de l'implémentation à l'aide du site précédemment cité et de premiers tests, nous avons remarqué que le tracker indique à chaque image si il a été capable de relocaliser l'objet. Dans ce cas toutes les caractéristiques de la bounding box produite sont fixées à 0. Afin d'essayer d'améliorer les performances nous avons donc fait l'hypothèse que de manière générale, d'une image à la suivante, l'objet change très peu de forme et se déplace de manière continue et uniforme. Ainsi si le tracker est incapable de localiser l'objet, nous supposons que la bounding box qui devrait être produite a les mêmes dimensions que la bounding box de l'image précédente et s'est déplacée de manière uniforme par rapport aux 2 images précédentes (même mouvement entre les 2 images précédentes et l'image actuelle et la précédente). Nous produisons ainsi une bounding box possédant ces caractéristiques. C'est pour cela que nous n'avons pas choisis la séquence "face", puisque le visage à suivre ne bouge pas, les bounding boxes produites par notre implémentation pourraient être correctes même si le tracker est dans l'incapacité de suivre le visage. Aussi, comme mentionné plus haut, nous rapportons la proportion d'images sur lesquelles le tracker a été capable de retrouver l'objet afin de comparer cette proportion à la précision globale et donc savoir si nos hypothèses sont valides ou non.

2) Paramètres principaux :

L'implémentation du tracker CSRT d'OpenCV possède divers paramètres. Ces paramètres peuvent être récupérés dans un fichier .yaml et modifié grâce à la lecture d'un tel fichier. On retrouve dans ces paramètres des taux d'apprentissage associés à chaque étape d'apprentissage du filtre de corrélation d'une image à l'autre (sélection des pixels appartenants à l'image, changement de poids des canaux, apprentissage des coefficients du filtre). La valeur de padding du filtre peut être changée, ainsi que le nombre de canaux utilisés pour la localisation.

Nous avons fait varier quelques paramètres (surtout les taux d'apprentissage) à la main mais avons très vite remarqué que les paramètres par défaut produisaient de très bon résultats sur la majorité des séquences choisies (comme nous allons le voir dans la prochaine partie). Nous avons donc choisit de ne modifier aucun paramètres, à part ceux renseignant que le tracker est utilisé sur des images en couleurs et non en niveaux de gris.

IV) Résultats

1) Présentation

Pour chaque séquences nous évaluons donc la précision, la robustesse et mesurons aussi le taux de succès de suivi. Ces données sont mesurées pour les différents IoU cutoff. Nous reportons aussi la vitesse de traitement moyenne (fps) de notre tracker. Nous obtenons alors les résultats suivants (deux images sont rapportées pour montrer les résultats, une image de base et une image avec en vert la bounding box issue du groundtruth et en rouge la bounding box produite par le tracker) :

"cup" :

IoU cuttoff (%)	accuracy (%)	robustesse (%)	pourcentage de réussite (%)
30	100	85	100
50	100	85	100
70	100	85	100

fps : 22,8

“iceskater” :

IoU cuttoff (%)	accuracy (%)	robustesse (%)	pourcentage de réussite (%)
30	100	61	100
50	86	64	100
70	21	77	100

fps : 18,4

“woman” :

IoU cuttoff (%)	accuracy (%)	robustesse (%)	pourcentage de réussite (%)
30	93	63	100
50	84	65	100
70	23	77	100

fps : 22,7

“diving” :

IoU cuttoff (%)	accuracy (%)	robustesse (%)	pourcentage de réussite (%)
30	33	65	88
50	25	72	88
70	14	85	88

fps : 29,1

2) Analyse

Sans grande surprise les résultats sur la vidéo “cup”, une situation sans difficulté particulière, permettent de confirmer l’efficacité générale du tracker CSRT.

Sur les autres exemples plus laborieux, la précision chute radicalement à partir d’un cutoff de 70%. Ce qui signifie que la majorité des encadrement on un IoU avec le groundtruth inférieur à cette valeur. Plus précisément, la robustesse dans les autres cas nous montre que la précision moyenne du cadrage de ce modèle se trouve autour de 60-65% (IoU) (d’où la chute pour le cutoff de 70%), avec une réussite d’encadrement toujours maximale, excepté pour la cas de “diving”.

Ces très bonnes performances pour “iceskater” et “woman” nous permettent donc d’avancer une certaine résilience de CSRT à la déformation, aux mouvements rapides, ainsi qu’à l’occlusion partielle et donc une fiabilité de ce modèle dans un grand nombre de situations.

Diving est objectivement un exemple très difficile pour grand nombre des modèle génératif ou discriminatif. L’objet fait des mouvement extrêmement brusque, subissant une déformation et rotation très ample, et difficile à isoler de l’arrière plan (on remarque que le tracker apprend sur des pixels d’arrière-plan et fini par ne plus être capable de localiser le plongeur). Ce que l’on peut observer dans les résultat nous montre juste que cette méthode a encore les mêmes failles que majorité des modèles déjà conçu, et n’apporte pas de solution à ce type de suivi très complexe à réaliser.

La vitesse de traitement reste autour de 20 fps pour des séquences à 30 fps avec chacune un petit nombre de frame (300 à 600). Cette vitesse pourrait être accélérée en modifiant certains paramètres liés à l’apprentissage (nombre d’itérations sur les méthodes d’apprentissage). On serait alors à même d’exécuter le suivi en temps réel, cependant peut être avec une perte de précision. L’usage d’un GPU plutôt qu’un CPU pourrait aussi accélérer le temps de traitement. Ceci paraît tout à

fait satisfaisant puisque CSRT permet (avec seulement un CPU) de s'approcher très sensiblement d'un traitement temps réel avec un modèle discriminatif très fiable.

Sur les images rapportées on remarque cependant que le changement de forme des bounding boxes peut nécessiter un peu de temps (plusieurs itérations sur plusieurs images successives). Cela ne pose pas de soucis pour des changements de forme graduels mais est plus problématique pour des changements brusques (dans la séquence 'diving' les changements sont plus rapides que pour 'iceskater' et le tracker ne peut plus se remettre de ses erreurs une fois qu'il a appris sur des pixels d'arrière plan alors que pour 'iceskater' les changements sont un peu plus lent, permettant au tracker d'être capable d'effectuer le suivi malgré une perte de précision). Ce phénomène peut cependant être amoindri en modifiant les paramètres d'apprentissage, mais la présence de pixels d'arrière-plan pourrait alors poser plus de problèmes (un changement de dimensions de bounding boxes plus rapide implique un apprentissage plus fort sur les dernières images et donc un apprentissage plus fort sur des pixels d'arrière-plan qui pourrait être considérés comme appartenant à l'objet).

3) Conclusions

CSRT est donc un tracker robuste et précis dans beaucoup de situations incluant des obstacles de base dans le domaine du suivi d'objets (occlusion, changement de forme). Cependant ce tracker possède certaines faiblesses qui peuvent être largement mises en avant lorsque plusieurs obstacles pour la détection sont rencontrés en même temps. Le paramétrage du tracker en fonction de la situation est aussi très important pour obtenir une précision maximale. Cependant un paramétrage optimal universel n'est alors pas possible et il faudrait effectuer une batterie de tests sur des datasets mélants des difficultés variées afin de déterminer quels paramètres sont optimaux dans quelles situations.

Aussi plusieurs situations n'ont pas été testées dans ce travail (nous nous sommes limités à l'étude de 3 situations non basiques). Il pourrait être intéressant de voir comment se comporte le tracker dans une situation où plusieurs objets ayant la même forme/couleur que l'objet à suivre sont présents, ou voir si ce tracker est capable de faire du suivi multiple de manière efficace, ou encore dans des conditions très mauvaises avec un bruit important ou une image floutée.

Enfin ce tracker peut être utilisé pour du suivi en temps réel sur des séquence avec un taux de rafraîchissement moyen mais sans fournir une grande marge de manoeuvre. Sur des vidéos avec un très grand taux de rafraîchissement il faudra cependant sûrement utiliser un autre tracker.

Bibliographie :

VOT2013 Challenge Dataset

Discriminative Correlation Filter Tracker with Channel and Spatial Reliability *Alan Lukeˇziˇc1, Tomᵓaˇs Vojᵓıˇr2, LukaˇCehovin Zajc1, Jiˇrᵓı Matas2and Matej Kristan11Faculty of Computer and Information Science, University of Ljubljana, Slovenia2Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic*

Comparison of Tracking Techniques on 360-Degree Videos *Tzu-Wei Mi and Mau-Tsuen Yang*

https://www.learnopencv.com/object-tracking-using-opencv-cpp-python/

https://www.pyimagesearch.com/2018/07/30/opencv-object-tracking/