

Sanchez_Sielskyi_Rapport_Tracking

January 16, 2021

1 ENSTA-ParisTech 3e année - Cours ROB313

2 TP4: Analyse vidéo et Tracking

Auteurs: Kevin Alessandro Sanchez Diaz et Yevhenii Sielskyi

2.1 Mean Shift

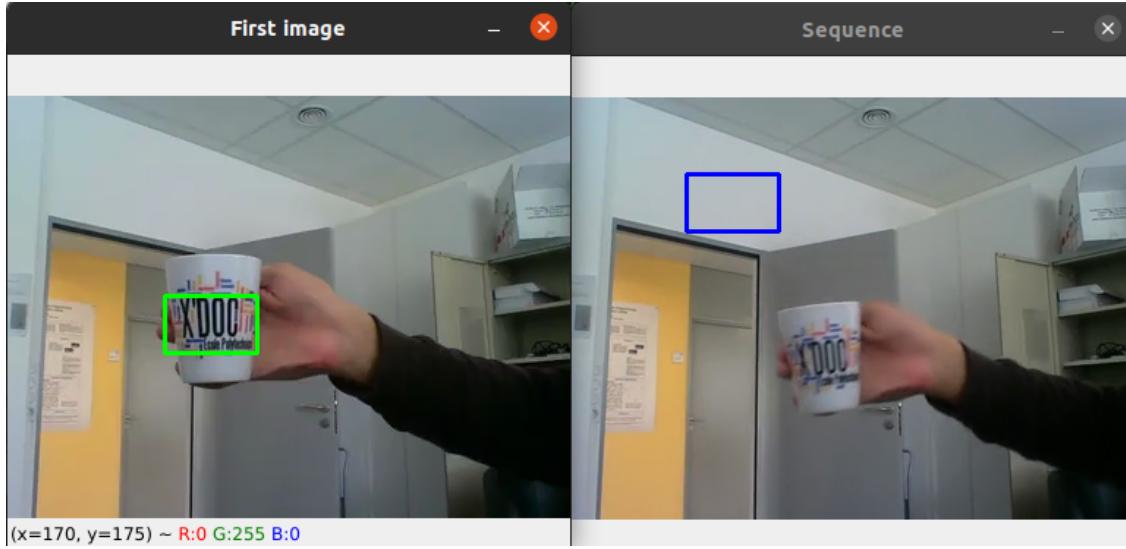
Q1 Expérimenter le suivi réalisé par le code de base `Tracking_MeanShift.py` fourni qui utilise l'algorithme de *Mean Shift*, avec la densité marginale f_H sur la composante H de teinte. Rappeler le principe de l'algorithme *Mean Shift*, et illustrer par vos expériences ses avantages et ses limitations.

[]: !python Tracking_MeanShift.py

Nous avons testé l'implémentation donnée de l'algorithme de *Mean Shift* sur 5 vidéos de tests différents dont les résultats sont suivants:

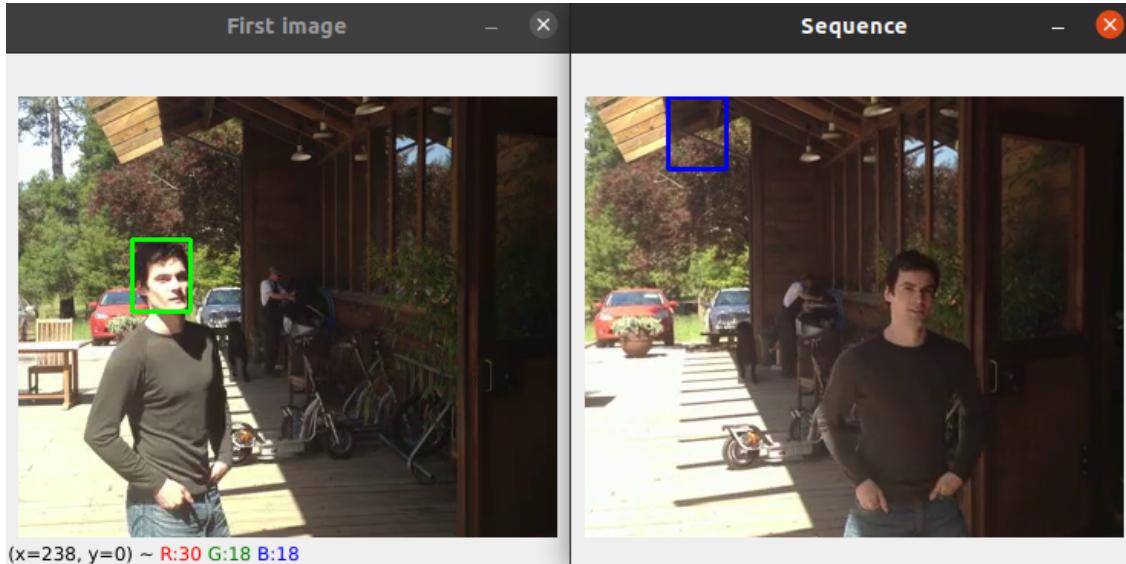
Mean Shift en tant qu'algorithme simple de l'estimation de la mode d'une distribution, souffre de la similarité des couleurs entre l'objet suivi et le fond, les autres objets perturbant les cadres analysés. Sur la **figure 1.1** nous voyons le suivi d'une tasse dont la couleur est grise claire, qui ressemble à celle du mur en arrière-plan. Dès le début du suivi, le rectangle bleu se place sur la région de mur mentionnée, si la taille de *RDI* (*Région D'Intérêt*) choisie n'est pas suffisamment grande.

Figure 1.1 — suivi de la tasse avec *Mean Shift*



Ceci est dû au fait que l'indice de Bhattacharyya est utilisé en tant que mesure de similarité entre la distribution locale actuelle et celle de référence: les petits changements de la teinte (ce qui correspond au canal *H* des images HSV, utilisé pour *Backproject*) d'objet peuvent aboutir à sa perte par RDI. *Figure 1.2* représente les résultats du suivi du visage d'un jeune homme, qui est “disparu” en entrant dans l'ombre.

Figure 1.2 — suivi d'un visage dans les zones à l'éclairage contrasté



Prenons maintenant l'exemple de suivi d'un objet qui se distingue par ses couleurs dans son environnement. Le ballon rouge ci-dessous est unique dans son genre sur le tapis et avec les meubles de couleurs neutres autour. Mais dans ce cas-là, nous faisons face à un autre inconvénient de *Mean Shift* — les changements rapides de la position d'objet relative à RDI: *figure 1.3* représente le cadre de perte du ballon du focus lors de son déplacement et du mouvement brusque de la caméra en même temps. Après avoir perdu le ballon et s'en être éloigné considérablement, RDI reste fixé sur le tapis. L'objet de suivi est recapturé lors de son passage à côté de rectangle bleu (voir *figure 1.4*).

Figure 1.3 — suivi du ballon rouge avec *Mean Shift*. Ballon perdu

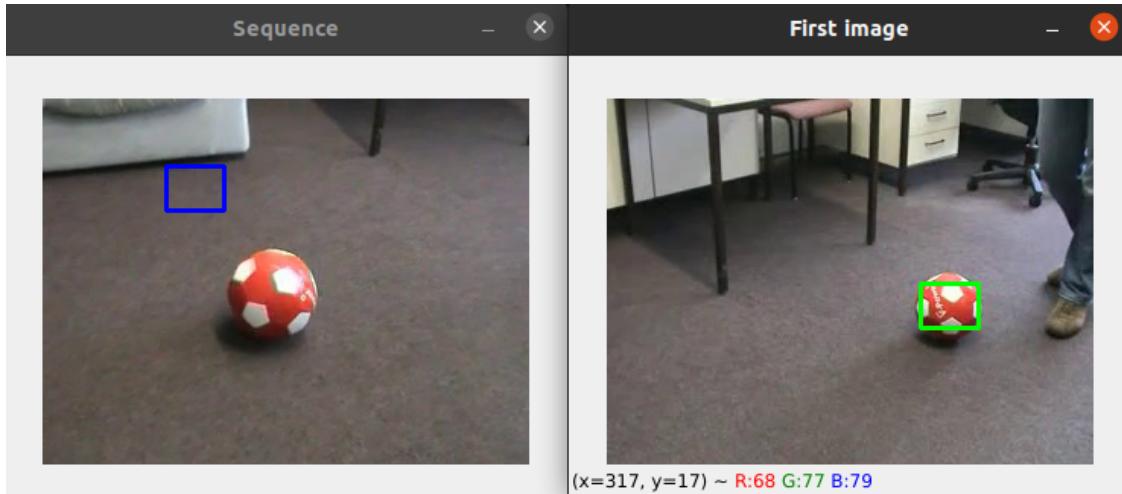
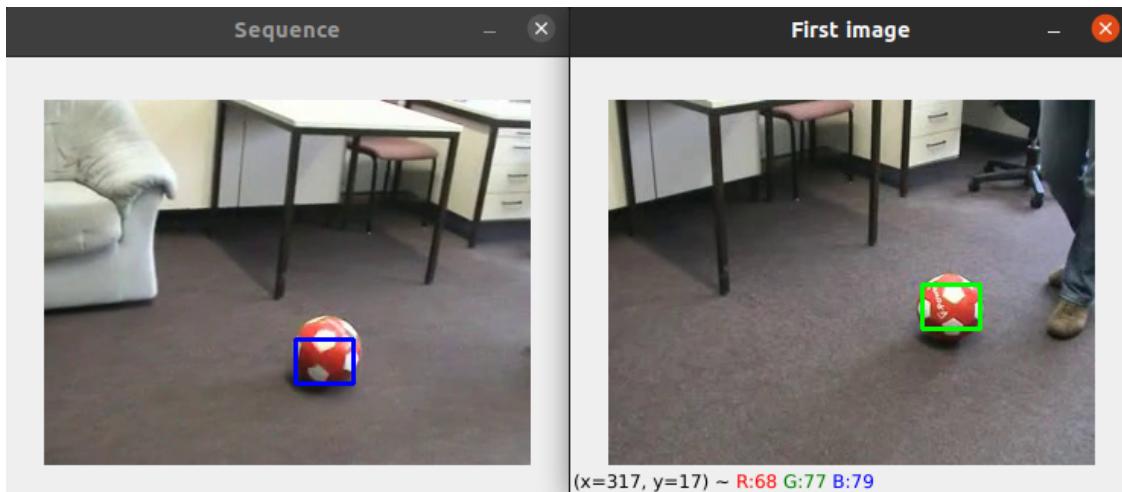
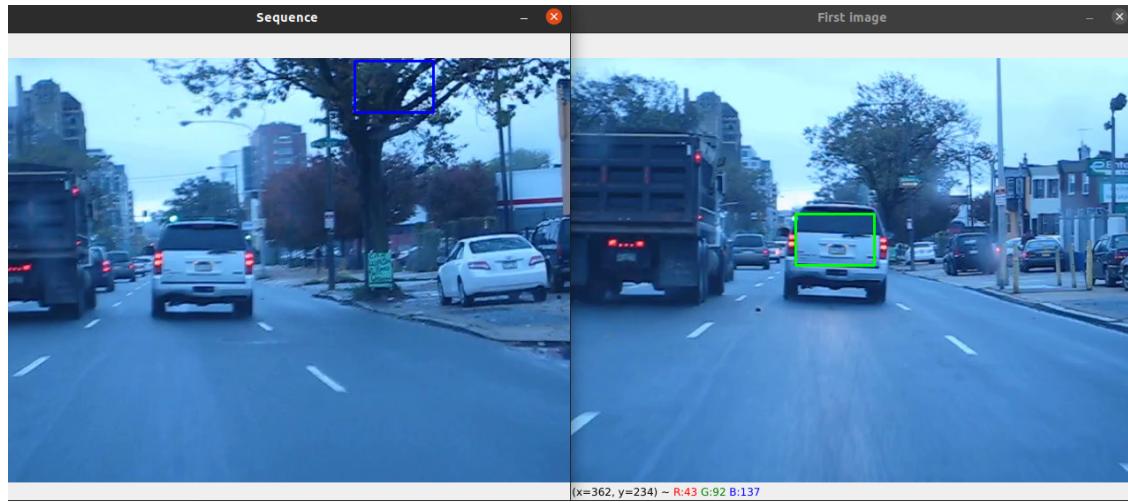


Figure 1.4 — suivi du ballon rouge avec *Mean Shift*. Ballon retrouvé



Le même est obtenu dans le cadre du suivi d'une voiture (voir *figure 1.5*). Notons ici l'instabilité de la caméra ainsi que les couleurs neutres de la voiture (les phares ne sont pas inclus dans le ROI initial — rectangle vert).

Figure 1.5 — suivi de la voiture avec *Mean Shift*



Encore une cause possible du changement rapide des distributions locales — zoom des vidéos. Les *figures 1.6 et 1.7* démontrent la démarche de *Mean Shift* avant et après la mise à l'échelle d'enregistrement de la promenade d'une femme.

Figure 1.6 — suivi d'une femme avec *Mean Shift*. Focus loinain

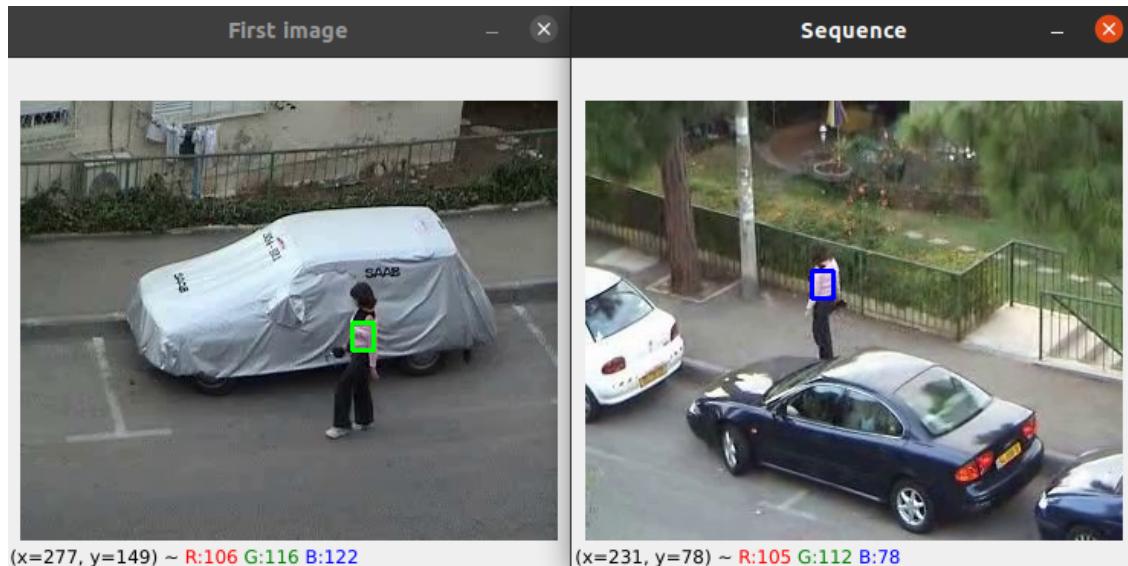


Figure 1.7 — suivi d'une femme avec *Mean Shift*. Focus mis à l'échelle



En revanche, la complexité et le temps de calculs de *Mean Shift* sont petits ce qui rend cette méthode acceptable pour les problèmes simples (objet clair sur le fond neutre, etc.) de suivi d'objets en temps réel, même si l'algorithme est sensible au choix de la RDI. Nous verrons par la suite, que la Transformée de Hough exige beaucoup plus de temps pour calculer chaque cadre avec le nombre important de points significatifs.

Q2 Analyser plus finement le résultat en affichant la séquence des poids à partir de la rétroprojection R_H de l'histogramme f_H de teinte, définie par $R_H(x, y) = f_H(H(x, y))$. Proposer et programmer des améliorations, en changeant la densité calculée et/ou en mettant en oeuvre une stratégie de mise à jour de l'histogramme modèle.

Grâce aux expériences réalisées dans les différentes vidéos, il a été possible d'apprécier comment la rétroprojection de l'histogramme de teinte $R_H(x, y)$ contient trop de bruit pour être utile à l'algorithme MeanShift, généralement dans la rétroprojection calculée directement l'objectif est confondu avec l'environnement dans les changements de lumière, ainsi que les changements brusques de position. De plus, les environnements ayant des tonalités similaires affectent fortement la localisation de la cible, bien qu'en général la précision soit très faible, les pires cas examinés dans les expériences étaient les vidéos de la voiture et du jeune homme, l'algorithme MeanShift ne parvient jamais à suivre les cibles, et cela se reflète clairement dans la rétroprojection.

Il est clair qu'un filtre sur la rétroprojection $R_H(x, y)$ doit être effectué afin que la cible puisse être différenciée au moment de l'application de l'algorithme MeanShift, un tel filtre peut être simplement un seuil sur la norme du gradient du cadre, c'est-à-dire supprimer (masquer) les valeurs de la rétroprojection qui ont une norme négligeable. Pour l'implémentation de ce filtre, le masque calculé à partir de la norme de gradient est obtenu à partir du script `voting_pixels.py` avec la fonction `orientation_mask`, qui reçoit le cadre ainsi que la valeur du quantile (`norm_quantile`) qui est utilisé sur la norme pour obtenir le seuil. L'application de ce filtre dans les différentes expériences (voir ci-dessous) démontre son efficacité pour le “nettoyage” de la rétroprojection, en laissant de côté pour un instant la grande différence de précision obtenue dans le suivi des objets, qui est également due en partie à la mise à jour du modèle, il est possible

d'apprécier sur les rétroprojections filtrées que les objectifs peuvent être plus facilement distingués de leur environnement, il suffit de regarder l'expérience de la voiture ou du jeune homme, où avec les quantiles de 0,7 et 0,95 respectivement, une quantité importante de bruit est éliminée, ce qui permet de distinguer visuellement ces objets sans problème.

Un autre aspect important est la mise à jour du modèle d'histogramme, pour cela une stratégie possible est de calculer l'erreur moyenne absolue (EMA) du modèle d'histogramme actuel par rapport au modèle candidat, qui est simplement le modèle d'histogramme calculé dans la position actuelle de la RDI, si cette erreur est inférieure à une précision donnée (seuil supérieur) alors le modèle peut être mis à jour avec le nouvel histogramme obtenu à partir de la RDI de la trame actuelle ; Cette procédure repose sur une idée simple : ne mettre à jour le modèle que si la RDI est “au-dessus” de la cible, il est clair que la définition de “au-dessus” ici ne peut pas être obtenue directement, nous utilisons donc l'EMA sur les histogrammes pour obtenir leur similitude, et ainsi savoir à quels moments il est correct de mettre à jour le modèle et à quels moments il ne l'est pas, pour éviter que le modèle ne soit mis à jour de manière incorrecte et représente un autre objet.

Avec l'application de la mise à jour du modèle et du filtre sur la rétroposition, des précisions dans le suivi des objets sont obtenues beaucoup plus élevées que les résultats présentés à la question précédente. Toutes les expériences réalisées sont présentées ci-dessous, et elles montrent les résultats obtenus avant et après les améliorations.

Figure 2.1 — suivi de la tasse avec *Mean Shift*

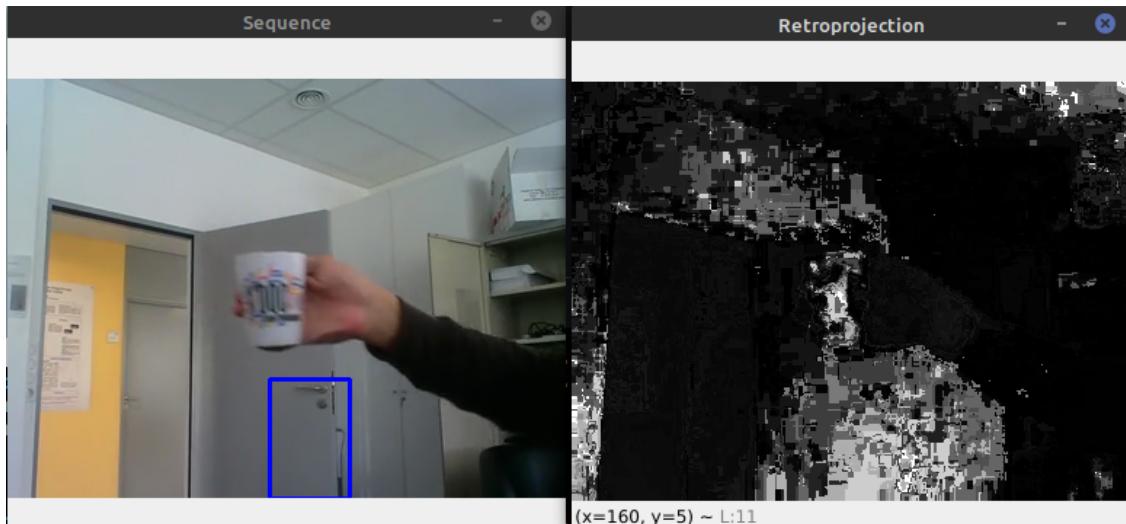


Figure 2.2 — suivi de la tasse avec *Mean Shift* amélioré, *norm_quantile=0.5*

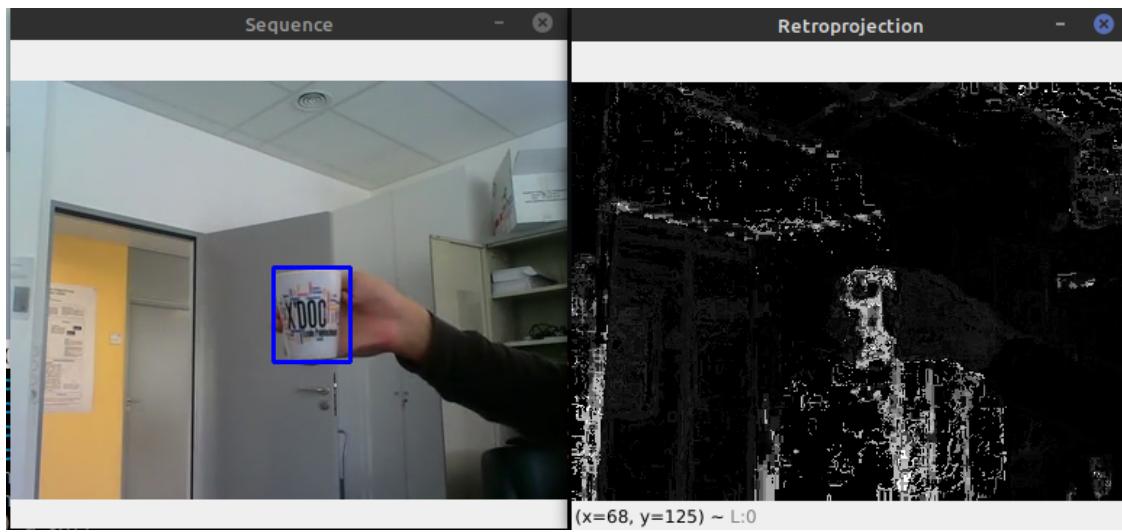


Figure 2.3 — suivi du ballon rouge avec *Mean Shift*

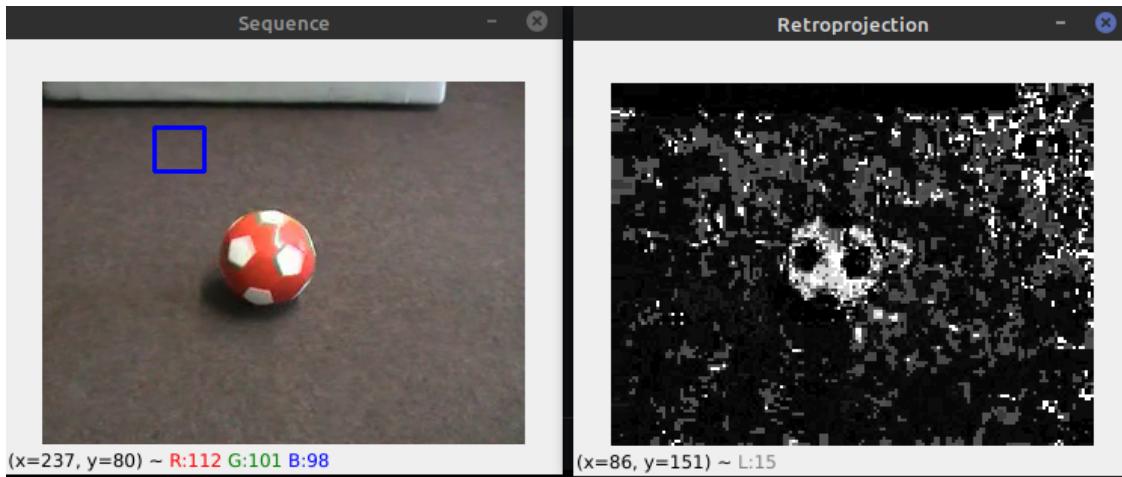


Figure 2.4 — suivi du ballon rouge avec *Mean Shift* amélioré, *norm_quantile=0.5*

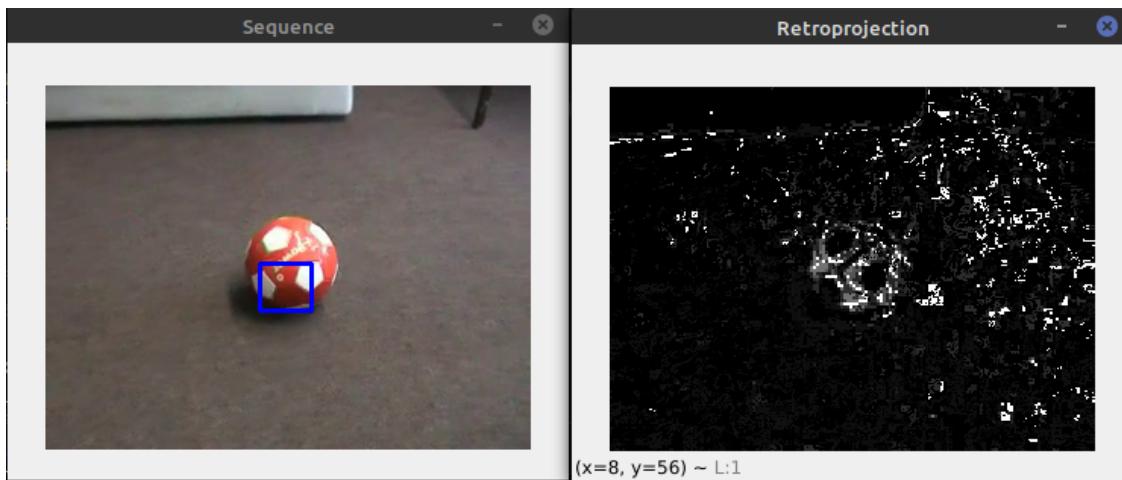


Figure 2.5 — suivi de la voiture avec *Mean Shift*

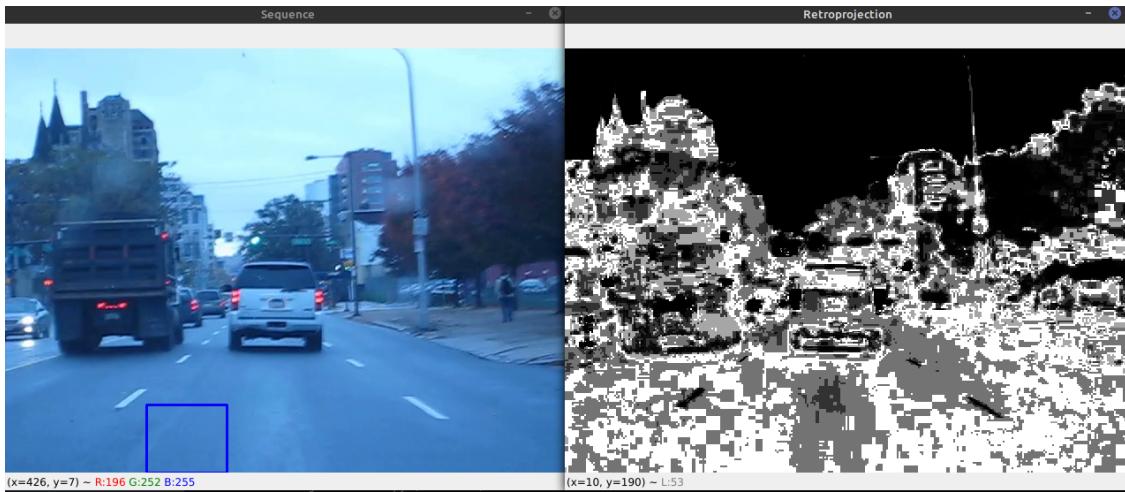


Figure 2.6 — suivi de la voiture avec *Mean Shift* amélioré, $norm_quantile=0.7$

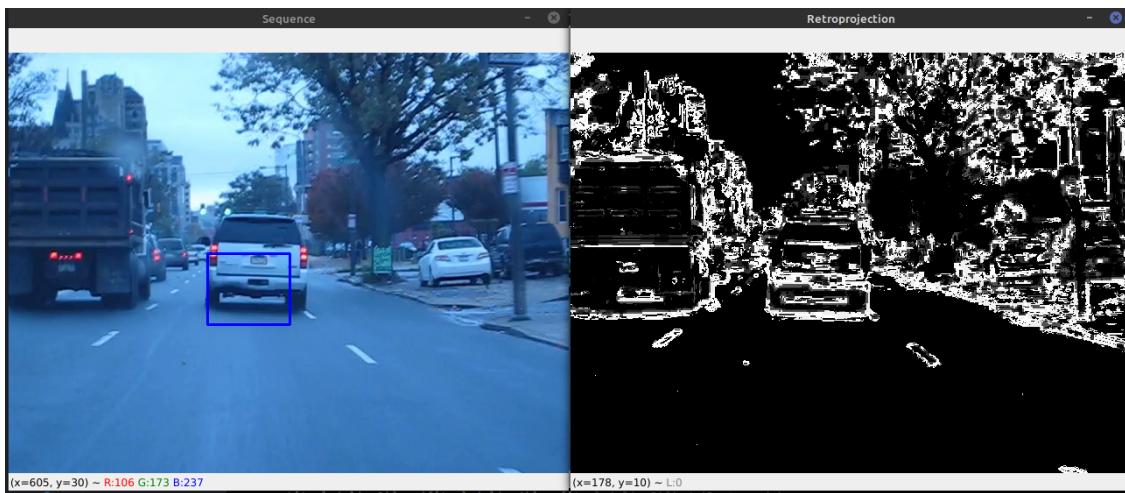


Figure 2.7 — suivi d'un visage dans les zones à l'éclairage contrasté avec *Mean Shift*

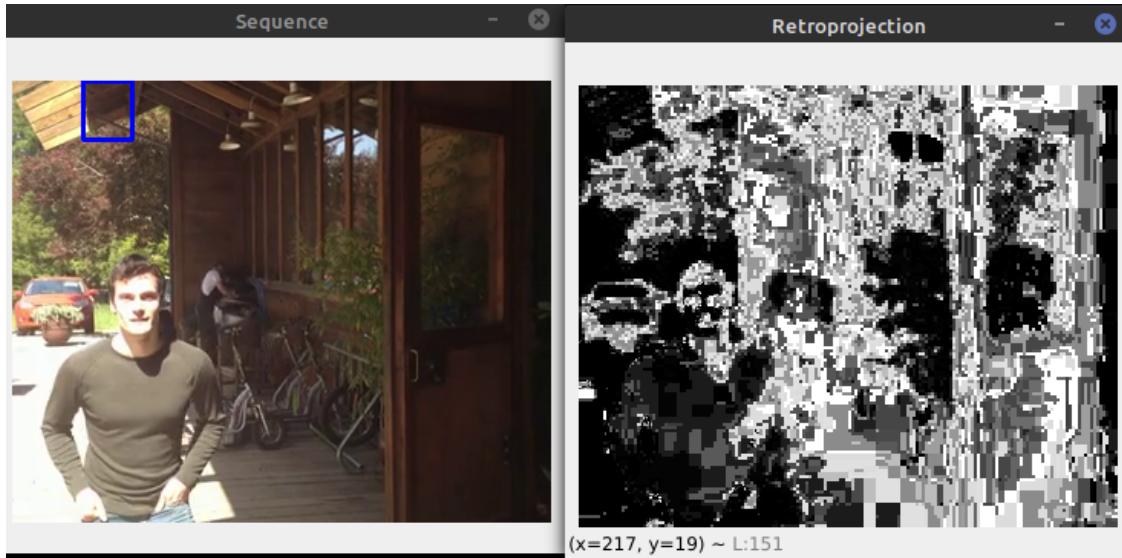
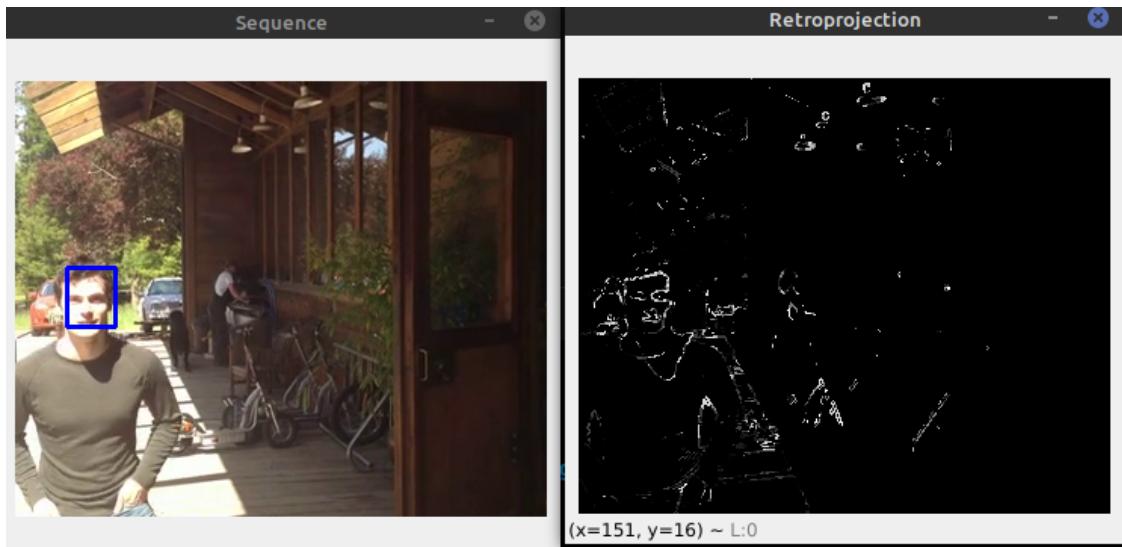


Figure 2.8 — suivi d'un visage dans les zones à l'éclairage contrasté avec *Mean Shift* amélioré, $\text{norm_quantile}=0.95$



2.2 Transformée de Hough

Q3 Calculer à chaque trame, l'orientation locale, i.e. l'argument du gradient des pixels de l'image, ainsi que le module du gradient. Définir un seuil sur le module du gradient pour masquer les pixels dont l'orientation n'est pas significative. Afficher ainsi la séquence des orientations où les pixels masqués apparaissent en rouge. L'objectif de cette question est de définir l'index de la *Transformée de Hough* (l'orientation), ainsi que l'ensemble des pixels votants, i.e. ceux dont l'orientation est significative.

Le code performant des transformations décrites ci-dessus se trouve dans le script `voting_pixels.py`. Les résultats de ses calculs sont présentés sur la figure 3.1.

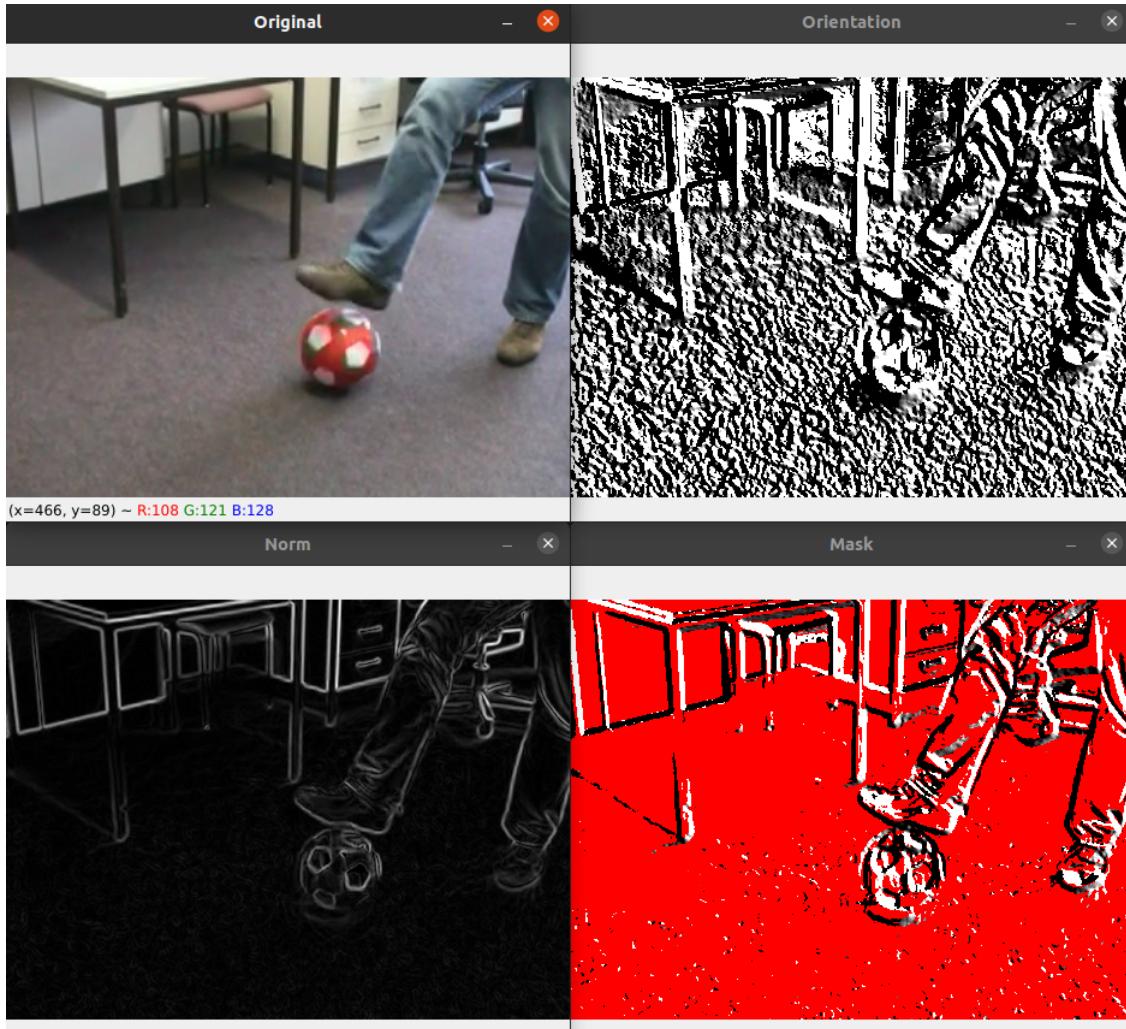
Figure 3.1 — orientation de gradient, sa norme et l'orientation masquée pour le cadre

de tasse



Le script lui-même peut être testé avec la commande suivante: `python voting_pixels.py` (qui supporte les images .jpg RGB), qui permet d'obtenir les images (voir *figure 3.2*) similaires aux exemples donnés dans l'énoncé.

Figure 3.2 — orientation de gradient, sa norme et l'orientation masquée pour le cadre de ballon



Q4 Construire un modèle de l'objet défini initialement sous la forme d'un modèle implicite indexé sur l'orientation (*R-Table*). Puis calculer la transformée de Hough associée sur toutes les images de la séquence. Calculer le suivi correspondant à la valeur maximale de la transformée de Hough à chaque image. Commenter et critiquer le résultat obtenu.

Le code de la Transformée de Hough est implémenté dans le script `Tracking_Hough_Transform.py` (`python Tracking_Hough_Transform.py`) ainsi que les calculs de *R-Table*. Selon la cumulation de pixels votants (dont la norme de gradient est suffisamment grande) dans les endroits spécifiques en fonction de leurs orientations, la Transformée de Hough résultante a de nombreuses régions linéaires distinctes (horizontales et verticales). Ses régions sont faciles à calculer pour les parties bien focussées d'image. La fonction `argmax` étant décisive, l'objet de suivi est bien dans la RDI, si ses contours sont clairs. Les *images 4.1* et *4.2* prouvent cette dépendance: si la vitesse de la tasse est petite (*figure 4.1* — l'inscription sur la tasse est claire) — elle est facile à suivre, lorsque la vitesse accroît et l'image de la tasse floue (*figure 4.2*) — l'objet est perdu. Par contre, le mur arrière a toujours une signature de Hough importante, ce qui attire parfois la RDI.

Figure 4.1 — suivi de la tasse avec la Transformée de Hough. Tasse sous focus. Nombre

d'orientations = 90

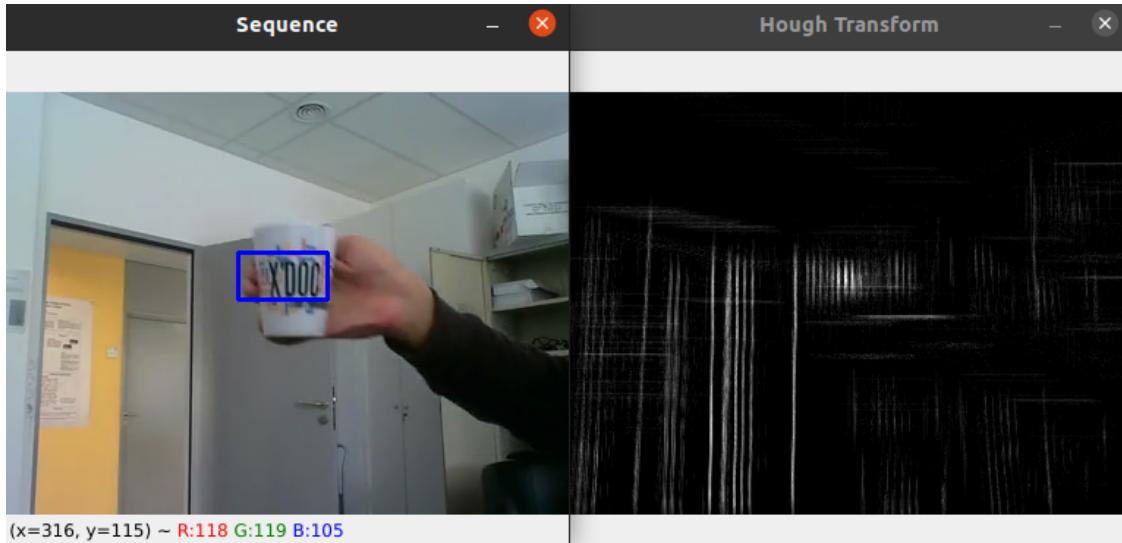
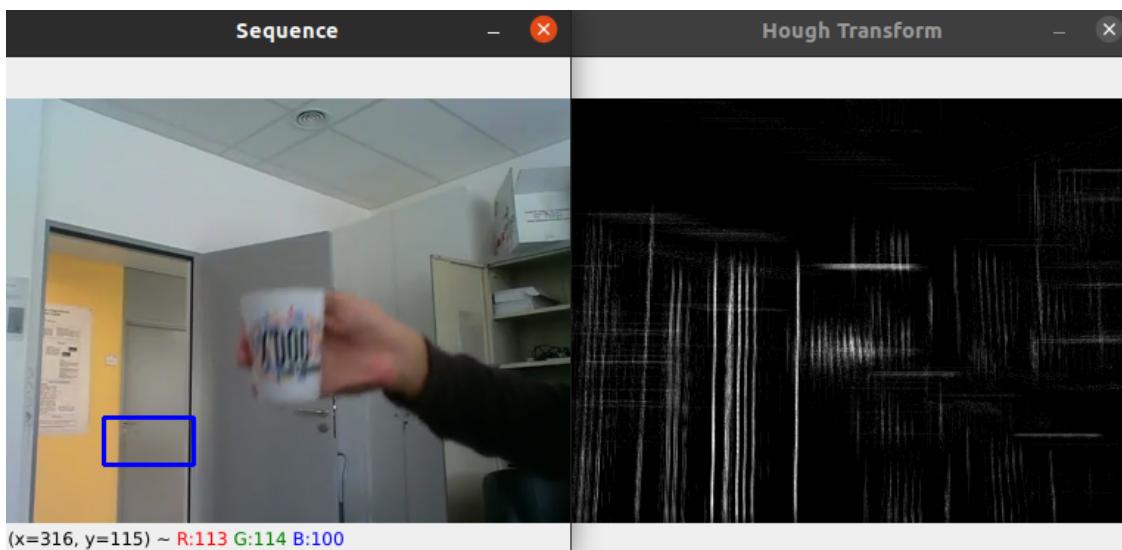


Figure 4.2 — suivi de la tasse avec la Transformée de Hough. Tasse perdue. Nombre d'orientations = 90



La situation est encore plus compliquée avec des objets qui n'ont pas de caractéristiques linéaires et de contours clairs. Par conséquent, le ballon se perd, entouré par les meubles rectangulaires (*images 4.4 et 4.6*) et il est suivi juste ailleurs de ses textures (*figures 4.3 et 4.5*).

Figure 4.3 — suivi du ballon avec la Transformée de Hough. Ballon sous focus loin de canapé. Nombre d'orientations = 180

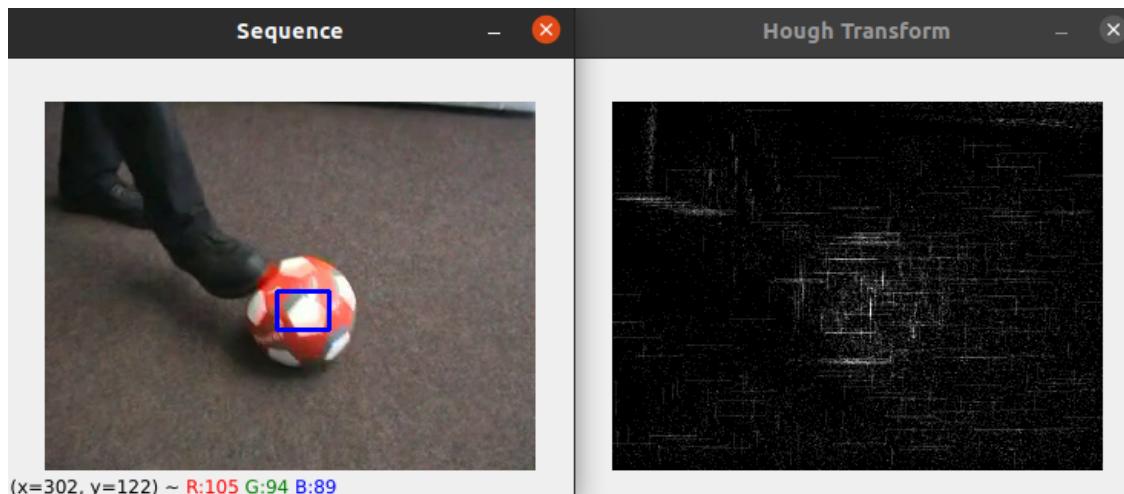


Figure 4.4 — suivi du ballon avec la Transformée de Hough. Ballon perdu prêt de canapé. Nombre d'orientations = 180

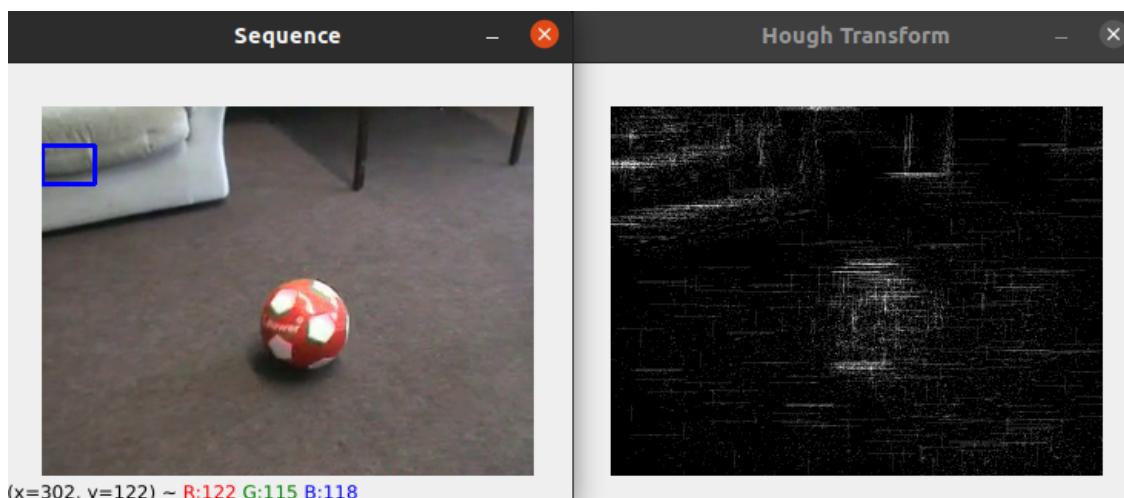


Figure 4.5 — suivi du ballon avec la Transformée de Hough. Ballon sous focus. Nombre d'orientations = 180

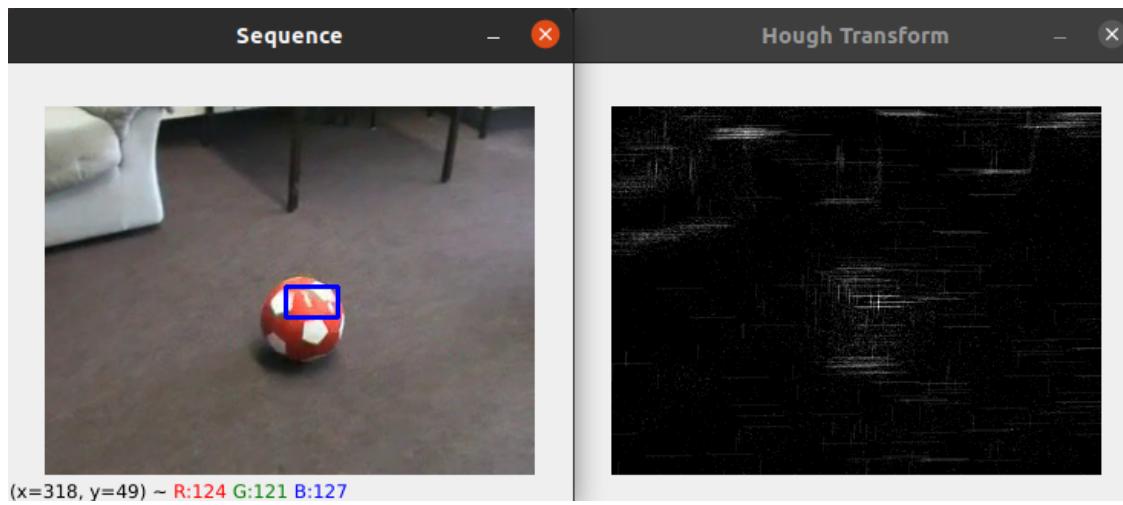
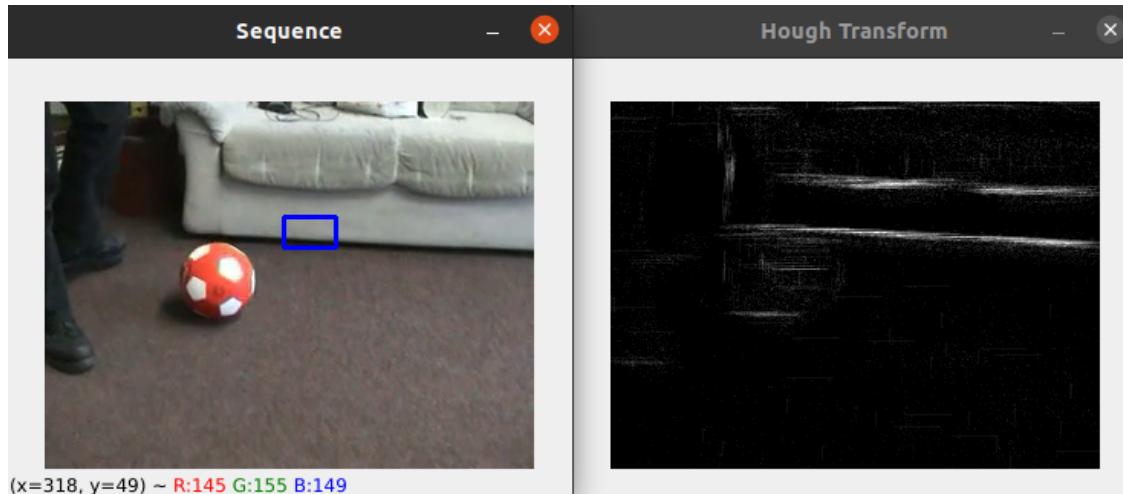


Figure 4.6 — suivi du ballon avec la Transformée de Hough. Ballon perdu. Nombre d'orientations = 180



Il faut également commenter le temps de calculs: avec beaucoup de points votants (30% de tous les points, réglé avec le quantile de la norme de gradient), chaque cadre est traité environ 0.6 secondes, l'impact sur la performance n'est nécessairement meilleur par rapport à un nombre moindre de points votants.

Q5 Remplacer le calcul du maximum par l'application du Mean Shift sur la transformée de Hough. Interpréter le résultat et le comparer avec le précédent. Proposer une stratégie de mise à jour du modèle qui permette de prendre en compte les déformations de l'objet.

La fusion de la Transformée de Hough avec *Mean Shift* améliore de suivi. En remplaçant *argmax* nous nous déplaçons vers la région la plus dense dont le centroid ne correspond forcément pas à la valeur maximale. Une telle approche permet de prioriser des régions avec un grand nombre moyen de votes (même s'ils sont dissipés) sur les points maximaux isolés. Ce qui donne le meilleur suivi de la tasse malgré la qualité basse de certains cadres (voir figures 5.1 - 5.3).

Figure 5.1 — suivi de la tasse avec la Transformée de Hough + Mean Shift. Nombre

d'orientations = 90

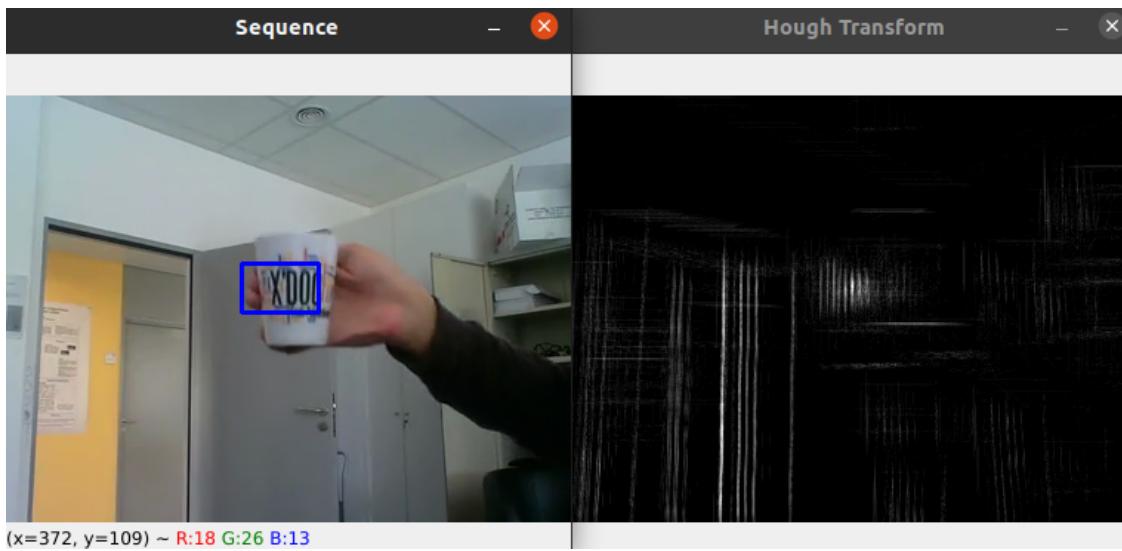


Figure 5.2 — suivi de la tasse avec la Transformée de Hough + *Mean Shift*. Nombre d'orientations = 90

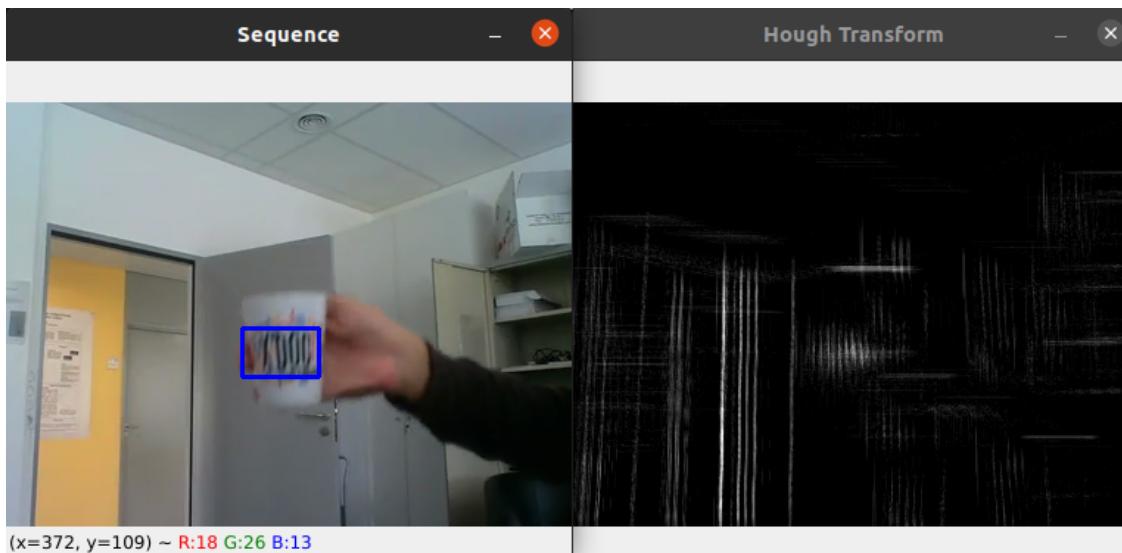
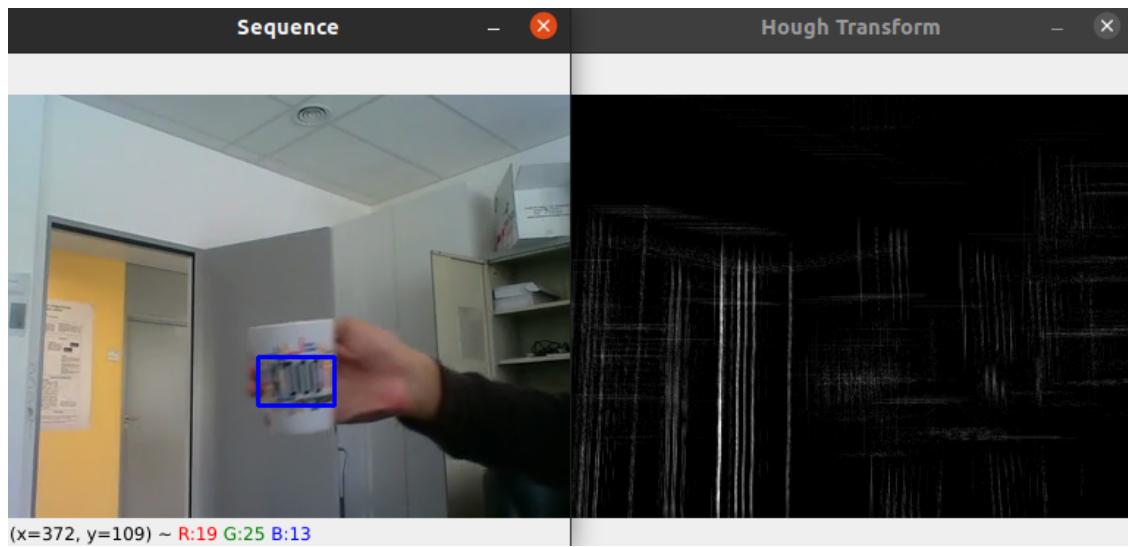


Figure 5.3 — suivi de la tasse avec la Transformée de Hough + *Mean Shift*. Nombre d'orientations = 90



Suivant l'approche décrite dans la question 2, nous pouvons utiliser la stratégie de mettre à jour *R-table*: lorsque la différence entre les orientations au sein de la RDI du modèle actuel et les orientations au sein de la RDI juste calculé est inférieure à une certaine précision. Par exemple, en utilisant une certaine erreur comme l'erreur quadratique moyenne ou l'erreur absolue moyenne.