

# ROB313

## Vision Robotique

### TP4 - Analyse vidéo et Tracking



Mateus LOPES RICCI  
Matheus MELO MONTEVERDE

16 janvier 2021

# Table des matières

<b>1</b>	<b>Mean Shift</b>	<b>4</b>
1.1	Différents canaux de HSV . . . . .	6
1.2	Méthode combinée entre les canaux de HSV . . . . .	7
1.3	Mise à jour de l'histogramme . . . . .	7
<b>2</b>	<b>Transformé de Hough</b>	<b>8</b>
2.1	Mise à jour de la R-Table . . . . .	13
2.2	Combinaison de méthodes . . . . .	13
<b>3</b>	<b>Conclusion</b>	<b>16</b>

## Table des figures

1.1	Algorithme Mean Shift . . . . .	4
1.2	Suivi par Mean Shift . . . . .	4
1.3	Frame 0017 - Vidéo VOT-Ball.mp4 . . . . .	5
1.4	Frame 0267 - Vidéo VOT-Ball.mp4 . . . . .	5
1.5	Frame 0267 - Vidéo VOT-Ball.mp4 . . . . .	5
1.6	Distribution des canaux du espace HSV . . . . .	6
1.7	Distribution de la Saturation - Vidéo Antoine_Mug.mp4 . . . . .	6
1.8	Intersection entre la distribution de la teinte et de la saturation - Vidéo VOT-Ball.mp4 . . . . .	7
2.1	Détermination de la norme du gradient, orientation et sélection des pixels représentatives . . . . .	9
2.2	Transformé de Hough . . . . .	11
2.3	Suivi de la balle par le valeur maximale de la transformée de Hough . . . . .	12
2.4	Suivi de la balle par la transformée de Hough et le Mean Shift . . . . .	13
2.5	Frame 0254 - Vidéo VOT-Ball.mp4 . . . . .	14
2.6	Frame 0033 - Vidéo Antoine_Mug.mp4 . . . . .	14
2.7	Frame 160 - Vidéo VOT-Car.mp4 . . . . .	15
2.8	Frame 0333 - Vidéo VOT-Woman.mp4 . . . . .	15

# Introduction

L'objectif de ce TP est d'étudier différentes propositions des algorithmes pour la suivi (Tracking) d'objets dans les séquences vidéo. Les applications du Video tracking sont diverses : l'interaction homme-machine, la sécurité et la surveillance, la communication et la compression vidéo, la réalité augmentée, le contrôle de la circulation, l'imagerie médicale, etc.

Pour ce TP, on expérimentera des solutions en utilisant des algorithmes bases de Mean Shift et de Transformées de Hough Généralisées. L'avantage de ces méthodes est de pouvoir choisir une région d'intérêt à suivre sans besoins d'avoir information préalable sur l'objet. En outre, il n'est pas nécessaire de prévoir des époques de traînement comme dans le cas des réseaux neuronaux.

# 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.

L'algorithme Mean Shift est un algorithme qui localise la valeur maximale de la fonction de densité. Il s'agit d'un algorithme itérative qu'estime le mode de la distribution à chaque itération.

L'algorithme Mean Shift a comme entrée la fenêtre d'observation dans le frame et la distribution à analyser. À chaque itération, le centre de gravité des points situés à l'intérieur de la fenêtre est calculé et le centre de la fenêtre est ensuite déplacé vers cette position. Cela se fait jusqu'à un nombre d'itérations souhaité ou un très petit mouvement de fenêtre. Cette méthode est illustré sur la figure suivante.

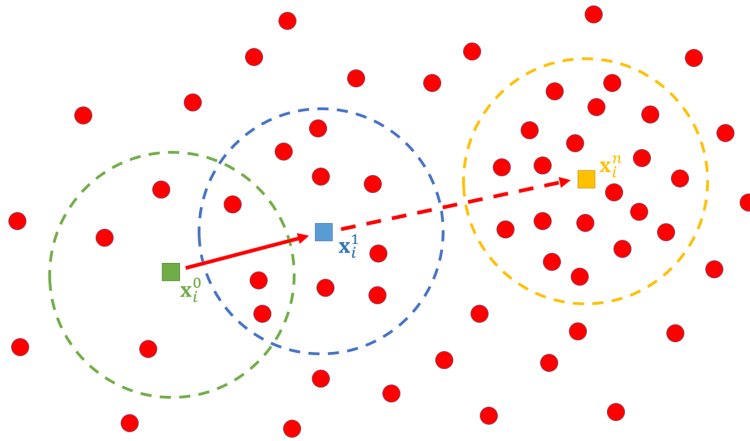


FIGURE 1.1 – Algorithme Mean Shift

L'un des principaux avantages de cet algorithme est qu'il peut être utilisé pour l'analyse de n'importe quelle distribution et en temps réel. Ainsi, il peut être utilisé dans les analyses et les applications les plus diverses.

Toutefois, comme on peut le constater à partir des tests effectués, la taille de la fenêtre a une grande influence sur les résultats du suivi. De cette façon, vous avez besoin d'un bon choix de fenêtre. Des études sur la taille de la fenêtre à utiliser peuvent être nécessaires pour une analyse plus approfondie. Parfois, il peut être intéressant d'utiliser une taille de fenêtre adaptative.

Les images suivantes illustrent le suivi par Mean Shift

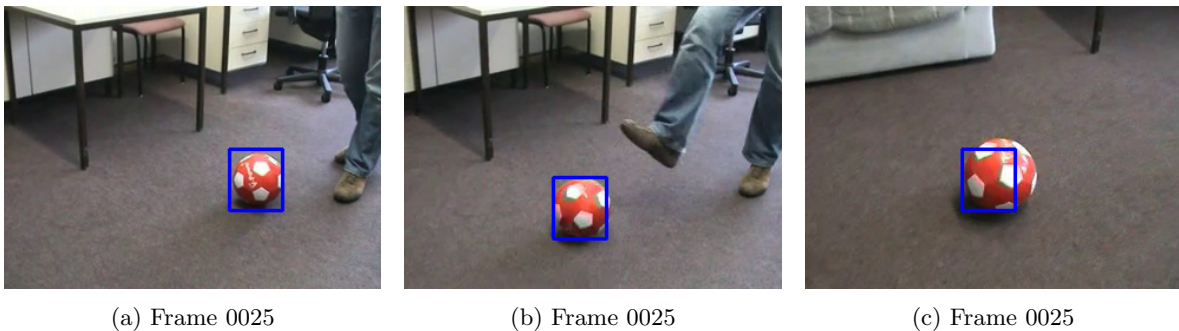
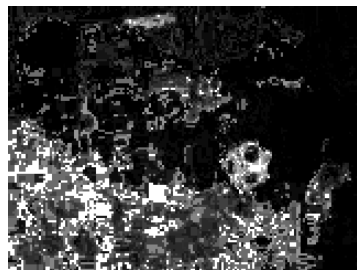


FIGURE 1.2 – Suivi par Mean Shift

**Q2. Analyser plus finement le résultat en affichant la séquence des poids à partir de la rétro-projection  $R_H$  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.**

Tout d'abord, l'analyse du suivi de la balle de la vidéo `VOT-Ball.mp4` a été faite, en affichant la distribution de la teinte et le résultat obtenu par Mean-Shift. On peut voir ci-dessous une image illustrant chacun de ces cadres.

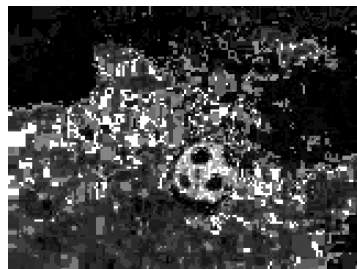


(a) Distribution de la teinte

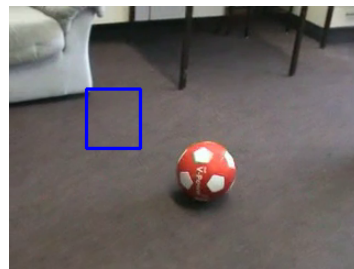


(b) Suivi par Mean-Shift

FIGURE 1.3 – Frame 0017 - Vidéo `VOT-Ball.mp4`



(a) Distribution de la teinte

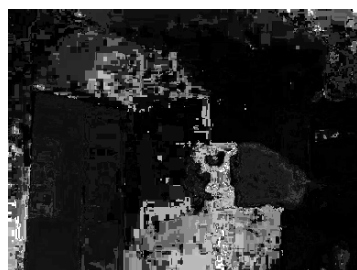


(b) Suivi par Mean-Shift

FIGURE 1.4 – Frame 0267 - Vidéo `VOT-Ball.mp4`

Par ces figures, on constate que par la rétroprojection de la distribution de la teinte, il n'est pas possible de faire une discrimination optimale de l'objet à suivre dans ce cas. Cela est dû au fait que la répartition de la teinte de la balle et du tapis sont proches. Même si les couleurs sont différentes, ces surfaces peuvent avoir des valeurs similaires dans ce canal de l'espace couleur HSV et des valeurs différentes pour les canaux de saturation et de valeur.

Le même phénomène est observé dans l'analyse d'une deuxième vidéo, `Antoine_Mug.mp4`.



(a) Distribution de la teinte



(b) Suivi par Mean-Shift

FIGURE 1.5 – Frame 0267 - Vidéo `VOT-Ball.mp4`

Pour essayer d'améliorer ces résultats, différentes configurations ont été proposées. Celles-ci sont décrites dans les sous-sections suivantes.

## 1.1 Différents canaux de HSV

Premièrement, une amélioration possible pourrait consister à analyser la distribution de l'histogramme d'autres canaux de l'espace couleur du HSV autres que la teinte, afin de vérifier si l'objet pourrait être bien discriminé par l'un d'entre eux. Dans les 3 figures suivantes, il est possible de voir la distribution de l'histogramme pour chacun de ces canaux.

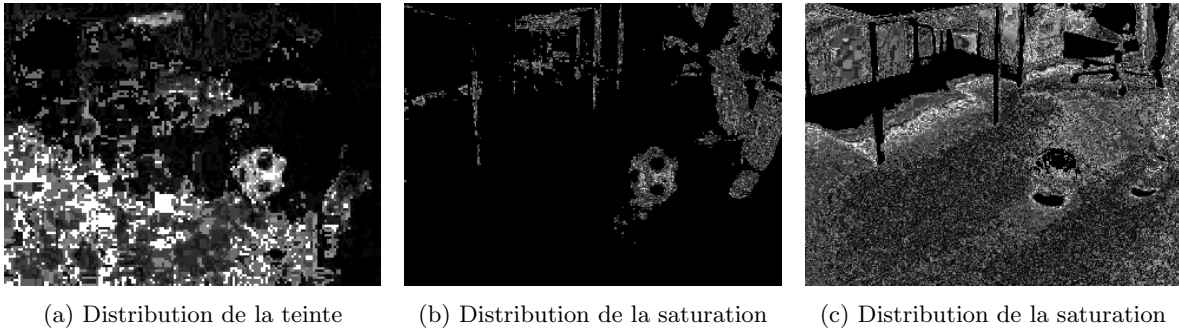


FIGURE 1.6 – Distribution des canaux du espace HSV

On constate à travers eux que la distribution concernant la saturation de l'image montre bien la caractéristique de la balle, ce qui pourrait aider à la discriminer. Déjà l'image du canal de valeur, il y a une distribution presque uniforme dans toute l'image, ce qui n'aide pas beaucoup à détecter l'objet.

Ainsi, un test de suivi de la balle a été effectué, en utilisant la distribution de la saturation. Dans ce nouvel essai, le suivi s'est mieux déroulé que celui utilisant le canal de Hue, en suivant la balle sur une période plus longue, mais la fenêtre a quand même été perdue au bout d'un certain temps.

De plus, cette discrimination par saturation n'est pas aussi bonne dans tous les cas comme on peut le constater dans le cas de la tasse (figure 1.8).

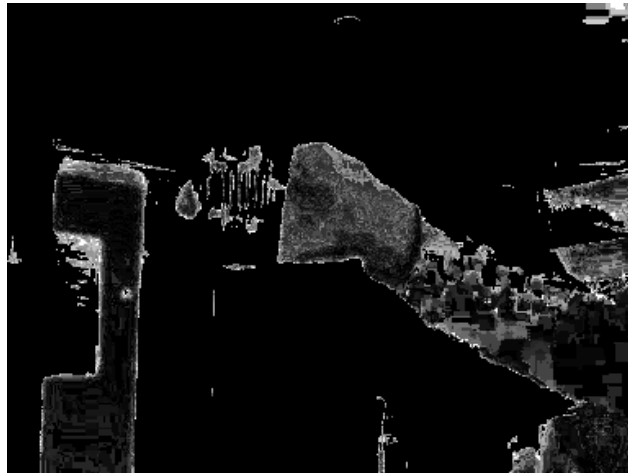


FIGURE 1.7 – Distribution de la Saturation - Vidéo Antoine\_Mug.mp4

## 1.2 Méthode combinée entre les canaux de HSV

Ensuite, une méthode a été proposée où les résultats des distributions des histogrammes de teinte et de saturation pouvaient être utilisés ensemble, puisque ceux-ci avaient des performances raisonnables pour discriminer l'objet dans la scène. Pour ce faire, chaque image de la vidéo a été calculée l'intersection entre la distribution des histogrammes entre ces deux canaux. De cette façon, il serait plus facile d'identifier l'objet à suivre, s'il avait peu de variation de couleur. La fonction python pour cela peut être vue ci-dessous :

```
1 # Combination of transformations
2 def Combine_HSV(bp_Hue, bp_Sat):
3     bp_Hue = cv2.convertScaleAbs(255*normalize(bp_Hue))
4     bp_Sat = cv2.convertScaleAbs(255*normalize(bp_Sat))
5
6     _, mask_and = cv2.threshold(bp_Sat, 30, 255, cv2.THRESH_BINARY)
7     Intersection = cv2.bitwise_and(bp_Sat, bp_Hue, mask = mask_and)
8     Intersection = cv2.convertScaleAbs(255*normalize(Intersection))
9     return Intersection
```

Code 1 – Intersection entre teinte et saturation

L'utilisation de cette fonction a permis d'observer une bonne discrimination de l'objet, dans le cas de la balle, et donc un bon suivi de celle-ci. Dans l'image suivante, on peut voir que l'intersection des ces deux canaux a permis d'exclure des parties de la scène qui n'étaient pas importantes, en mettant en évidence les pixels qui représentent la balle.

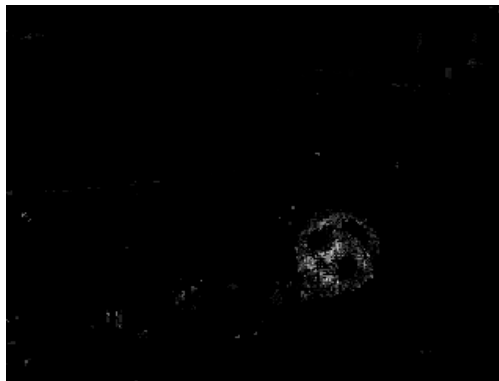


FIGURE 1.8 – Intersection entre la distribution de la teinte et de la saturation - Vidéo VOT-Ball.mp4

Même si cette méthode a obtenu un bon résultat dans ce cas, elle ne présente pas forcément les mêmes performances dans tous les cas. Cette méthode peut poser des problèmes pour détecter les objets qui ont des couleurs différentes et ceux qui peuvent changer de couleur.

## 1.3 Mise à jour de l'histogramme

Enfin, une proposition visant à mettre à jour les histogrammes de l'espace couleur HSV a été analysée. Même si cette méthode semble intéressante, une fois qu'il pourrait identifier un changement de couleur de l'objet à détecter, cela n'a pas été vérifié dans les tests effectués.

Avec la mise en œuvre de cette mise à jour de l'histogramme, le suivi pose des problèmes lorsque la région d'intérêt présente une légère erreur conduisant à ce que l'histogramme soit réalisé sur une surface qui n'est pas celle de l'objet souhaité. Par conséquent, la région d'intérêt propage cette erreur jusqu'à ce que la fenêtre perde l'objet.



## 2 Transformé 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. Un exemple de résultat attendu est illustré Figure 1.

Pour la suite, il est proposé la suivi des objets à l'aide de la Transformée de Hough. Cela est fait en déterminant les pixels le plus représentatives de l'image et ensuite on calcule la R-Table de la région d'intérêt qui sera comparé avec l'image pour obtenir la Transformé de Hough.

Pour déterminer les pixels le plus représentatives, on définit un seuil pour la norme du gradient des images. La norme du gradient est déterminé en calculant les approximations de la dérive par rapport à x et y à travers de l'utilisation des Kernels montré ci-dessous et en calculant la norme des résultats obtenu par l'application des kernels.

Pour approcher la dérivé partielle  $I_x = \frac{\partial I}{\partial x}$ , on utilise le noyaux suivant :

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Pour approcher la dérivé partielle  $I_y = \frac{\partial I}{\partial y}$ , on utilise le noyaux suivant :

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Ensuite, pour approcher le module du gradient  $||\nabla I|| = \sqrt{I_x^2 + I_y^2}$ , on utilise  $I_x$  et  $I_y$  appliqués à l'image et on prendre son norme. Finalement, pour calculer l'orientation des pixels de l'image on prend l'arc tangente des dérivés calculés,  $arg\nabla I = arctan(I_y, I_x)$ .

Pour la sélection des pixels plus représentatives, on a définit 0.2 du valeur maximale de la norme comme seuil.

Les images obtenus pour le module du gradient, l'orientation et sélection des pixels importantes sont montrés ci-dessous :

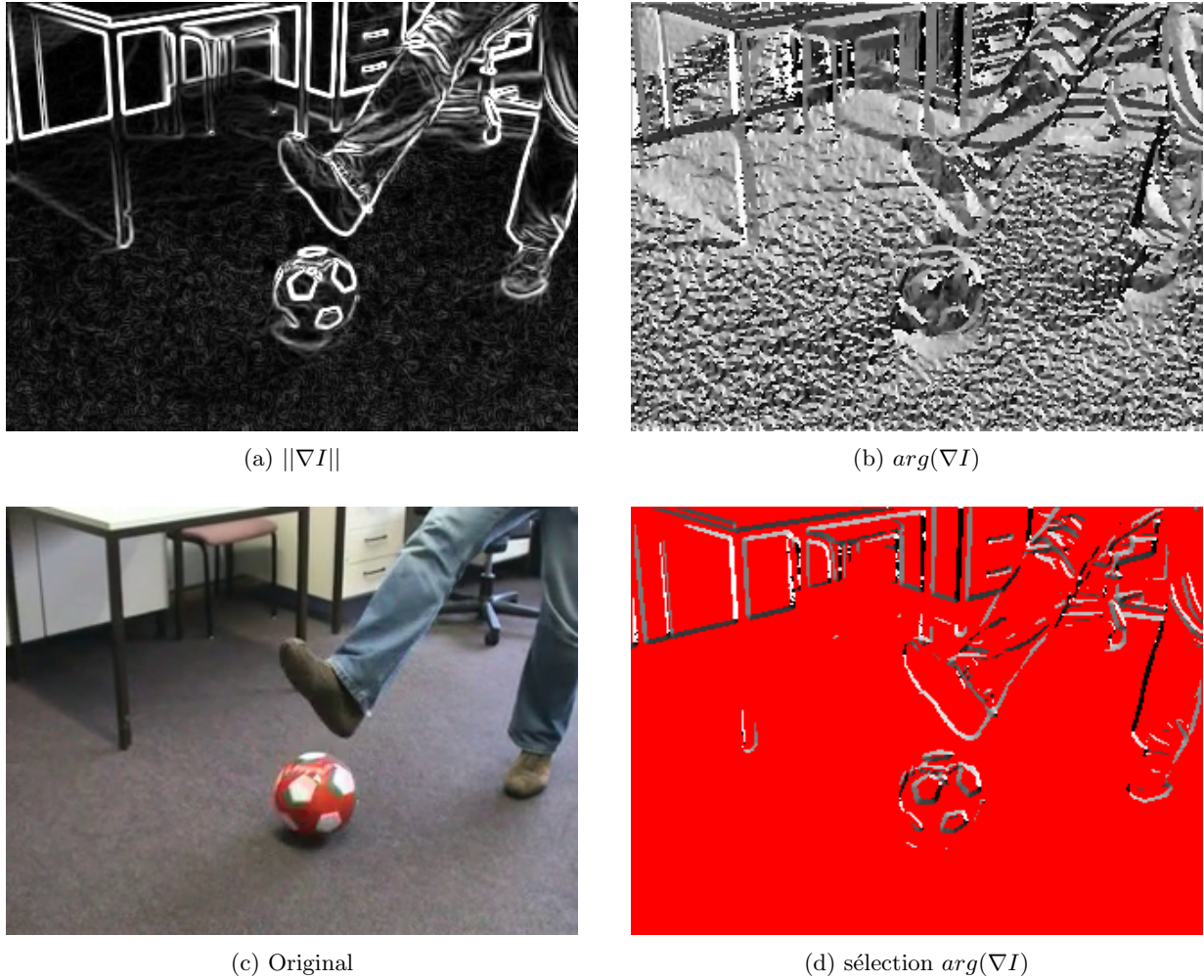


FIGURE 2.1 – Détermination de la norme du gradient, orientation et sélection des pixels représentatives

**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.**

Pour la suite, on calcule la R-Table pour regrouper les points d'intérêt et représenter l'objet sous la forme d'un modèle implicite indexé sur l'orientation. Ensuite, on calcule la transformée de Hough de l'image et on détermine le maxima de la fonction pour suivre l'objet souhaité.

#### Création de la R-Table

Soit  $\Omega$  le centre de la région défini pour l'objet, la R-Table est construit à partir des points d'intérêt présents dans cette région. Ici, on considère les points d'intérêt comme les pixels représentatives, définis comme une valeur plus grande qu'un seuil du module de la norme du gradient (Figure 2.1d).

Chaque point d'intérêt  $M$  de la région d'objet est indexé par l'orientation de l'image (Figure 2.1b), correspondant à une ligne de la R-Table. Ici, on a discrétisé l'orientation en 360 valeurs de  $]-\pi, \pi]$  pour la création de la R-Table.

Ensuite, on ajoute le vecteur déplacement  $\overline{M\Omega}$  dans la ligne correspondant à l'orientation du pixel.

La création de la R-Table est fait par la fonction ci-dessous. Cette fonction reçoit comme entrées, l'orientation de la région d'intérêt, les index le plus représentative de cette région et la discrétisation des angles.

```

1 def create_R_Table(arg_roi, indexs, dis_ang):
2     center = np.array( [ arg_roi.shape[1] // 2, arg_roi.shape[0] // 2 ] )
3     arg_roi = arg_roi * dis_ang // np.pi
4     R_Table = defaultdict(list)
5
6     for idx, idy in indexs:
7         dist = center - np.array([idx, idy])
8         R_Table[arg_roi[idx, idy]].append(dist)
9
10    return R_Table

```

Code 2 – Création R-Table

## Calcul de la Transformé de Hough

L'étape suivante est déterminer la Transformé de Hough de l'image. Cela est fait à partir de la comparaison des pixels le plus importants de l'image avec la R-Table créée pour la région d'intérêt. Cela est fait de la façon suivant :

- On initialise l'image avec  $\mathbf{H}(\mathbf{x}) = \mathbf{0}$  partout.
- Ensuite, pour tout point **représentatif**  $\mathbf{x}$  de l'image, soit  $\lambda(x)$  l'orientation de l'image :
  - **Pour toute** occurrence  $j$  de la R-Table associée a  $\lambda(x)$ , faire :
 
$$H(x + \delta_{\lambda(x)}^j) += \omega_{\lambda(x)}^j$$

L'extrait du code développé est montré ci-dessous. Les entrées de la fonction sont la R-Table défini pour la région d'intérêt, les valeurs de l'orientation du frame, les index les plus représentatives, le valeur de la discrétisation.

```

1 def Gen_Hough_Transform(R_Table, arg_frame, indexs, dis_ang, decay):
2     hough_img = np.zeros(arg_frame.shape, dtype = np.float64)
3     arg_frame = arg_frame * dis_ang // np.pi
4
5     for idx, idy in indexs:
6         coef = 1
7         for dist in R_Table[arg_frame[idx, idy]]:
8             dist_x = idx + dist[0]
9             dist_y = idy + dist[1]
10
11             if not (0 <= dist_x < arg_frame.shape[0]) or not (0 <= dist_y < arg_frame.shape[1]):
12                 continue
13
14             hough_img[dist_x, dist_y] += 1*coef
15             coef *= decay
16
17    return hough_img

```

Code 3 – Transformé de Hough

On montre ci-dessous quelques résultats pour la transformé de Hough.



(a) Frame 0023 - Vidéo VOT-Ball.mp4



(b) Transformé de Hough - Frame 0023



(c) Frame 0192 - Vidéo VOT-Ball.mp4



(d) Transformé de Hough - Frame 0192

FIGURE 2.2 – Transformé de Hough

À partir de la transformation de Hough on espère identifier les locales avec des caractéristiques similaires avec la région d'intérêt défini et il est exactement ce qu'on observe dans le figures 2.2.

Dans un première moment, avant que la balle commence à tourner, on a une grande similarité avec la région d'intérêt. Cela est vu dans la figure 2.2b, on a une concentration des point dans le centre de la balle.

Quand la balle commence à tourner, la région de la balle continue avec un grande similarité mais elle diminue un peu et on voit d'autres points avec une grand valeur (Figure 2.2d). Cela est due au fait que la région d'intérêt est défini par une position spécifique de la balle et, ainsi, ce algorithme n'est pas invariant à la rotation.

### Suivi par le maxima de la Transformé de Hough

Finalement, on exécute le tracking de la région d'intérêt par calculant le maxima de la transformé de hough. Le valeur maximal de la transformé de hough correspond au centre de la traking window. Le code pour la suivi est montré ci-dessous :

```

1 def Track_Hough(img_0, window):
2     img = img_0.copy()
3     _, _, h, w = window
4     max_pose_x, max_pose_y = mat_argmax(img)
5     window_hough = max_pose_x - (w // 2), max_pose_y - (h // 2), h, w
6
7     return window_hough

```

Code 4 – Suivi du valeur maximal de la Transformé de Hough

La suivi pour le valeur maximale de la transformé de hough est montré pour quelques frames ci-dessous :

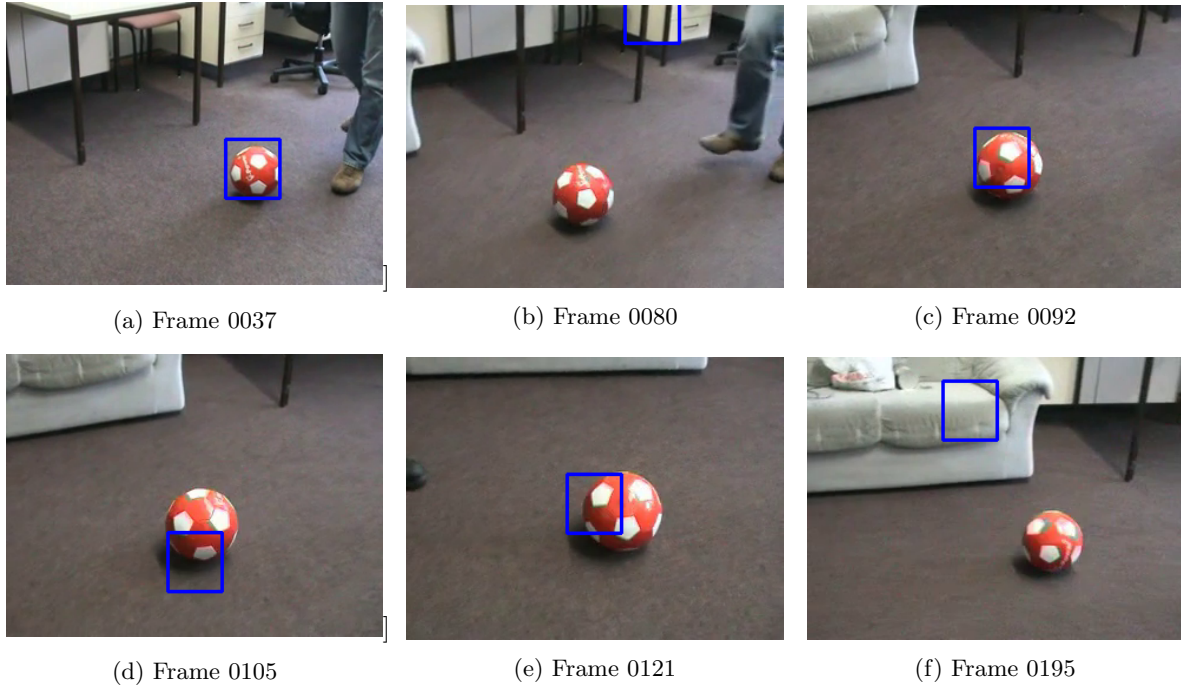


FIGURE 2.3 – Suivi de la balle par le valeur maximale de la transformée de Hough

Pour la suivi, on espère que la valeur maximale de la Transformé de Hough va être liée à la région le plus similaire à la région d'intérêt définie. Dans la figure 2.3, on peut voir que la suivi n'est pas très bonne pour tout les frames. Cela est due au fait que la transformé de Hough n'arrive pas à prendre en compte la variation pour rotation de la balle. Ainsi, quand la balle tourne, les valeurs de la transformé de Hough augmente en dehors de la région d'intérêt et parfois ces valeurs sont plus grand que dans la balle, ce qui amènera à un mauvaise suivi.

De cette façon, d'autres méthodes peuvent être utilisés ensemble avec la Transformé de Hough pour mieux déterminer la suivi de la région d'intérêt. Cela est abordé dans la section suivante.

**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.**

Avec la distribution des points de la transformée de Hough, obtenue dans la section précédente, il est possible d'utiliser le Mean Shift pour que la région d'intérêt suive les zones de plus forte concentration et qui sont proches de la région trouvée dans une image précédente de la vidéo. De cette façon, on évite que la fenêtre ne se déplace vers une région de la scène qui est loin de l'objet d'intérêt.

Avec la transformée de Hough et le Mean Shift, il y a eu un bon suivi de la balle dans la vidéo VOT-Ball.mp4, en suivant uniquement les pixels qui indiquent un bon accord avec la R-Table effectuée au début de la séquence. Les trois figures suivantes illustrent 3 moments du suivi de la balle.

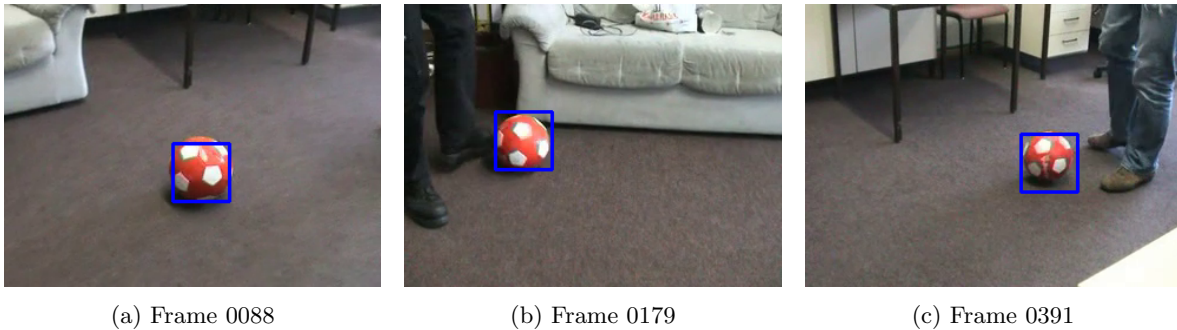


FIGURE 2.4 – Suivi de la balle par la transformée de Hough et le Mean Shift

Cette même méthode a été testée pour d'autres vidéos, dans lesquelles elle avait un bon suivi au début mais a fini par perdre l'objet d'intérêt. Cela peut être dû à une mauvaise qualité d'image causée par le mouvement rapide de la caméra ou de l'objet, qui rend les contours moins définis. Un autre problème est le changement de forme ou de caractéristique de la région d'intérêt, comme c'est le cas d'une personne en mouvement.

Pour essayer d'améliorer encore cette méthode, deux nouvelles configurations ont été proposées et analysées.

## 2.1 Mise à jour de la R-Table

Afin d'identifier les éventuels changements de forme de la région d'intérêt, une mise à jour de la R-Table a été implémentée à chaque frame de la séquence.

Malheureusement, les résultats obtenus par cette méthode n'ont pas été aussi bons que prévu. Une fois de plus, le mouvement rapide de la scène provoque une grande dégradation dans la définition des contours qui génère la création d'un R-Table peu représentatif. De plus, une éventuelle obstruction de l'objet à suivre, même si elle est rapide, provoque la création d'une R-Table qui représente une autre surface. La fenêtre finira donc par suivre cette nouvelle surface.

## 2.2 Combinaison de méthodes

Enfin, une méthode combinant la méthode des histogrammes des canaux HSV (section 1.2) et de la transformée de Hough a été proposée. La fonction suivante a été utilisée à cet effet.



```

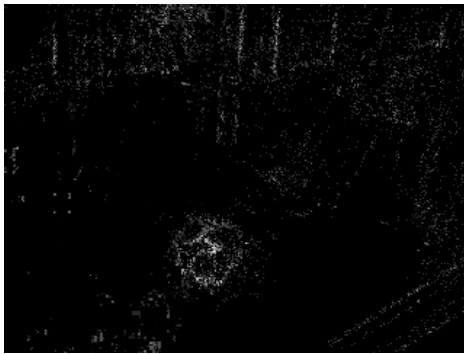
1  # Combination of transformations
2  def Combine_all(bp_Hue, bp_Sat, HG):
3      HG = cv2.convertScaleAbs(255*normalize(HG))
4      Intersection = Combine_HSV(bp_Hue, bp_Sat)
5      Comb = Intersection + HG
6      normalize(Comb)
7      return Comb

```

Code 5 – Combinaison de méthodes

Cette méthode reçoit l'intersection entre la distribution des histogrammes de la teinte et de saturation et ajoute la distribution de la transformée de Hough. Le résultat est ensuite normalisé. La somme de ces deux matrices fait que les points communs ont une plus grande valeur et ont donc une plus grande influence sur le suivi.

Cette nouvelle méthode a donné de bons résultats, permettant de bien suivre à la fois la balle et la tasse, comme l'illustrent les figures ci-dessous.

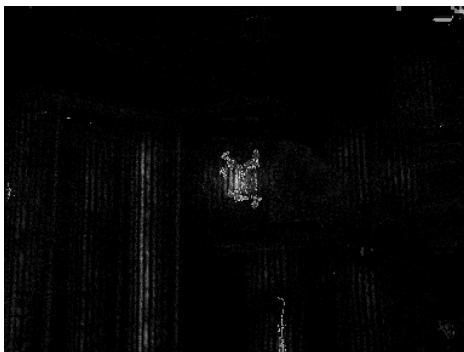


(a) Distribution des histogrammes de la teinte et de la saturation avec la transformée de Hough



(b) Suivi par Mean-Shift des histogrammes de la teinte et de la saturation avec la transformée de Hough

FIGURE 2.5 – Frame 0254 - Vidéo VOT-Ball.mp4



(a) Distribution des histogrammes de la teinte et de la saturation avec la transformée de Hough



(b) Suivi par Mean-Shift des histogrammes de la teinte et de la saturation avec la transformée de Hough

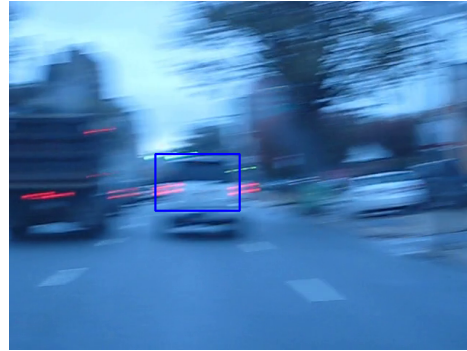
FIGURE 2.6 – Frame 0033 - Vidéo Antoine\_Mug.mp4

A travers ces résultats, on peut voir que cette nouvelle méthode, présente une bonne performance. En plus de ne favoriser que les points les mieux définis et les plus significatifs, il est capable de corriger des erreurs occasionnelles dues à la transformation de Hough ou aux histogrammes de couleur. De cette façon, même en cas de mauvaise qualité des images ou d'objets aux couleurs peu distinctes, il est possible de suivre raisonnablement bien la région d'intérêt.

En testant cette nouvelle méthode dans d'autres vidéos, on peut voir que même avec quelques petites erreurs, il était possible de suivre certains objets qui bougent beaucoup, comme la voiture et la fille qui marche dans la rue. Les images suivantes illustrent ces cas.



(a) Distribution des histogrammes de la teinte et de la saturation avec la transformée de Hough

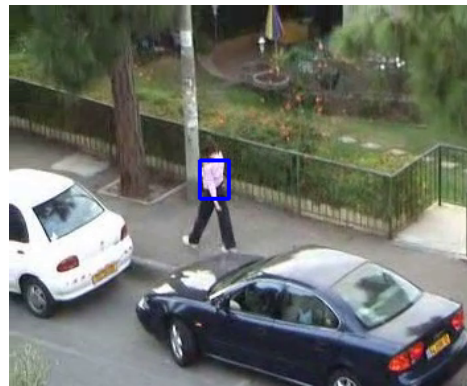


(b) Suivi par Mean-Shift des histogrammes de la teinte et de la saturation avec la transformée de Hough

FIGURE 2.7 – Frame 160 - Vidéo VOT-Car .mp4



(a) Distribution des histogrammes de la teinte et de la saturation avec la transformée de Hough



(b) Suivi par Mean-Shift des histogrammes de la teinte et de la saturation avec la transformée de Hough

FIGURE 2.8 – Frame 0333 - Vidéo VOT-Woman .mp4

Cependant, cette méthode présente encore quelques problèmes pour suivre les régions d'intérêt dans des scènes sont plus complexes et avec beaucoup de couleurs et des scènes où la région d'intérêt change beaucoup la luminosité, comme c'est le cas des vidéos VOT-Basket .mp4 et VOT-Sunshade .mp4, respectivement.



### 3 Conclusion

Dans ce travail, certaines méthodes de suivi d'une région d'intérêt sur différentes vidéos ont été analysées. Les régions d'intérêt ont été définies en traçant un rectangle sur l'objet souhaité.

Tout d'abord, une étude a été réalisée sur le suivi par la distribution des histogrammes des canaux de l'espace colorimétrique HSV pour chaque trame de la séquence. Ensuite, la région d'intérêt a été mise à jour pour chaque trame par l'algorithme de Mean Shift. A travers ces techniques, on a observé un bon suivi des objets qui ont des couleurs particulières dans les scènes.

Dans un deuxième temps, la méthode de la transformée de Hough a été analysée, par laquelle il a été possible d'identifier les endroits de l'image qui ont des caractéristiques plus similaires à la région d'intérêt sélectionnée. Dans cette méthode, le suivi avait des problèmes en cas de mouvements soudains de la caméra ou des objets et aussi par le changement déformé des objets choisis.

Enfin, une méthode combinée a été mise en pratique, ce qui a permis une meilleure performance dans le suivi des surfaces sélectionnées. Dans cette combinaison, les petites erreurs liées à l'une des méthodes pourraient être corrigées par les autres.

Bien que la méthode combinatoire a obtenu de bonnes performances pour des vidéos plus simples, on ne peut pas affirmer la même chose pour des vidéos plus complexes avec beaucoup d'ombres. Ainsi, pour que le suivi soit bien effectué dans ces cas, des méthodes plus précises seraient nécessaires. Pour cela, on peut citer la détection d'objets à l'aide de réseaux de neurones, qui sont largement utilisés pour ces fonctions.