

# TP4-Analyse vidéo et Tracking

Yan CHEN

Janvier 2020

## 1 Mean Shift

### 1.1 Théorie de Mean Shift

#### 1.1.1 Principale de MeanShift

1. Choisir la région d'objet On choisit la région d'objet dans le première frame de vidéo. Comme indique à la Figure 1.



FIGURE 1 – Région d'objet

2. Transformer la région d'objet de l'espace RGB à l'espace HSV. On peut le réaliser avec le code suivant :

```
hsv_roi = cv2.cvtColor(roi, cv2.COLOR_BGR2HSV)
```

3. Calculer le historisme marginal de Hue composant de la région d'objet. On extrait le Hue composant et supprime l'arrière-plan, puis compte le nombre de chaque valeur de Hue et fait la normalisation de histogramme. On le réalise avec le code suivant :

```
mask = cv2.inRange(hsv_roi, np.array((0.,30.,20.)), np.array
((180.,255.,235.)))
# Marginal histogram of the Hue component
roi_hist = cv2.calcHist([hsv_roi],[0],mask,[180],[0,180])
# Histogram values are normalised to [0,255]
cv2.normalize(roi_hist,roi_hist,0,255,cv2.NORM_MINMAX)
```

4. Calculer back projection. La valeur de chaque pixel de la histogramme à back projection reflète la probabilité que le pixel soit la cible suivie. Le code suivant :

```
hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    # Backproject the model histogram roi_hist onto the
    # current image hsv, i.e. dst(x,y) = roi_hist(hsv(0,x,y))
dst = cv2.calcBackProject([hsv],[0],roi_hist,[0,180],1)
```

5. Appliquer Mean Shift. Au début, déterminer la size de région d'objet et location initiale sur frame prochaine(même avec frame dernière). Puis, calculer la centre de la région d'objet et le laisser être la centre de nouvelle région d'objet, et la size de la région ne change pas. Répéter de calculer la centre de la région pour obtenir les nouvelles régions d'objet jusqu'à ce que la région d'objet ne bouge pas ou qu'elle soit inférieure à un certain seuil. Le code suivant :

```
# apply meanshift to dst to get the new location
ret, track_window = cv2.meanShift(dst, track_window, term_crit)
```

Lorsque l'algorithme MeanShift est utilisé pour la poursuite de cibles vidéo, l'histogramme couleur de la cible est utilisé comme fonction de recherche, et en itérant continuellement le vecteur MeanShift fait converger l'algorithme vers la position réelle de la cible, de manière à atteindre l'objectif de poursuite. Normalement, il a des avantages et des inconvénients. J'ai énuméré ci-après certains de ses avantages et inconvénients par expérimentation.

### 1.1.2 Avantages et limitations

**Avantages :**

1. **Pas trop intensif en termes de calcul, entièrement capable de suivre en temps réel lorsque la zone cible est connue.** Figure 2 présente les temps d'exécution MeanShift par fois. On peut voir qu'il faut du 3e-4 seconds au 5e-4 seconds.

```
time : 0.00034618377685546875
time : 0.0004639625549316406
time : 0.0003681182861328125
time : 0.0003859996795654297
time : 0.0004048347473144531
time : 0.00037980079650878906
time : 0.00038504600524902344
time : 0.0004067420959472656
time : 0.0002658367156982422
time : 0.0003921985626220703
```

FIGURE 2 – Temps de exécution de MeanShift

2. **Un modèle d'histogramme des fonctions du noyau est utilisé, qui est insensible à l'occlusion des bords, à la rotation de la cible, à la distorsion et au mouvement du fond.**

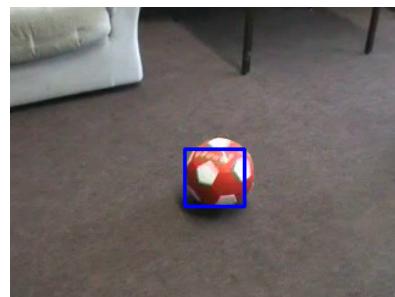


FIGURE 3 – Balle en rotation. Il n'est pas sensible à la rotation de la cible.

**3. Il est insensible à l'occlusion des bords.**



FIGURE 4 – La tête de femme est à l'occlusion. Il a encore une bonne performance de poursuivre.

**Inconvénients :**

- 1. Le suivi échoue lorsque l'échelle de la cible change beaucoup,** car la taille de la fenêtre reste constante pendant le suivi.

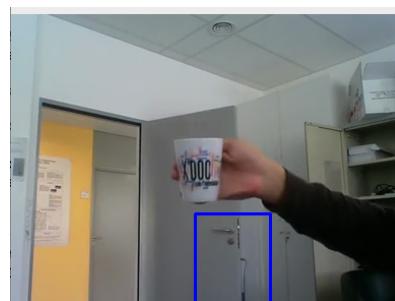


FIGURE 5 – Performance de suivre n'est pas bonne.

- 2. La poursuite n'est pas efficace lorsque le caméra n'est pas stable bien.**



FIGURE 6 – Performance de suivre n'est pas bonne lorsque caméra n'est pas stable.

3. Les caractéristiques des histogrammes sont légèrement insuffisantes en termes de caractérisation des couleurs cibles et manquent d'informations spatiales.

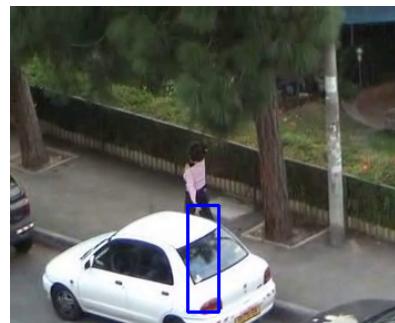
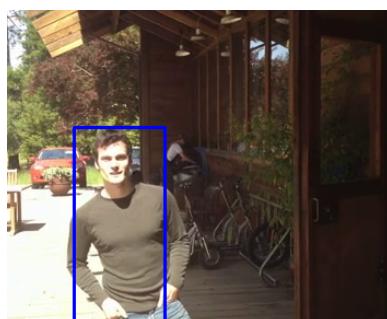
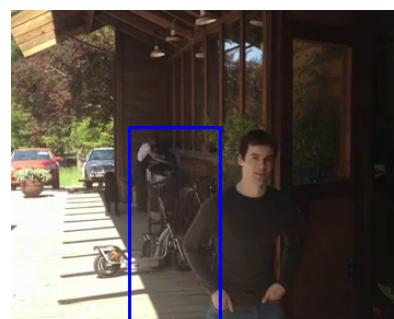


FIGURE 7 – Performance de suivre n'est pas bonne lorsqu'il y a couleur composant similaire avec l'autre région dans la frame.

4. Il est sensible à luminosité d'image.



(a) Détection correcte



(b) Détection incorrecte

FIGURE 8 – Il est sensible à luminosité d'image. On a choisi la région d'objet avec luminosité haute. Détection correcte pour objet dans la région d'image avec haute luminosité. Mais détection incorrecte pour objet dans la région d'image avec luminosité faible.

## 1.2 Analyse de la rétro-projection de histogramme de teinte et amélioration TP

5. Il ne fonction pas bien pour les objets déformés.



(a) Région d'objet



(b) Détection incorrecte

FIGURE 9 – Joueur de basket-ball cours dans la vidéo. L'algorithme ne peut pas détecter l'objet correctement.

### 1.2 Analyse de la rétro-projection de histogramme de teinte et amélioration

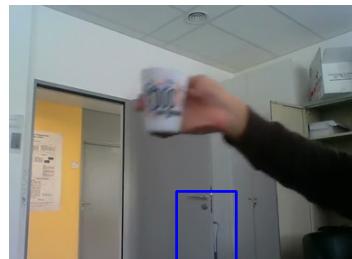
#### 1.2.1 Visualisation de la rétro-projection de histogramme de teinte



(a) Détection correcte



(b) La rétro-projection de histogramme de teinte pour détection correcte



(c) Détection correcte



(d) La rétro-projection de histogramme de teinte pour détection incorrecte

FIGURE 10 – Détection d'objet et la rétro-projection de histogramme de teinte. Il existe la région de haute densité dans la plan-arrière. Particulièrement, la région en base.

On affiche des rétro-projections de histogramme de teinte de la vidéo "Antoine\_Mug.mp4" pour analyser plus finement dans la Figure 10. La région est plus noire, la densité est plus faible. Par contre, la région est plus blanche, la densité est plus haute. On peut voir que région d'objet évidemment a plus haute densité. Mais il existe aussi la région de la densité haute dans la plan-arrière. Donc lorsque objet déplace près de la région de haute densité dans la plan-arrière. Après d'algorithme MeanShift, la région d'objet converge à la région de haute densité dans la plan-arrière. Il ne peut pas donc poursuivre l'objet bien.

### 1.2.2 Amélioration

#### 1. Changer la densité calculée

L'idée de changer la densité calculée est de réduire la densité calculée qui ne sont pas dans la région d'objet. On ajoute le code suivant après `dst` (75ème ligne). Le code fixe à 0 les valeurs de pixels inférieur à seuil. En autre terme, ils sont fixés à région sans densité.

```
dst[np.where(dst[:, :] < 70)] = [0]
```

Séquence	Seuil	Note
Antoine-Mug	210	bien
VOT-Ball	75	bien
VOT-Basket	none	
VOT-Car	none	
VOT-Sunshade	none	
VOT-Woman	70	pas bien

TABLE 1 – Seuil à chaque séquence

En testant sur chaque vidéo, on obtient le tableau de seuil suivant. Comme on le voit, on réussis de trouver des seuils pour les vidéos Antione-Mug et VOT-Ball. L'algorithme fonctionne très bien. Mais on ne trouve pas les seuils propres pour les vidéos VOT-Basket, VOT-Car and Sunshade.

En fait, dans la vidéo VOT-Basket, l'algorithme original fait un bon travail de suivi des joueurs portant des vêtements sombres. Mais pour les joueurs qui portent du blanc, ni l'algorithme original ni l'algorithme amélioré ne fonctionnent bien. Lorsque le seuil est fixé à 230, le nouvel algorithme peut suivre les joueurs portant du blanc, mais il a tendance à tomber dans la région de non-densité.

Dans la vidéo VOT-Car, il n'est pas possible de trouver le bon seuil pour pouvoir suivre la voiture car la teinte de la voiture était relativement similaire à la teinte de l'arrière-plan.

Dans la vidéo VOT-Sunshade, la teinte des vêtements de la personne est similaire à la couleur du fond dans l'ombre, il est donc impossible de trouver un

## 1.2 Analyse de la rétro-projection de histogramme de teinte et amélioration TP

seuil approprié pour suivre la région cible, y compris les vêtements. Mais on peut faire du pantalon de la personne la région cible, car la teinte du pantalon est très différente de la teinte du fond, et on peut trouver un seuil approprié pour un suivi précis de la cible. La valeur seuil est de 50. mais lorsque la personne se déplace très vite, il est facile de tomber dans la zone sans densité.

Dans la vidéo dernière, on trouve un seuil 70, mais l'algorithme ne peut pas fonctionne parfait pour tous les seuils possibles car la teinte de voiture est similaire à la teinte de la femme. En fait, Le seuil 70 est un bon choix relativement pour suivre la femme parmi tous les seuils. Mais on fait le vêtement la région cible, l'algorithme peut suivre l'objet bien.

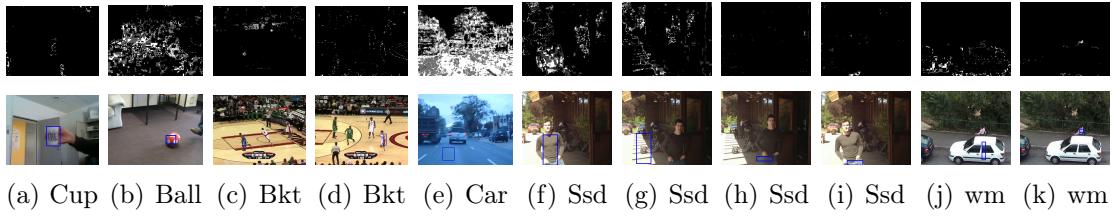


FIGURE 11 – Les résultats d'algorithme de changer la densité. Bkt : Basket, Ssd : Sunshade, wm : Woman.

### 2. Mis à jour de l'histogramme modèle

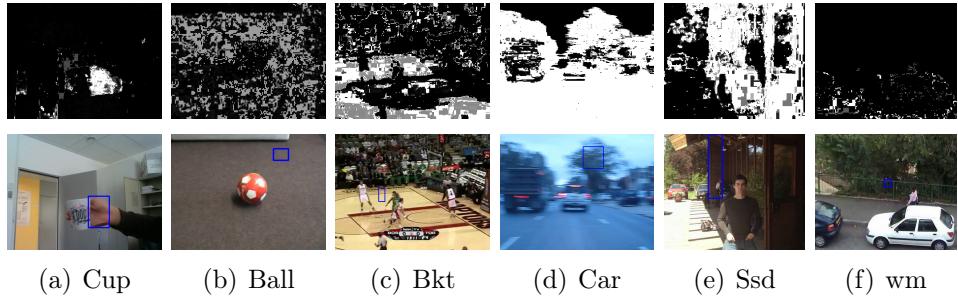


FIGURE 12 – Les résultats d'algorithme de mis à jour de l'histogramme modèle. Bkt : Basket, Ssd : Sunshade, wm : Woman.

Lorsque l'algorithme original suit la cible, il produit une fenêtre de la même taille que la fenêtre originale mais dans une position différente. L'algorithme de mise à jour de l'histogramme du modèle signifie que la fenêtre utilisée pour la frame suivante est la même que la fenêtre générée pour cette frame, y compris l'histogramme de la teinte et la taille de la fenêtre.

En fait, l'algorithme considérant mis à jour de l'histogramme du modèle ne fonctionne pas bien pour les vidéos, car il ne tient compte que de la position de la fenêtre, et non de sa taille et du fait qu'il s'agisse ou non d'une cible à l'intérieur de la fenêtre. S'il n'y a pas de cible dans la fenêtre de la frame actuelle, l'algorithme

génère un histogramme de couleurs incorrect pour l'image suivante. L'objectif suivi par la frame suivante sera modifié. Cette erreur est souvent irréversible.

La Figure 12 montre que l'algorithme de mis à jour de l'histogramme modèle ne fonctionne pas bien dans toutes les vidéos.

Par conséquence, le première algorithme a une meilleure performance que la dernière. Le première peut réussir de régler la difficulté rencontrée dans les vidéo Antoine-Aut.mp4 et VOT-Ball.mp4. Mais, le dernière ne fonctionne pas bien dans toutes les vidéos. Sa performance est pire que l'algorithme originale.

Grâce à des recherches plus approfondies, il existe maintenant d'algorithmes améliorés pour régler les problèmes au-dessus.

1. **Mean Shift Tracking with Self-updating Tracking Window [3]** qui a proposé une algorithme amélioré pourrait sélectionner la taille appropriée de la fenêtre de suivi dans les scénarios non seulement d'échelle croissante mais aussi d'échelle décroissante.
2. **Mean Shift tracking algorithm based on background optimization** qui a proposé une algorithme a utilisé l'histogramme et la re-quantification des jonctions croisées, et a optimisé le fond en diminuant le poids de la fonction de densité de probabilité (PDF) des pixels du fond, ce qui a permis de réduire l'impact du fond sur la région cible [1].
3. L'article [4] La position de poursuite initiale de l'algorithme de poursuite de cible à décalage moyen est déterminée en fonction de l'état de mouvement de la cible à l'aide du filtrage de Kalman, puis la position de la cible dans la zone voisine est recherchée et le modèle de cible est construit, et enfin la position finale est déterminée en fonction de la correspondance du coefficient de similarité, et ainsi de suite de manière itérative pour obtenir une poursuite précise de la cible en mouvement.
4. **Target tracking algorithm based on Kalman filter and optimization MeanShift [5]** qui a aussi proposé une algoritnmes résout les changements de fond et de forme de la cible et a une plus grande précision de suivi, une meilleure propriété en temps réel et est en même temps plus robuste que l'algorithme traditionnel de suivi du décalage moyen.

## 2 Transformée de Hough

### 2.1 Calcule du gradient et du norme

Dans cette section, on calcule l'orientation locale à chaque frame qui est effectue pour transformée de Hough. Figure 13 montre que l'affichage d'image du gradient en x gradient en y et norme du gradient. On defines un seuil sur le module du gradient pour masquer les pixels dont l'orientation n'est pas significative. Et on l'affiche orientations où les pixels masquées apparaissant en rouge. Le code suivant :

Listing 1 – Calcule de la orientation

```



```

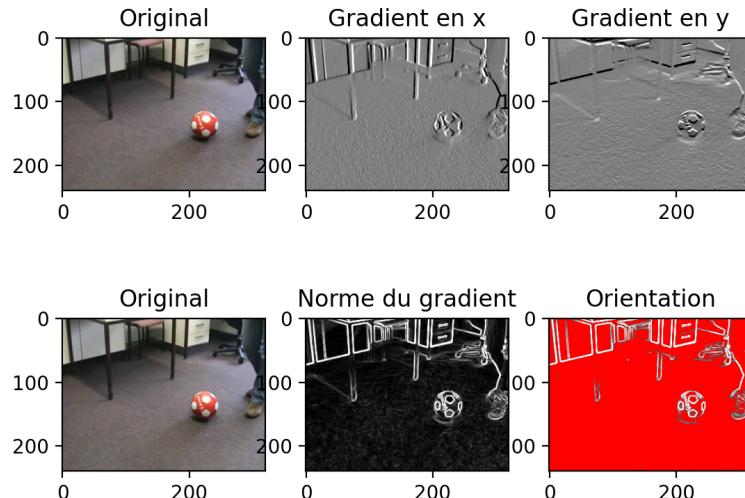


FIGURE 13 – Calcul de l'index de vote (orientation du gradient), avec sélection des pixels votants (Norme du gradient) dans la vidéo VOT-Ball

Séquence	Seuil
Antoine-Mug	30
VOT-Ball	75
VOT-Basket	100
VOT-Car	70
VOT-Sunshade	60
VOT-Woman	150

TABLE 2 – les seuils à chaque séquence

On effectue toutes les vidéos de test pour trouver les seuils à chaque vidéo. Les résultats sont présentes en Table 2.

## 2.2 Suivi de Hough Transformation

Pour réaliser la transformation de hough généralisée, on premièrement génère R-Table puis, on vote dans accumulateur par R-Table pour frame détectée. Le code se réfère à [ce lien](#).

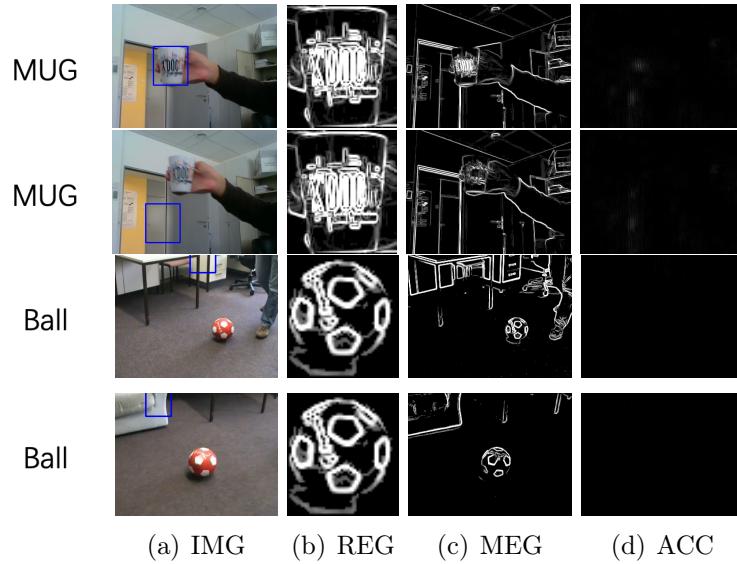


FIGURE 14 – Les résultats d’algorithme de Hough Transformation dans les vidéo Antoine-MUG.mp4 et VOT-Ball.mp4. IMG : frame, REG : edges of template, MEG : edges of matched figure, ACC : accumulator

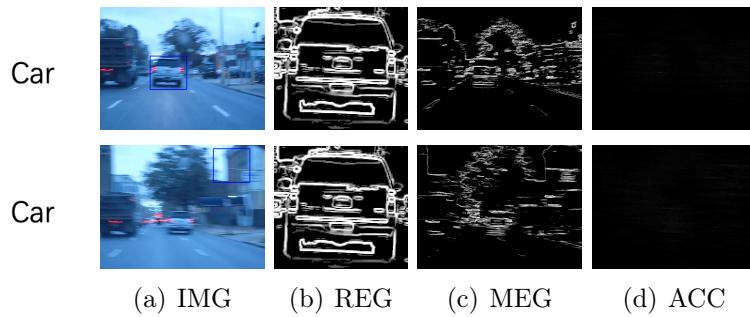


FIGURE 15 – Les résultats d’algorithme de Hough Transformation dans la vidéo VOT-Car.mp4.



FIGURE 16 – Les résultats d’algorithme de Hough Transformation dans les vidéos VOT-Basket.mp4 et VOT-Subshade.mp4.

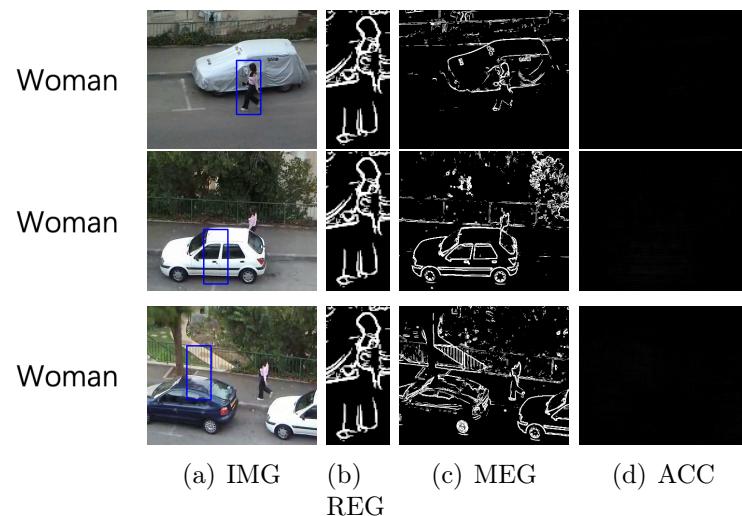


FIGURE 17 – Les résultats d’algorithme de Hough Transformation dans la vidéo VOT-Woman.mp4.

Comme on peut le voir dans les Figures 14, 17, 15, 16, les performances du suivi de cible basé sur la transformation de Hough généralisée ne sont pas très bonnes dans toutes les vidéos testées.

En effet, l'algorithme de la transformation de Hough généralisée mis en œuvre par le code ne prend pas en compte le changement de taille et la rotation de la cible. Donc, théoriquement, la performance de suivi des cibles dont taille change et qui tournent n'est pas très bon. Par exemple, dans la vidéo VOT-Ball.mp4, la cible subit une rotation et un changement de taille. La performance de l'algorithme est alors médiocre.

Les caractéristiques extraites par la transformée de Hough généralisée sont fixes, de sorte que l'algorithme n'est pas robuste lorsque la cible est déformée et lorsqu'elle est occultée. Les vidéos VOT-Basket.mp4, et VOT-Woman.mp4 vérifient cette propriété. Le plan-arrière dans lequel se trouve la cible a également un impact significatif sur les performances de l'algorithme. Par exemple, dans la vidéo VOT-Sunshade.mp4, lorsque la cible est dans l'ombre, les bords de la cible sont difficiles à extraire, ce qui entraîne un échec de la poursuite. Une scène similaire est visible dans la vidéo Antoine-MUG.mp4. Lorsque la tasse (blanche) se déplace devant un mur blanc, elle perd certaines de ses caractéristiques de bord, ce qui entraîne une défaillance du suivi. Dans cette vidéo, il y a également un cas où la taille de la cible change.

Cependant, l'algorithme est un peu robuste dans le cas où les caractéristiques de bord de la cible de poursuite ne changent pas et où la cible ne se déplace pas à grande vitesse. Par exemple, dans la vidéo VOT-Car.mp4, la suivie de la voiture.

## 2.3 MeanShift avec la Transformation de Hough généralisée

### 2.3.1 Analyse de performance

La fusion des deux algorithmes peut être réalisée en passant l'accumulateur obtenu par la transformée de Hough généralisée dans l'algorithme de MeanShift. Référez-vous à [ce lien](#) pour le code mis en œuvre.



FIGURE 18 – Les résultats d'algorithme de MeanShift avec la Transformation de Hough généralisée dans la vidéo VOT-Car.mp4. ACC : accumulator

La fusion de ces deux algorithmes n'améliore pas les performances en cas de

rotation, de déformation et de changement de taille de la cible. Mais il améliore les performances en cas de bougé de l'appareil et lorsque la forme de la cible ne change pas. Dans la vidéo VOT-Car.mp4, l'algorithme a de bonnes performances. Comme le montre la Figure 18. Par rapport à l'algorithme précédent, les performances de l'algorithme sont dans ce cas bien meilleures. L'algorithme est capable de mieux suivre la cible même sous le violent bougé de la caméra.

### 2.3.2 Proposition de stratégie

Dans cette section, une stratégie de mise à jour du modèle qui permette de prendre en compte les déformations de l'objet est proposée. Par plus étude plus profond, un filtre de particules adaptatif basé sur la couleur [2] a trouvé. En fait, le filtrage des particules s'est avéré très efficace pour les problèmes d'estimation non linéaire et non gaussienne. Ce méthode considère l'intégration des distributions de couleurs dans le filtrage particulaire, qui a généralement été utilisé en combinaison avec des caractéristiques d'image basées sur les bords. Les distributions de couleurs sont appliquées pour qu'elles sont robustes à l'occlusion partielle, sont invariantes en termes de rotation et d'échelle et efficaces en termes de calcul.

## Références

- [1] Wei LIU, Yi-ning LUO et Nan SUN. "Mean Shift tracking algorithm based on background optimization [J]". In : *Journal of Computer Applications* 29.4 (2009), p. 1015-1017.
- [2] Katja NUMMIARO, Esther KOLLER-MEIER et Luc VAN GOOL. "An adaptive color-based particle filter". In : *Image and vision computing* 21.1 (2003), p. 99-110.
- [3] HM QIAN, YB MAO et ZQ WANG. "Mean shift tracking with self-updating tracking window". In : *Journal of Image and Graphics* 12.2 (2007), p. 245-249.
- [4] Xiaoliang WANG et Chenglin WEN. "Exploration of Kalman filtering fusion optimized Mean Shift for target tracking". In : *Technology Innovation and Application* 26 (2015), p. 39-39.
- [5] Heng WU, Tao HAN et Jie ZHANG. "Target tracking algorithm based on Kalman filter and optimization MeanShift". In : *LIDAR Imaging Detection and Target Recognition 2017*. T. 10605. International Society for Optics et Photonics. 2017, p. 1060526.