



UNIVERSITÉ DE MONTPELLIER

MASTER 2 - IMAGINE

Vision, Réalité virtuelle et Augmentée

TP/Projet - Poursuite de cible

[HAI935I]

COMBOT Evan

05/11/2024

[Lien GitHub vers les vidéos du projet](#)

1 Préambule

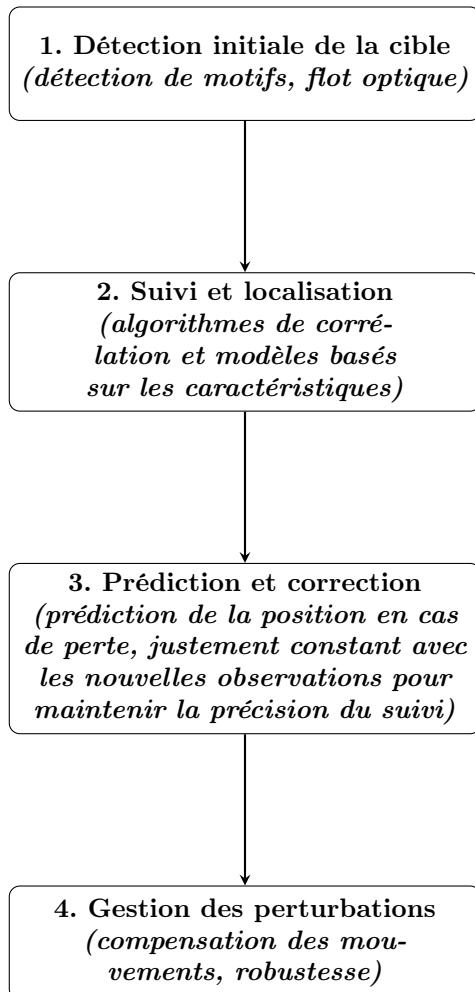
Nous avions le choix entre le TP de vision stéréoscopique et le TP sur la poursuite de cible. J'ai choisi le TP sur la poursuite de cible.

Vous trouverez toutes les vidéos citées ci-dessous dans le lien GitHub ci-dessus.

2 Poursuite de cible

La **poursuite de cible** est une technique consistant à détecter, identifier et suivre en continu un objet spécifique dans une séquence d'images ou une vidéo malgré les changements de contexte et les perturbations (éclairage, orientation, mouvement de la caméra). La poursuite de cible est très efficace pour de nombreuses applications comme la vidéosurveillance, la navigation autonome, les systèmes de sécurité et la réalité augmentée.

2.1 Le processus général



1. **Détection initiale de la cible** : Un objet d'intérêt est d'abord identifié pour être suivi. Pour la *mise en correspondance de motifs*, cela implique de détecter un motif spécifique de l'objet. En ce qui concerne le *flot optique*, des points d'intérêt ou des pixels distinctifs sont sélectionnés pour faciliter le suivi dans la séquence d'images.

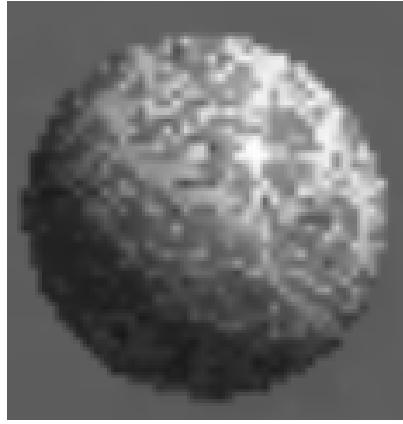


FIGURE 1 – Exemple de motif (1)



FIGURE 2 – Exemple de motif (2)



FIGURE 3 – Exemple de motif (3)



FIGURE 4 – Exemple de motif (4)

2. **Suivi et localisation :** Une fois la cible détectée, sa position est réactualisée à chaque image successive pour maintenir le suivi.
3. **Prédiction et correction :** En cas de perte de la cible, des algorithmes de prédiction estiment sa position probable pour faciliter sa reprise une fois qu'elle est réapparue.
4. **Gestion des perturbations :** Le système gère les variations d'apparence de la cible dues aux changements d'éclairage, rotations ou mouvements rapides. Des techniques de compensation de mouvement et des modèles d'apparence améliorent la robustesse du suivi.

3 Méthodes de mises en correspondance

La mise en correspondance de motifs repose sur une mesure de similarité entre deux images : l'image courante et le motif recherché qui est une sous-image de l'image courante. Ces mesures de similarité peuvent être basées sur des distances statistiques ou sur des coefficients de corrélation.

3.1 La corrélation

La corrélation R des motifs de deux images $I_1(x, y)$ et $I_2(x, y)$ est donnée par :

1. **Corrélation :**

$$R = \frac{1}{N_x N_y} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} I_1(x, y) I_2(x, y)$$

où N_x et N_y représentent les dimensions communes des deux images.

Pour éviter les problèmes de variations lumineuses dus aux ajustements automatiques des capteurs d'images modernes, il existe une autre version de la corrélation, centrée et normalisée appelée **coefficient de corrélation de Pearson**.

2. Coefficient de corrélation de Pearson :

$$\rho = \frac{1}{N_x N_y} \sum_x \sum_y \frac{(I_1(x, y) - \mu_1)(I_2(x, y) - \mu_2)}{\sigma_1 \sigma_2}$$

avec :

- La moyenne de $I_1(x, y)$:

$$\mu_1 = \frac{1}{N_x N_y} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} I_1(x, y)$$

- La moyenne de $I_2(x, y)$:

$$\mu_2 = \frac{1}{N_x N_y} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} I_2(x, y)$$

- L'écart-type de $I_1(x, y)$:

$$\sigma_1 = \sqrt{\frac{1}{N_x N_y} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} (I_1(x, y) - \mu_1)^2}$$

- L'écart-type de $I_2(x, y)$:

$$\sigma_2 = \sqrt{\frac{1}{N_x N_y} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} (I_2(x, y) - \mu_2)^2}$$

3.2 Les distances statistiques

Il est également possible de mesurer une distance entre les deux motif grâce au SAD (somme absolue des distances) et SSD (somme des écarts quadratiques) :

1. SAD (Somme des distances absolues) :

$$SAD = \frac{1}{N_x N_y} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} |I_1(x, y) - I_2(x, y)|$$

2. SSD (Somme des écarts quadratiques) :

$$SSD = \frac{1}{N_x N_y} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} (I_1(x, y) - I_2(x, y))^2$$

3.3 Ce que j'ai fais

Pour pouvoir faire de la poursuite de cible à l'aide de la mise en correspondance de motifs, j'ai suivi ce procédé :

3.3.1 Sélection du motif

On sélectionne une région de l'image (ici un rectangle) qui représente le motif à suivre, extrait à partir de la première image de la séquence d'images ou de la vidéo.



FIGURE 5 – Exemple d'un motif extrait

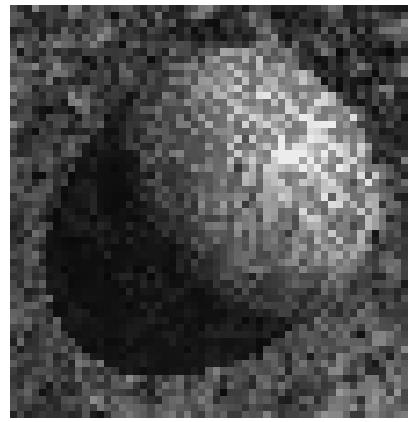


FIGURE 6 – Exemple d'un motif extrait

3.3.2 Extraction du motif

Une fois que le motif est sélectionné, il est extrait et il est gardé en mémoire.

3.3.3 Correspondance de modèles

Ici, j'ai choisi d'utiliser l'algorithme **Template Matching**. Cet algorithme permet de trouver de petites parties d'une image qui correspondent à une image modèle (ici, le motif). Après avoir calculé la corrélation de Pearson pour toutes les positions, l'objectif de l'algorithme est d'identifier la position (x^*, y^*) où la valeur de la corrélation est maximale.

$$(x^*, y^*) = \arg \max_{(x,y)} C(x, y)$$

3.3.4 Détermination de la meilleure correspondance

Après avoir calculé la matrice de résultats à l'aide du **Template Matching**, la prochaine étape consiste à identifier les valeurs minimales et maximales dans cette matrice ainsi que leurs positions respectives. Cela permet de déterminer où le motif correspond le mieux à l'image d'entrée.

$$result(i, j) < minVal \rightarrow minVal = result(i, j), \quad minLoc = (i, j)$$

$$result(i, j) > maxVal \rightarrow maxVal = result(i, j), \quad maxLoc = (i, j)$$

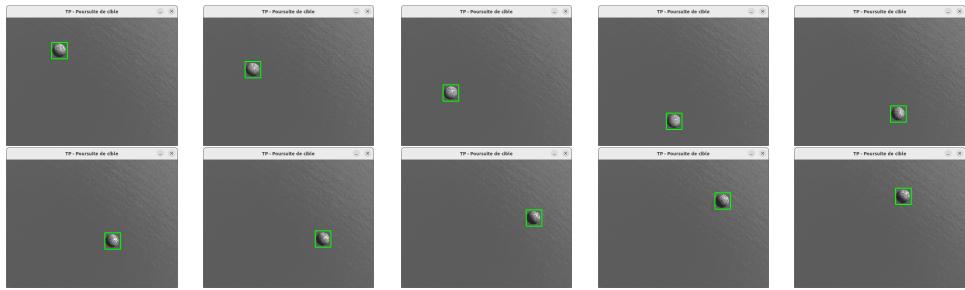
Ensuite, on introduit un seuil qui permet de choisir les correspondances qui sont acceptables ou non. La valeur de base de mon seuil est de 0.7.

3.3.5 Résultats

Concernant mes résultats, j'ai mis quelques images représentant la mise en correspondance de motif dans une séquence d'images (image par image) ou une vidéo.

Après la sélection du motif, un rectangle vert est affiché. Ce rectangle est de la même dimension que le motif sélectionné et est dessiné autour de la meilleure correspondance.

Séquence d'images (sans variation)



On peut remarquer que la poursuite de cible est correcte. Tout au long de la séquence, le rectangle suit correctement le motif.

Voir la vidéo : [SequencCorrSansVar](#)

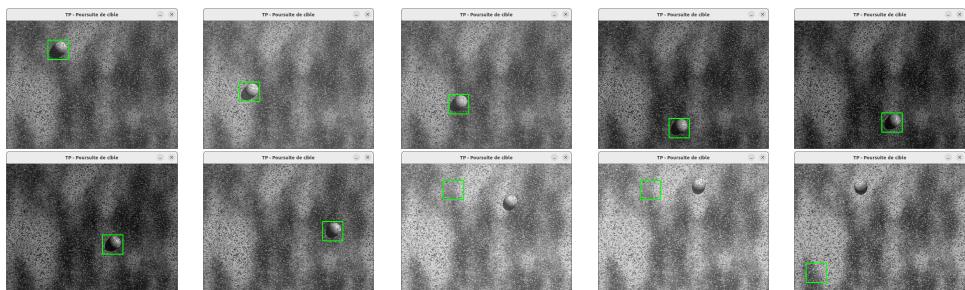
Séquence d'images (avec des variations)



Comme pour la séquence sans variation, ici, le rectangle vert suit correctement le motif. La poursuite de cible est donc correcte.

Voir la vidéo : [SequencCorrAvecVar](#)

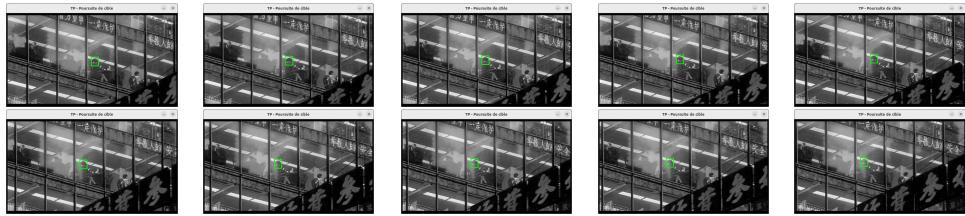
Séquence d'images (avec des variations et du bruit)



On observe qu'à un certain moment, la correspondance n'est plus correcte. Cela s'explique par le fait que la corrélation entre le motif et l'image devient incohérente. En effet, j'ai volontairement sélectionné le motif avec son ombre mais à un moment donné, la séquence devient de plus en plus lumineuse et le motif ne présente plus d'ombre. Ce changement entraîne une erreur de corrélation et donc le rectangle apparaît au mauvais endroit de l'image.

Voir la vidéo : [SequencCorrAvecFleurs](#)

Séquence d'images (Ghost2)



On peut remarquer que le rectangle suit correctement le motif. J'ai choisi un motif assez petit ce qui permet de minimiser les erreurs liées à la corrélation.

Voir la vidéo : [SequenceCorrGhost2](#)

Vidéos



Par exemple si l'on prend comme référence la vidéo [VideoCorrVideo1](#), le motif représenté par la voiture bleue est bien détecté tant qu'il reste visible dans la vidéo. Cependant, dès que le motif disparaît du champ de vision, la correspondance n'est plus trouvée et le programme tente, tant bien que mal de retrouver la meilleure correspondance possible. Dès fois, c'est un fond de la voiture (avec le conducteur) qui est sélectionné et des fois ce sont des voitures (avec teinte plus ou moins identique) qui sont sélectionnés.

Voir les vidéos : [VideoCorrVideo1](#) et [VideoCorrVideo2](#)

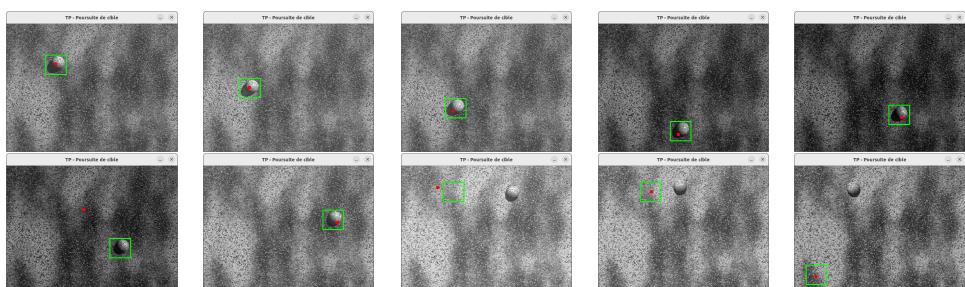
4 D'autres tests

4.1 Avec le filtre de Kalman

Le filtre de Kalman peut être utilisé pour améliorer la mise en correspondance de motifs. Il permet de prédire la position future d'un objet, ce qui peut être utilisé pour affiner la recherche de motifs dans des images successives.

Le point rouge représente la position prédictive par le filtre de Kalman. Le filtre de Kalman estime où la cible devrait se trouver en fonction de sa position et de sa vitesse estimées dans les images précédentes. Cette estimation permet de prédire l'endroit où la cible pourrait être même si elle est temporairement ou partiellement caché.

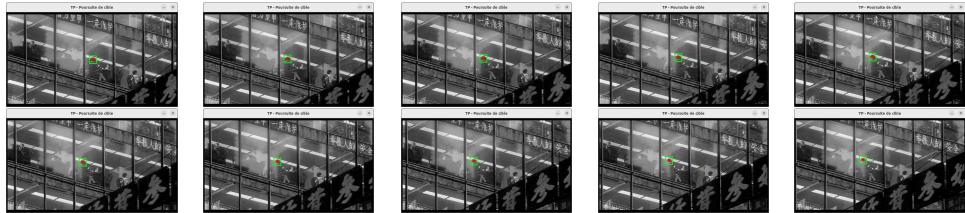
Séquence d'images (avec des variations et du bruit)



On peut remarquer que la position prédictive par le filtre de Kalman est cohérente avec la correspondance trouvée.

Voir la vidéo : [SequenceCorrAvecFleurs_Kalman](#)

Séquence d'images (Ghost2)



Ici aussi, les positions prédictes au fil des images de la séquence par le filtre de Kalman sont correctes.

Voir la vidéo : [SequenceCorrGhost2_Kalman](#)

4.2 Évaluation de la qualité de la correspondance

Pour pouvoir visualiser et évaluer la qualité de la correspondance, j'ai modifié le seuil de qualité. Pour chaque séquence d'images, j'ai utilisé le motif que pour les mêmes tests ci-dessus.

4.2.1 Séquence d'images (sans variation)

Seuil = 0.2

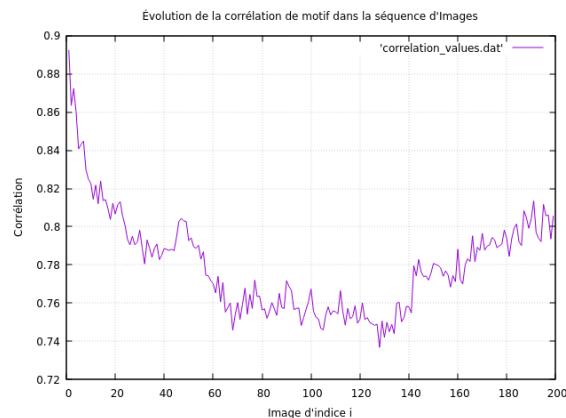


FIGURE 7 – Évolution de la corrélation de motif dans la séquence d'images

Seuil = 0.5

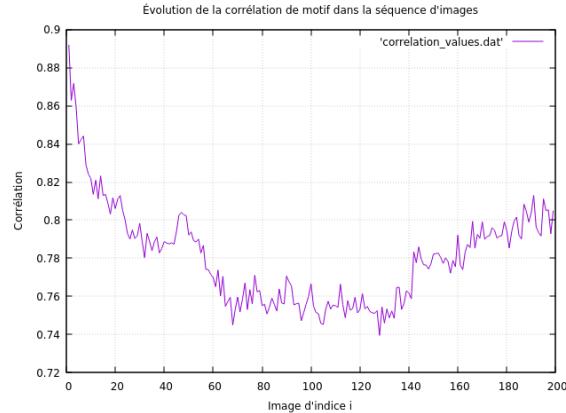


FIGURE 8 – Évolution de la corrélation de motif dans la séquence d'images

Seuil = 0.8

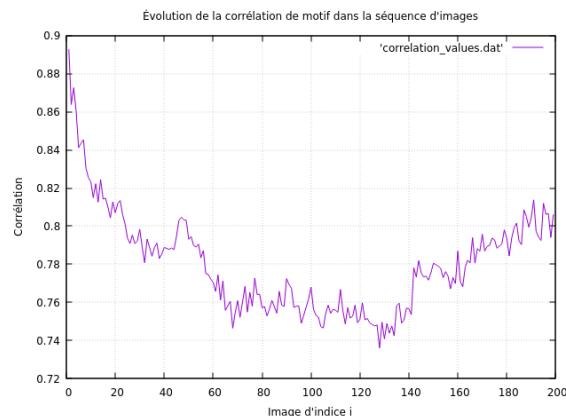


FIGURE 9 – Évolution de la corrélation de motif dans la séquence d'images

On peut remarquer ici que le changement de seuil n'a pas d'effet majeur sur l'évolution de la correspondance. On observe un pic dès les premières images, ce qui est normal, car la séquence est encore sombre au début. Un creux apparaît au milieu de la séquence, ce qui correspond à des images plus claires. La corrélation est alors plus faible, car le motif est initialement entouré de pixels sombres. On observe ensuite une hausse vers la fin de la séquence, car celle-ci redevient sombre.

4.2.2 Séquence d'images (avec des variations et du bruit)

Seuil = 0.2

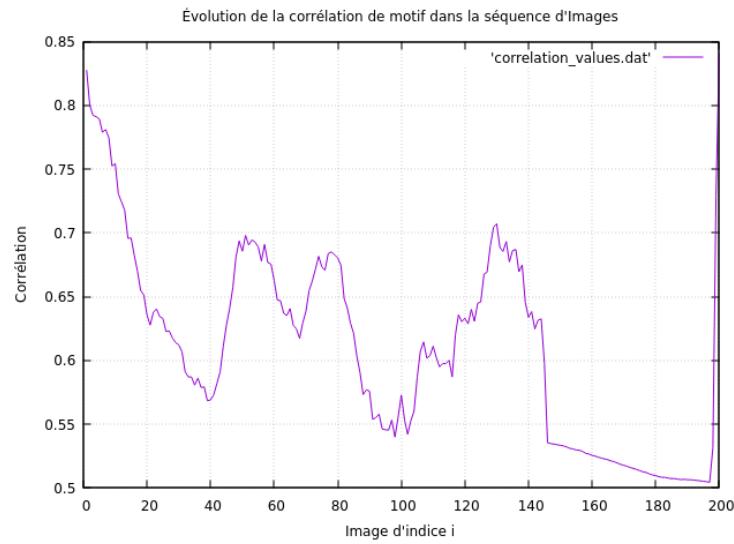


FIGURE 10 – Évolution de la corrélation de motif dans la séquence d'images

Seuil = 0.5

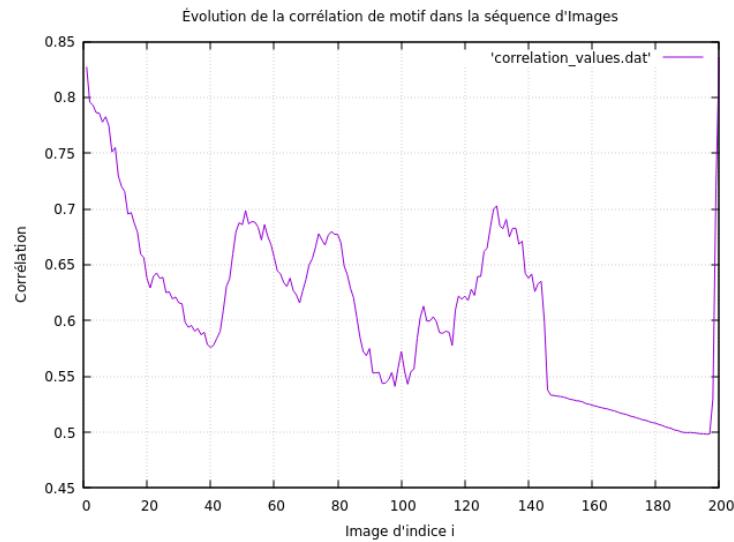


FIGURE 11 – Évolution de la corrélation de motif dans la séquence d'images

Seuil = 0.8

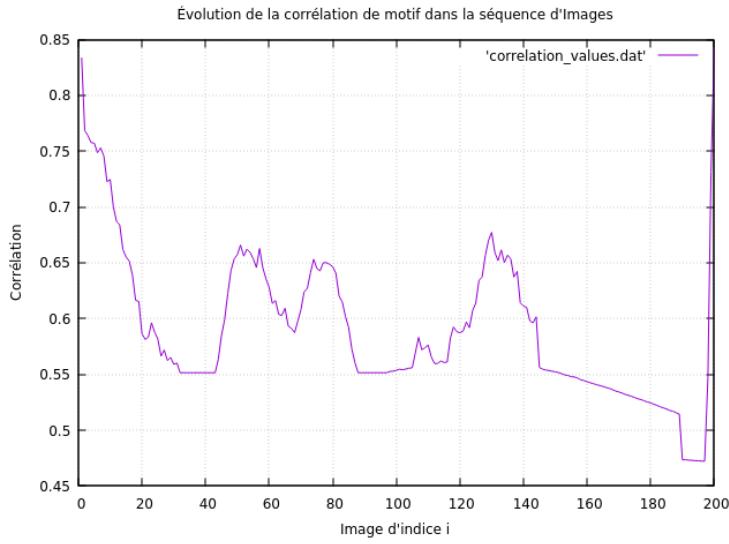


FIGURE 12 – Évolution de la corrélation de motif dans la séquence d'images

On remarque que plus la valeur du seuil augmente, plus la corrélation est "lissée" entre les grandes variations de valeurs (augmentations et diminutions). On observe de nombreux changements, quelle que soit la valeur du seuil, ce qui indique que le motif a du mal à être détecté. Comme sur la vidéo, on peut remarquer dans les toutes dernières images, la valeur de la corrélation augmente énormément ce qui correspond au fait que le motif a été retrouvé et qu'il est arrivé à un emplacement qui ressemble à son emplacement d'origine.

4.2.3 Séquence d'images (Ghost2)

Seuil = 0.2

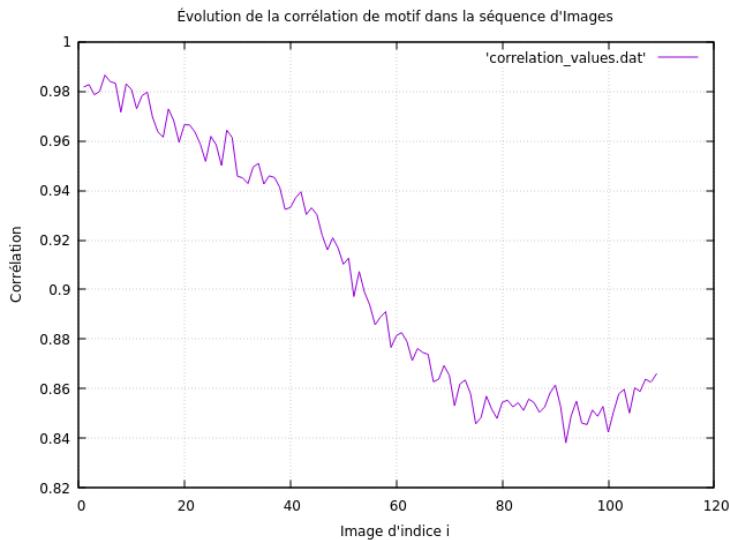


FIGURE 13 – Évolution de la corrélation de motif dans la séquence d'images

Seuil = 0.5

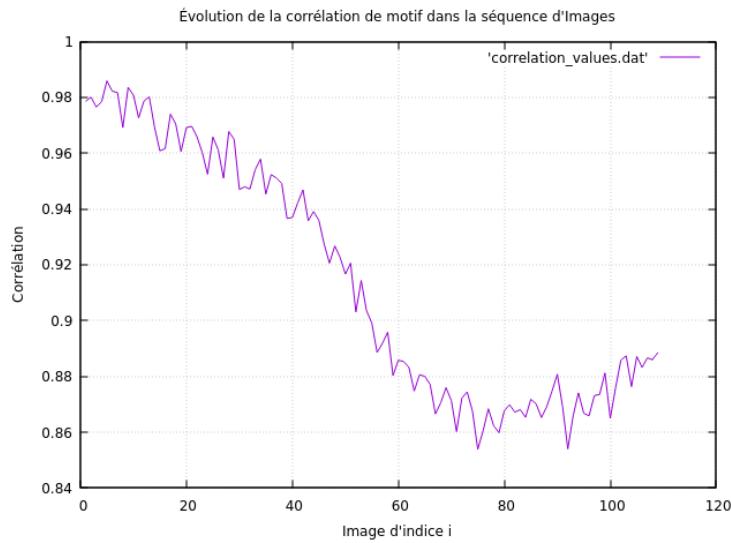


FIGURE 14 – Évolution de la corrélation de motif dans la séquence d’images

Seuil = 0.8

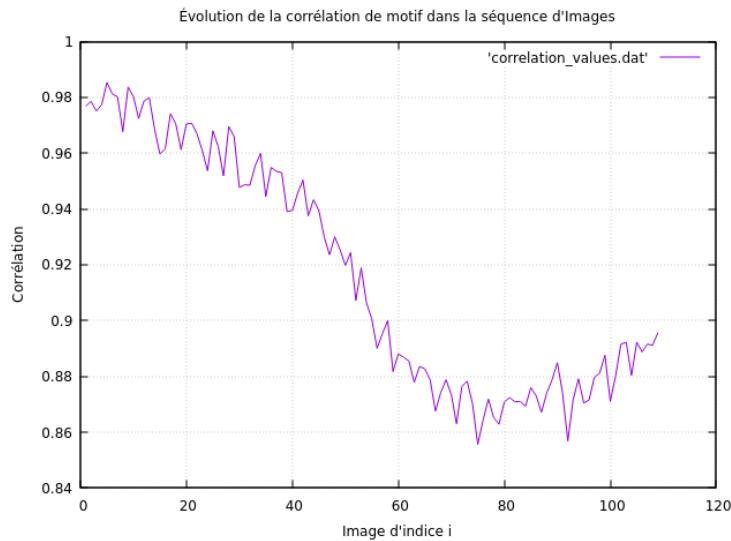


FIGURE 15 – Évolution de la corrélation de motif dans la séquence d’images

On peut remarquer que le changement de la valeur du seuil n'a presque aucun impact sur l'évolution de la corrélation car les graphiques sont quasiment identiques. Dans la séquence d'images, on observe qu'une sorte de brume apparaît progressivement sur le motif. On distingue très clairement le moment où la brume s'installe car on constate une diminution progressive de la corrélation sur les courbes.

4.3 Corrélation croisée

Pour expérimenter la mise en correspondance de motif, je me suis intéressé à la corrélation croisée.

J'ai gardé l'algorithme **Template Matching** mais au lieu d'utiliser la corrélation de Pearson, j'utilise la corrélation croisée définie par :

$$R(x, y) = \frac{\sum_{i,j} (T(i, j) - \bar{T}) \cdot (I(x + i, y + j) - \bar{I}_{(x,y)})}{\sqrt{\sum_{i,j} (T(i, j) - \bar{T})^2 \cdot \sum_{i,j} (I(x + i, y + j) - \bar{I}_{(x,y)})^2}}$$

- $R(x, y)$ représente le résultat de la corrélation en position (x, y) .
- \bar{T} et $\bar{I}_{(x,y)}$ sont les moyennes des pixels dans T et dans la sous-région correspondante de I , respectivement.

4.3.1 Séquence d'images (sans variation)

Seuil = 0.2

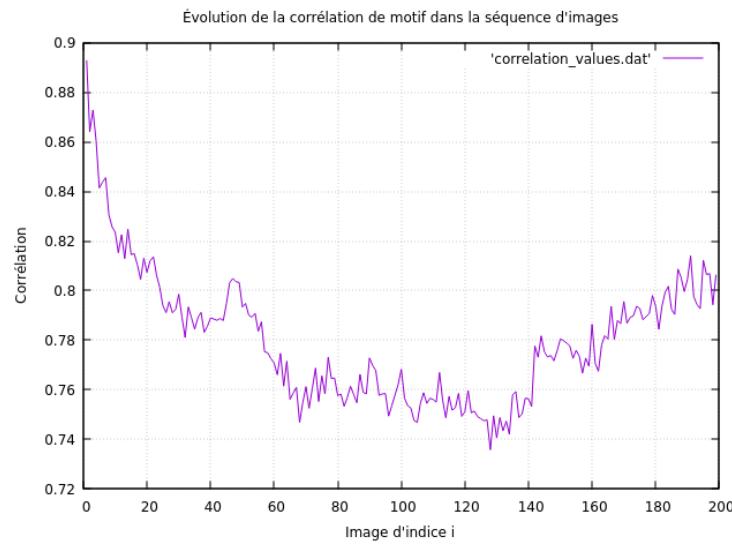


FIGURE 16 – Évolution de la corrélation croisée de motif dans la séquence d'images

Seuil = 0.5

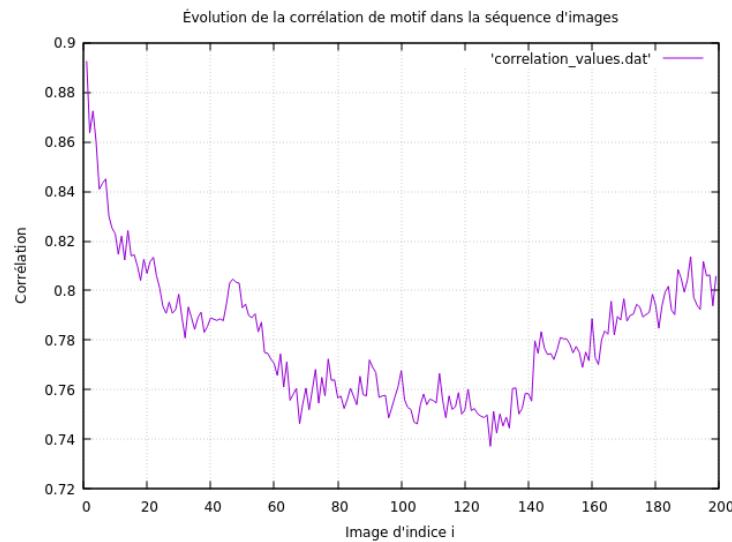


FIGURE 17 – Évolution de la corrélation croisée de motif dans la séquence d'images

Seuil = 0.8

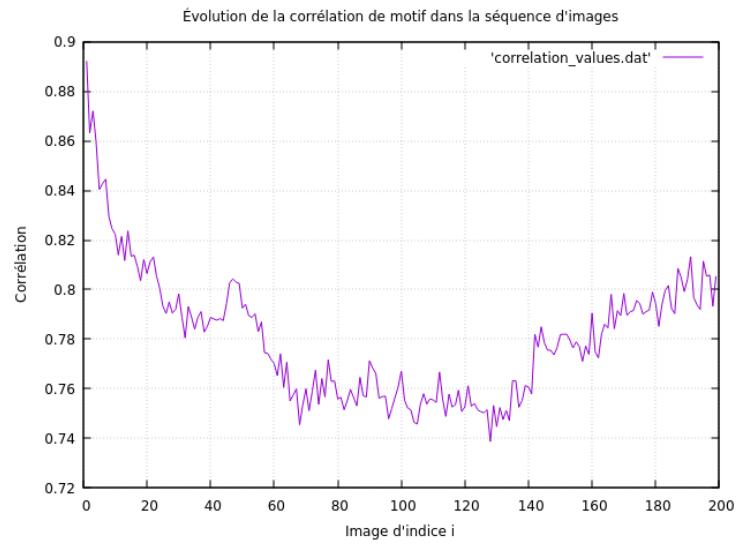


FIGURE 18 – Évolution de la corrélation croisée de motif dans la séquence d'images

On peut remarquer que les courbes avec les différents valeurs de seuils sont similaires. Ces courbes ne sont pas très différentes de celles obtenues avec la corrélation de Pearson.

4.3.2 Séquence d'images (avec des variations et du bruit)

Seuil = 0.2

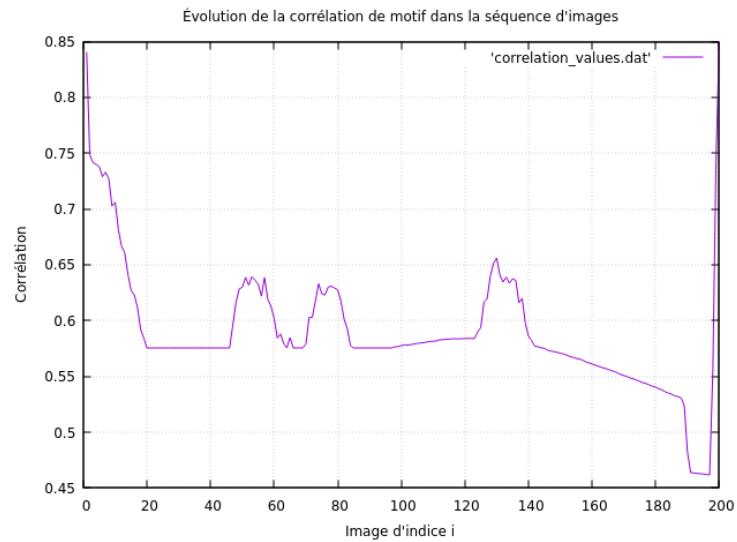


FIGURE 19 – Évolution de la corrélation croisée de motif dans la séquence d'images

Seuil = 0.5

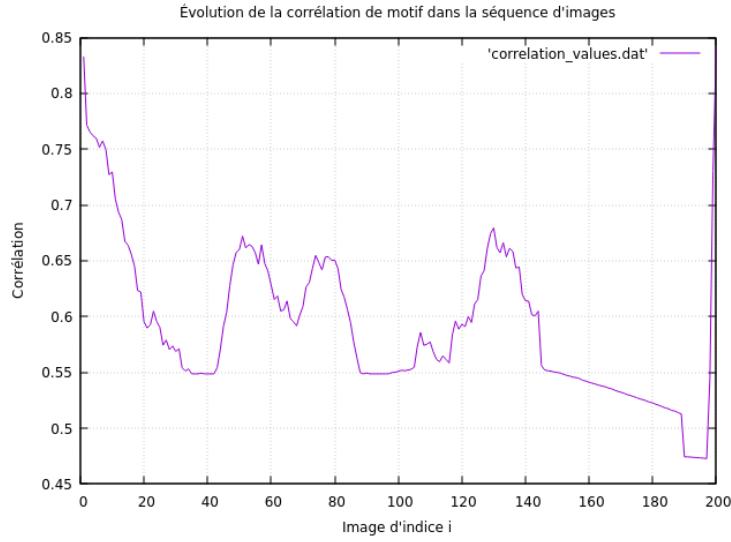


FIGURE 20 – Évolution de la corrélation croisée de motif dans la séquence d'images

Seuil = 0.8

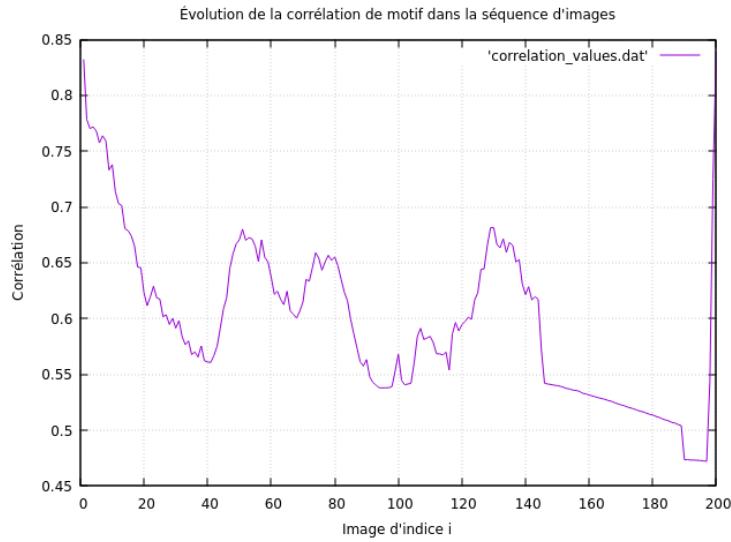


FIGURE 21 – Évolution de la corrélation croisée de motif dans la séquence d'images

On peut remarquer que plus la valeur du seuil augmente et plus il y a de variations. On peut remarquer que les valeurs obtenues avec la corrélation croisée sont supérieures à celles obtenues avec la corrélation de Pearson. Cela se voit tout particulièrement lorsque la valeur du seuil est bas car les valeurs sont à la fois hautes et stables.

4.3.3 Séquence d'images (Ghost2)

Seuil = 0.2

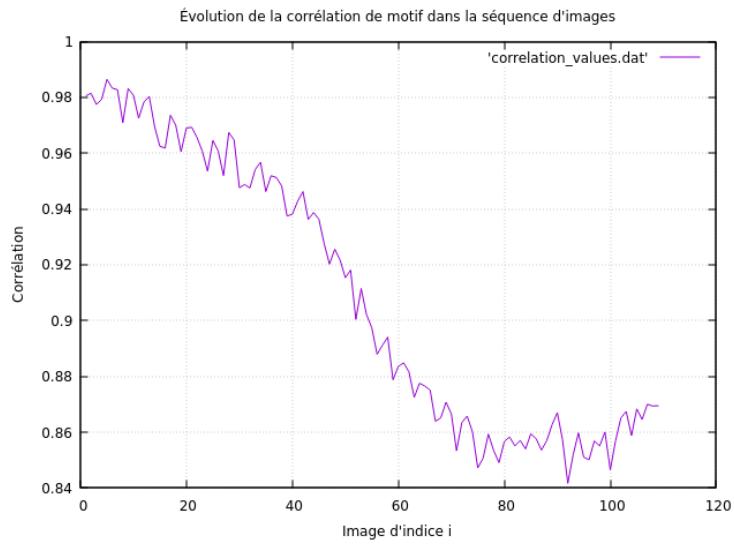


FIGURE 22 – Évolution de la corrélation croisée de motif dans la séquence d'images

Seuil = 0.5

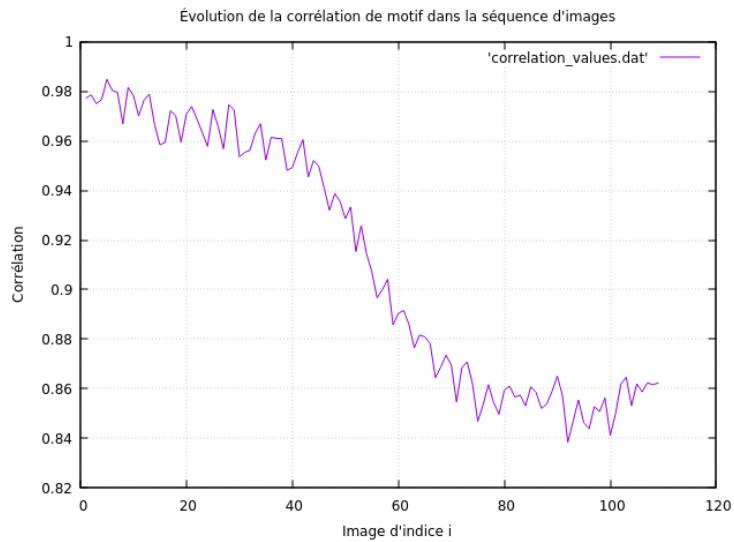


FIGURE 23 – Évolution de la corrélation croisée de motif dans la séquence d'images

Seuil = 0.8

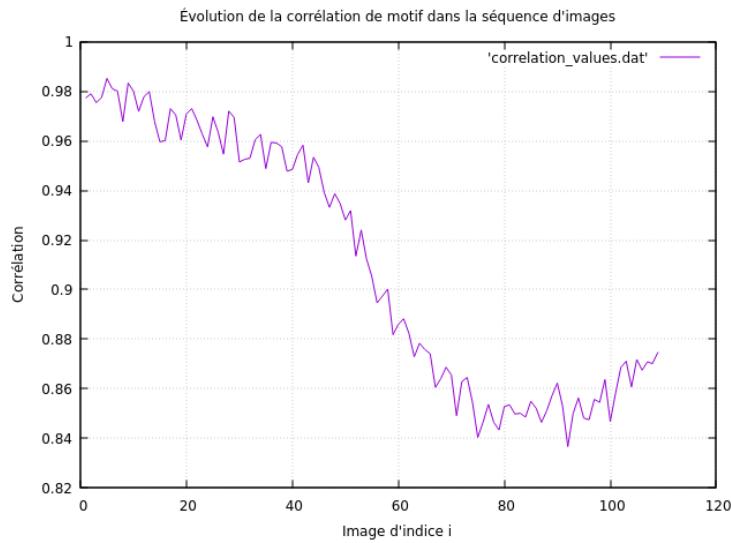


FIGURE 24 – Évolution de la corrélation croisée de motif dans la séquence d’images

On peut remarquer que les courbes sont quasiment similaires entre elles et entre celles obtenues avec la corrélation de Pearson.

4.4 Conclusion

Je trouve que l’utilisation de la corrélation croisée pour des séquences d’images / vidéos avec de fortes variations et l’idéal car cela permet d’obtenir un meilleur suivi d’une cible.

5 Le flot optique

La méthode du flot optique consiste à supposer que le motif est une fonction discrète en 2D produite par la discréttisation (par le capteur d’images) de l’illumination locale.

Soit $I(x, y, t)$ la répartition de l’illumination dans une fenêtre courante. Si le motif est correctement recalé alors $I(x, y, t)$ représente également la fonction d’illumination du motif. À l’instant suivant, si le motif se déplace de (dx, dy) sur une période dt , l’illumination demeure inchangée ce qui se traduit par l’équation :

$$I(x + dx, y + dy, t + dt) = I(x, y, t).$$

Tout d’abord, on effectue une expansion au premier ordre de l’équation :

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + O^2$$

Une expansion du premier ordre fait référence à la dérivation de l’équation du flot optique basée sur la conservation de l’intensité lumineuse entre deux images consécutives avec l’idée que l’intensité lumineuse ne change pas d’une image à l’autre.

On utilise ensuite un opérateur appelé **extracteur de gradient** qui permet d’obtenir à partir d’une image $I(x, y, t)$ les valeurs estimées des gradients dans les deux dimensions pour chaque point de l’image. L’extracteur de gradient calcule les quantités $I_x(x, y)$ et $I_y(x, y)$ représentant respectivement les gradients de l’intensité de l’image le long de x et y :

$$I_x(x, y) = \frac{\partial I(x, y, t)}{\partial x}, \quad I_y(x, y) = \frac{\partial I(x, y, t)}{\partial y}$$

Le système d'équations à partir de l'équation du flot optique est créé :

$$A(u, v) = - \begin{pmatrix} I_x & I_y & I_t \\ I_x & I_y & I_t \\ I_y & I_y & I_t^2 \end{pmatrix} \iff A = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

Le système d'équations du flot optique formulé comme suit :

$$\begin{aligned} A \begin{pmatrix} u \\ v \end{pmatrix} &= B \\ A \begin{pmatrix} u \\ v \end{pmatrix} &= - \begin{pmatrix} I_x I_t \\ I_y I_t \end{pmatrix} \end{aligned}$$

où

$$A = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

Pour résoudre un système surdéterminé (où le nombre d'équations est supérieur au nombre d'inconnues), la méthode des moindres carrés est appliquée. Supposons que nous ayons un système linéaire de la forme suivante :

$$A \begin{pmatrix} u \\ v \end{pmatrix} = \mathbf{b}$$

où A est une matrice des gradients spatiaux, typiquement construite à partir de I_x et I_y (les dérivées de l'image en x et y), et \mathbf{b} est une matrice construite à partir de $-I_t$ (la dérivée temporelle).

La solution du système en moindres carrés, qui minimise l'erreur au sens des moindres carrés, est donnée par :

$$\begin{pmatrix} u \\ v \end{pmatrix} = (A^T A)^{-1} A^T \mathbf{b}$$

Dans le cas où A est surdéterminée, la solution optimale peut être trouvée par la décomposition en valeurs singulières (SVD) :

$$A = U \Sigma V^T$$

où U et V sont des matrices orthogonales, et Σ est une matrice diagonale contenant les valeurs singulières de A .

5.1 Ce que j'ai fais

5.1.1 Détection de coins de Harris

Tout d'abord, je détecte les **coins de Harris**. Un coin est défini comme une zone où l'intensité change fortement dans deux directions orthogonales. Contrairement à un bord où la variation est forte dans une seule direction, un coin présente des changements significatifs dans toutes les directions. Pour chaque pixel de l'image, l'algorithme de Harris calcule la **matrice de structure du gradient** dans une petite fenêtre autour de ce pixel. Cette matrice M est définie par :

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

- $\sum I_x^2$: somme des carrés des dérivées horizontales, qui représente la variation d'intensité en direction x .
- $\sum I_y^2$: somme des carrés des dérivées verticales, qui représente la variation d'intensité en direction y .
- $\sum I_x I_y$: produit des dérivées dans les deux directions, représentant la corrélation entre les changements en x et y .

La réponse de Harris R , qui estime la présence d'un coin en fonction des valeurs de λ_1 et λ_2 est calculé :

$$R = \det(M) - k \cdot (\text{trace}(M))^2$$

où :

- $\det(M) = \lambda_1 \cdot \lambda_2$ est le déterminant de la matrice M .
- $\text{trace}(M) = \lambda_1 + \lambda_2$ est la trace de la matrice M (la somme des valeurs propres).
- k est une constante empirique (souvent fixée entre 0.04 et 0.06), le mien est fixé à 0.04.



FIGURE 25 – Exemple de coins de Harris

5.1.2 Sélection de coins de Harris

Une fois les coins de Harris détectés, une boucle parcourt chaque pixel de l'image contenant les réponses de Harris normalisées. J'applique un seuil (128 dans mon cas) et seuls les pixels ayant une réponse supérieure à ce seuil sont conservés comme des coins intéressants.

5.1.3 Calcul du flot optique

Pour chaque paire d'images consécutives, j'utilise l'algorithme de Lucas-Kanade. Cet algorithme suppose que le mouvement des pixels est constant dans une petite fenêtre (généralement de 3×3 ou 5×5 pixels) (du 3×3 pour moi) autour de chaque point d'intérêt. La méthode des moindres carrés est ensuite appliquée pour résoudre le système d'équations formé par l'algorithme de Lucas-Kanade pour chaque pixel.

5.1.4 Résultats

Séquence d'images (sans variation)



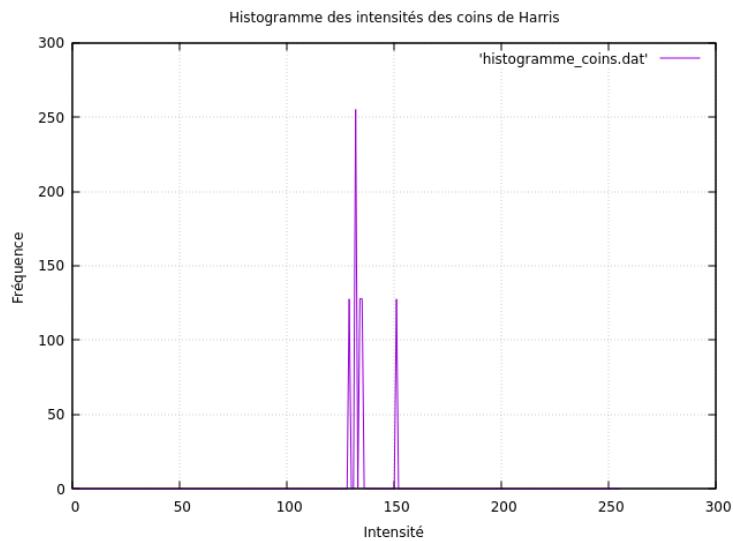


FIGURE 26 – Histogramme des intensités des coins de Harris

On peut remarquer que les points de Harris détectés sont cohérents avec le déplacement du motif.

Voir la vidéo : [SequenceFLOTsansVar](#)

Séquence d'images (avec des variations)

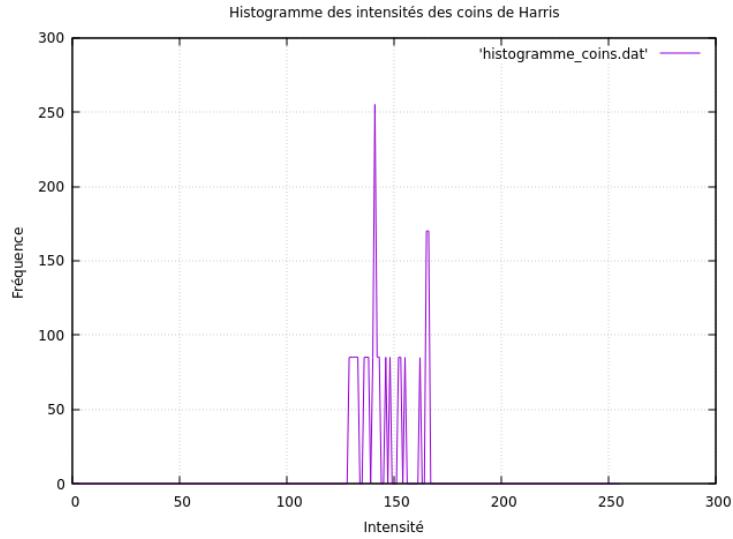
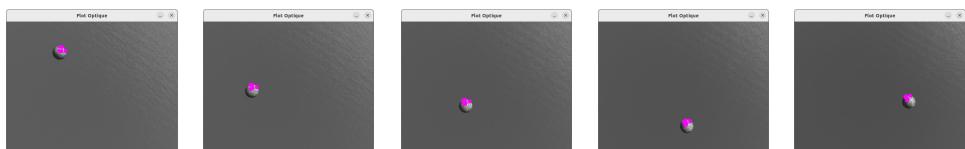


FIGURE 27 – Histogramme des intensités des coins de Harris

Ici aussi, les points de Harris détectés sont cohérents avec le déplacement du motif. Les points de Harris ont une intensité répartie autour de 140.

Voir la vidéo : [SequenceFLOTavecVar](#)

Séquence d'images (avec des variations et du bruit)

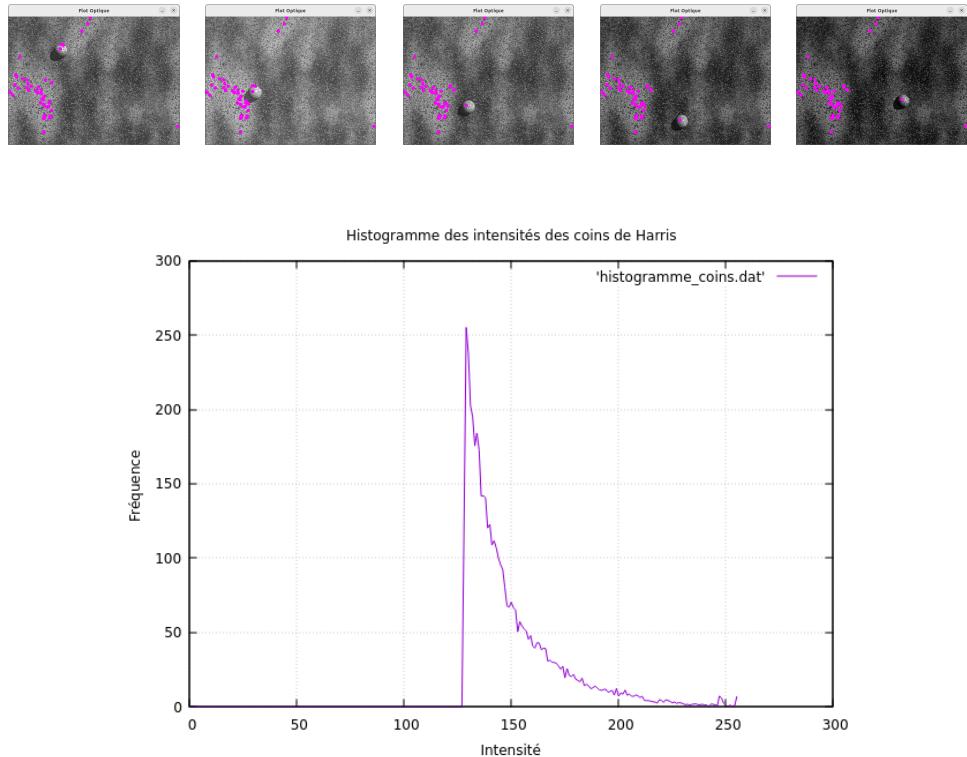


FIGURE 28 – Histogramme des intensités des coins de Harris

Le bruit dans la séquence d'images entraîne l'apparition de différents coins de Harris, y compris celui associé au motif cible. Au fil de la séquence, le coin de Harris lié au motif est détecté correctement. On observe un pic de fréquence autour d'une intensité de 125, puis une diminution progressive jusqu'à une intensité de 250.

Voir la vidéo : [SequenzeFLOT](#)

Vidéo



Les points de Harris sont cohérents et sont correctement détectés au fil des images de la séquence.

Voir la vidéo : [VideoFLOT](#)

5.2 D'autres méthodes

Je me suis également intéressé aux méthodes de TV-L1 et notamment la méthode de Horn et Schunk mais malheureusement, cette dernière ne me donne pas de bons résultats.

6 Conclusion

Pour conclure, les 2 méthodes sont efficaces dans le domaine de la poursuite de cible. Personnellement, ma méthode préférée est la méthode de mise en correspondance de motif car elle permet de chercher un motif en particulier. On pourrait étendre la méthode pour chercher plusieurs motifs en même temps ou rechercher des motifs similaires au motif original.

J'ai vraiment apprécié ce TP car la vision par ordinateur est un domaine que je trouve intéressant.