

# Analyse d'image : Détection de mouvement

SINGARIN-SOLE — BELGUIDOUM

Juin 2024

## Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Introduction</b>                        | <b>2</b> |
| <b>2</b> | <b>Méthode de Horn et Schunck</b>          | <b>2</b> |
| 2.1      | Principe . . . . .                         | 2        |
| 2.2      | Influence du paramètre $\lambda$ . . . . . | 3        |
| 2.2.1    | Petit $\lambda$ . . . . .                  | 4        |
| 2.2.2    | Grand $\lambda$ . . . . .                  | 4        |
| <b>3</b> | <b>Méthode de Lucas et Kanade</b>          | <b>5</b> |
| 3.1      | Principe . . . . .                         | 5        |
| 3.2      | Influence des paramètres . . . . .         | 7        |
| 3.2.1    | Poids . . . . .                            | 7        |
| 3.2.2    | Nombre de voisins . . . . .                | 7        |
| <b>4</b> | <b>Application à une séquence d'images</b> | <b>7</b> |
| <b>5</b> | <b>Choix lors de l'implémentation</b>      | <b>8</b> |
| 5.1      | Méthode 1 . . . . .                        | 8        |
| 5.2      | Méthode 2 . . . . .                        | 8        |
| <b>6</b> | <b>Conclusion</b>                          | <b>8</b> |

# 1 Introduction

On se propose dans ce TP d'effectuer l'implémentation et l'analyse de deux méthodes bien connues pour la détection du flux optique : la méthode de Horn et Schunck et la méthode de Lucas et Kanade. Ces techniques permettent de déterminer le mouvement entre plusieurs images successives d'une séquence vidéo, en estimant le déplacement des pixels d'une image à l'autre. Pour les 2 méthodes on fera les hypothèses suivantes :

- **Invariance de l'intensité** : L'invariance de l'intensité suppose que l'intensité d'un pixel ne change pas au cours du temps lorsqu'il se déplace.

$$\nabla_x I \cdot u + \nabla_y I \cdot v + \nabla_t I = 0$$

- **Cohérence spatiale** : La cohérence spatiale repose sur l'hypothèse que le champ de vitesse varie lentement localement. En d'autres termes, les vecteurs de déplacement  $(u, v)$  sont supposés être lisses dans une petite région de l'image. Mathématiquement, cela se traduit par :

$$\|\nabla u\| \approx 0 \quad \text{et} \quad \|\nabla v\| \approx 0$$

## 2 Méthode de Horn et Schunck

### 2.1 Principe

Considérons un pixel  $I(x, y, t)$  dans la première image (Remarquez qu'une nouvelle dimension, le temps, est ajoutée ici. Auparavant, nous ne travaillions qu'avec des images, donc il n'y avait pas besoin du temps). Ce pixel se déplace de la distance  $(dx, dy)$  dans la frame suivante prise après un temps  $dt$ . Puisque ces pixels sont les mêmes et que l'intensité ne change pas, nous pouvons dire :

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

Ensuite, prenons l'approximation par série de Taylor du côté droit, supprimons les termes communs et divisons par  $dt$  pour obtenir l'équation du flux optique :

$$f_x u + f_y v + f_t = 0$$

La méthode de Horn et Schunck repose sur la recherche du flux  $(\hat{u}, \hat{v})$  qui est solution du problème d'optimisation :

$$(\hat{u}, \hat{v}) \in_{u,v} \left\{ \underbrace{\|\nabla_x I \cdot u + \nabla_y I \cdot v + \nabla_t I\|_2^2}_{(H1)} + \lambda \underbrace{(\|\nabla u\|_2^2 + \|\nabla v\|_2^2)}_{(H2)} \right\}$$

avec  $\lambda > 0$  un paramètre permettant de régler l'influence de la cohérence spatiale.

L'algorithme de cette méthode consiste à calculer  $I_x$ ,  $I_y$  et  $I_t$ , gradients de  $I_1$  et  $I_2$ . Puis nous appliquons une descente de gradient afin de faire converger  $u$  et  $v$  définis de cette manière :

$$u = \bar{u} - \frac{I_x (I_x \bar{u} + I_y \bar{v} + I_t)}{\lambda^2 + I_x^2 + I_y^2}$$
$$v = \bar{v} - \frac{I_y (I_x \bar{u} + I_y \bar{v} + I_t)}{\lambda^2 + I_x^2 + I_y^2}$$

Dans un premier temps, on va s'intéresser à deux images. Ces dernières correspondent à humain fermant le coffre de sa voiture.

## 2.2 Influence du paramètre $\lambda$

Maintenant, on va étudier l'influence du paramètre  $\lambda$ , lorsque l'on fait converger l'énergie ( $\epsilon = 0.1$ ) :

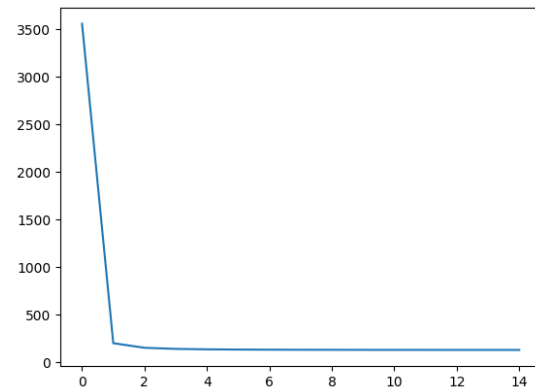


Figure 1:  $\lambda = 0.1$

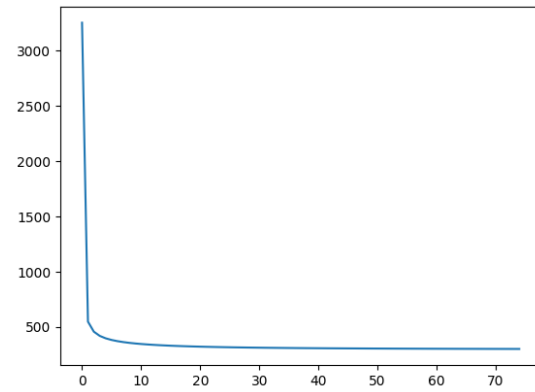


Figure 2:  $\lambda = 1$

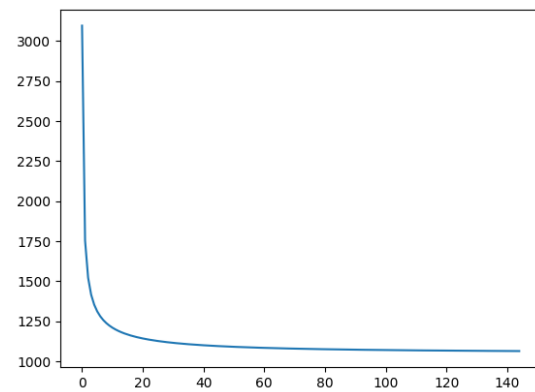


Figure 3:  $\lambda = 10$

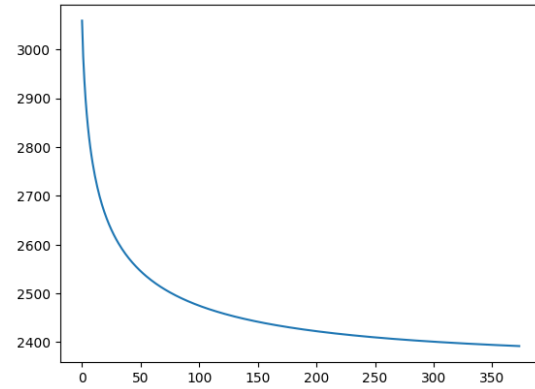


Figure 4:  $\lambda = 100$

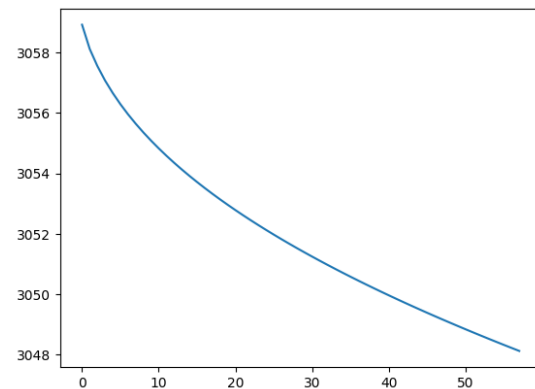


Figure 5:  $\lambda = 1000$

Dans notre fonction de cout nous avons ce terme :

$$(\nabla_x I \cdot u + \nabla_y I \cdot v + \nabla_t I)^2$$

Ce terme impose que le déplacement  $(u, v)$  conserve l'intensité du pixel. Il s'agit d'une contrainte de fidélité aux données. Si ce terme domine, la solution tend à suivre les gradients des images de manière stricte, capturant ainsi plus de détails locaux, mais aussi potentiellement plus de bruit.

L'autre membre est le terme de régularisation

$$\lambda(\|\nabla u\|^2 + \|\nabla v\|^2)$$

Ce terme impose que le champ de vitesse  $(u, v)$  soit lisse, en minimisant les variations spatiales de  $u$  et  $v$ . Si ce terme domine, la solution sera plus lisse et régulière, mais pourrait ne pas capturer les détails fins des mouvements locaux. Influence de  $\lambda$ :

### 2.2.1 Petit $\lambda$

- La minimisation est dominée par le terme de cohérence de l'intensité. Donc on est censé avoir un résultat très cohérent
- Mais la solution peut devenir bruitée ce qui affecte le résultat.

### 2.2.2 Grand $\lambda$

- La minimisation est dominée par le terme de régularisation.
- Dans le code on doit augmenter la valeur du `scaling_factor` dans `showOpticalFlow` pour obtenir un bon résultat.

- La solution  $(u, v)$  est plus lisse et moins sensible aux variations locales, ce qui réduit l'impact du bruit. Cependant, cela peut aussi rendre la solution moins fidèle aux données réelles, en lissant les détails fins des mouvements.

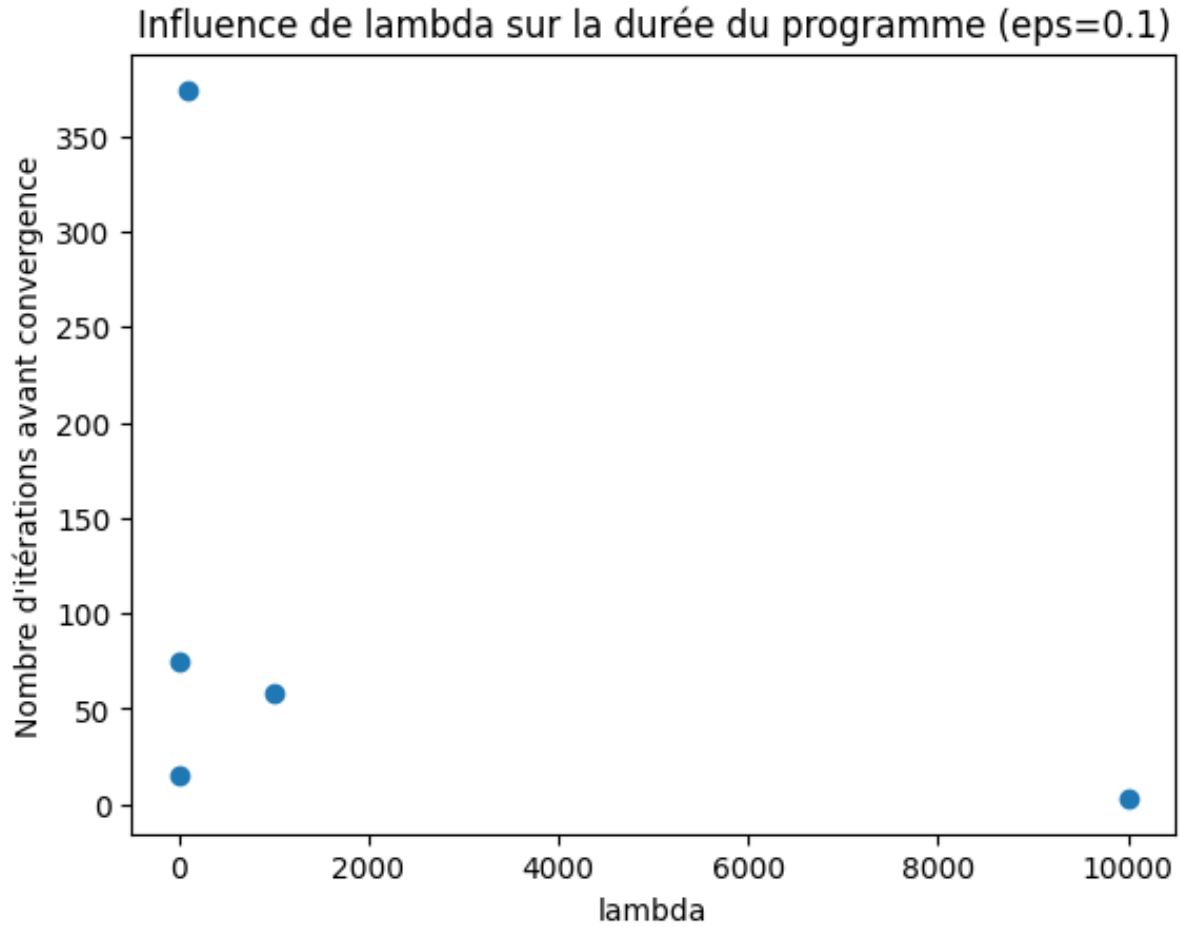


Figure 6: Influence de  $\lambda$  sur la durée du programme

On remarque que lorsque  $\lambda$  est petit, l'énergie converge rapidement, puis quand il augmente le nombre d'itérations augmente. Cependant, si  $\lambda$  devient trop grand, alors le nombre d'itérations redevient petit mais la détection du mouvement perd en précision. On constate ainsi que l'on a une meilleure détection quand le nombre d'itération est haut ( $\lambda = 100$ ).

### 3 Méthode de Lucas et Kanade

#### 3.1 Principe

La méthode de Lucas-Kanade prend un voisinage autour du point. Ainsi, tous les points dans ce voisinage ont le même mouvement. Nous pouvons trouver  $(f_x, f_y, f_t)$  pour ces points. Notre problème se ramène donc à une résolution de plusieurs équations avec deux inconnues, ce qui est problématique. Cependant une solution est obtenue avec la méthode des moindres carrés et c'est ce que nous allons montrer ci-dessous. Nous partons de l'équation du flux optique :

$$I_x u + I_y v + I_t = 0$$

Pour chaque pixel dans un voisinage  $\Omega$  de taille  $n \times n$ , nous cherchons à minimiser l'erreur quadratique de cette équation pour tous les pixels du voisinage. L'erreur quadratique pour un pixel  $(i, j)$  est :

$$E(i, j) = (I_x(i, j)u + I_y(i, j)v + I_t(i, j))^2$$

Nous cherchons donc à minimiser la somme des erreurs quadratiques pour tous les pixels dans le voisinage  $\Omega$  :

$$\sum_{(i,j) \in \Omega} (I_x(i, j)u + I_y(i, j)v + I_t(i, j))^2$$

Pour simplifier cette expression, nous vectorisons les gradients et les termes de l'équation :

$$\mathbf{g}_x = \begin{bmatrix} I_x(i_1, j_1) \\ I_x(i_2, j_2) \\ \vdots \\ I_x(i_{n^2}, j_{n^2}) \end{bmatrix}, \quad \mathbf{g}_y = \begin{bmatrix} I_y(i_1, j_1) \\ I_y(i_2, j_2) \\ \vdots \\ I_y(i_{n^2}, j_{n^2}) \end{bmatrix}, \quad \mathbf{g}_t = \begin{bmatrix} I_t(i_1, j_1) \\ I_t(i_2, j_2) \\ \vdots \\ I_t(i_{n^2}, j_{n^2}) \end{bmatrix}$$

Nous formons la matrice  $A$  et le vecteur  $b$  comme suit :

$$A = [\mathbf{g}_x \quad \mathbf{g}_y] = \begin{bmatrix} I_x(i_1, j_1) & I_y(i_1, j_1) \\ I_x(i_2, j_2) & I_y(i_2, j_2) \\ \vdots & \vdots \\ I_x(i_{n^2}, j_{n^2}) & I_y(i_{n^2}, j_{n^2}) \end{bmatrix}$$

$$b = -\mathbf{g}_t = \begin{bmatrix} -I_t(i_1, j_1) \\ -I_t(i_2, j_2) \\ \vdots \\ -I_t(i_{n^2}, j_{n^2}) \end{bmatrix}$$

Nous cherchons maintenant à minimiser l'expression quadratique suivante et c'est ici que la méthode des moindres carrés intervient :

$$\|A \begin{bmatrix} u \\ v \end{bmatrix} + b\|^2$$

Pour cela, nous introduisons une matrice de poids  $W$  :

$$\|W(A \begin{bmatrix} u \\ v \end{bmatrix} + b)\|^2$$

Pour minimiser cette expression, nous devons résoudre le système d'équations normal suivant :

$$A^T W^2 A \begin{bmatrix} u \\ v \end{bmatrix} = A^T W^2 b$$

En posant :

$$M = A^T W^2 A \quad \text{et} \quad \nu = A^T W^2 b$$

nous avons :

$$M \begin{bmatrix} u \\ v \end{bmatrix} = \nu$$

La matrice  $M$  et le vecteur  $\nu$  sont explicitement donnés par :

$$M = \begin{bmatrix} \sum_{\Omega} w^2(i, j) I_x(i, j)^2 & \sum_{\Omega} w^2(i, j) I_x(i, j) I_y(i, j) \\ \sum_{\Omega} w^2(i, j) I_x(i, j) I_y(i, j) & \sum_{\Omega} w^2(i, j) I_y(i, j)^2 \end{bmatrix}$$

$$\nu = \begin{bmatrix} -\sum_{\Omega} w^2(i, j) I_x(i, j) I_t(i, j) \\ -\sum_{\Omega} w^2(i, j) I_y(i, j) I_t(i, j) \end{bmatrix}$$

Ainsi, pour chaque pixel  $(i, j)$ , nous résolvons le système linéaire pour trouver les composantes  $u$  et  $v$  du flux optique en inversant  $M$  avec la méthode de Tikhonov qui permet de toujours inverser la matrice:

$$M \begin{bmatrix} u \\ v \end{bmatrix} = \nu$$

## 3.2 Influence des paramètres

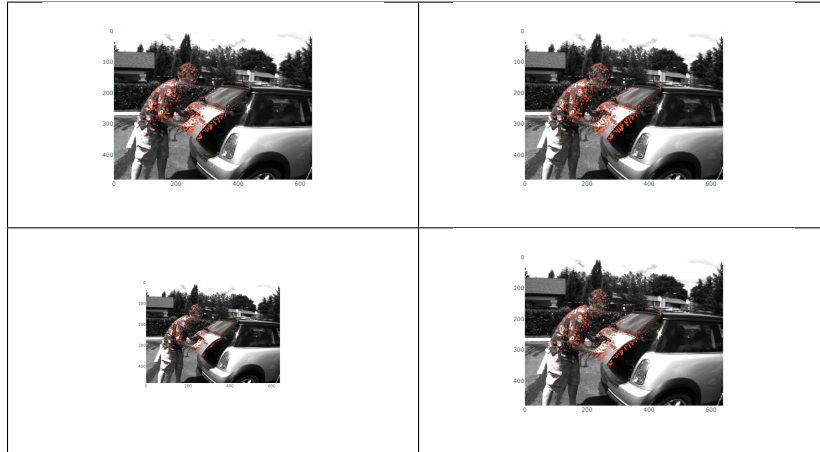


Table 1: Ligne 1 ( voisin =3 et poids linéaire) Ligne 2 ( voisin =5 et poids gaussien)

### 3.2.1 Poids

Les poids gaussiens, qui accordent plus d'importance aux pixels centraux du voisinage, favorisent une meilleure résistance au bruit. En revanche, les poids linéaires, qui diminuent progressivement l'influence des pixels en fonction de leur distance au centre, offrent une transition plus douce et peuvent mieux capturer les variations spatiales graduelles dans l'image.

### 3.2.2 Nombre de voisins

Le nombre de voisins, ou la taille du voisinage, influence directement la quantité d'information utilisée pour estimer le flux optique. Un voisinage plus large permet de capturer plus de contexte spatial. Cependant, un voisinage trop large peut également intégrer trop de variabilité, réduisant la précision de l'estimation locale et augmentant les calculs. À l'inverse, un voisinage trop petit peut ne pas capturer suffisamment d'information pour une estimation fiable. On doit donc trouver un équilibre pour choisir une taille de voisinage appropriée en fonction des caractéristiques de l'image et du mouvement attendu.

## 4 Application à une séquence d'images

L'algorithme des différences finies est utilisé pour estimer le flux optique entre une séquence d'images dans le domaine de la vision par ordinateur. Plutôt que de simplement calculer le gradient temporel entre deux images, cet algorithme prend en compte l'ensemble de la séquence d'images pour une estimation plus robuste du flux optique.

Soit  $I_i(x, y)$  la fonction d'intensité de l'image  $i$  à la position  $(x, y)$ , où  $i$  varie de 1 à  $n$  pour une séquence d'images de longueur  $n$ . Le gradient temporel  $\nabla I(x, y, t)$  est défini comme la dérivée partielle de l'intensité par rapport au temps  $t$  :

$$\nabla I(x, y, t) = \lim_{\Delta t \rightarrow 0} \frac{I(x, y, t + \Delta t) - I(x, y, t)}{\Delta t} \quad (1)$$

Pour estimer ce gradient temporel sur une séquence d'images, nous utilisons les différences finies centrées. La différence finie centrée  $\delta_t I_i(x, y)$  entre deux images consécutives  $i$  et  $i + 1$  est calculée comme suit :

$$\delta_t I_i(x, y) = I_{i+1}(x, y) - I_i(x, y) \quad (2)$$

Cette différence finie nous donne une estimation du gradient temporel entre chaque paire d'images consécutives dans la séquence.

Ainsi, nous avons un flux optique qui dépend du temps.

## 5 Choix lors de l'implémentation

### 5.1 Méthode 1

**Filtre de Sobel** : On a utilisé des filtres dérivateurs de Sobel car la méthode de différence finie est trop peu précise dans notre cas.

**Poids** : Dans l'algorithme, on utilise des filtres moyenneurs pour faire une moyenne sur les voisins car, avec des poids gaussiens, la différence est minime.

### 5.2 Méthode 2

**Tikhonov** : Pour inverser  $M$  et on utilise un  $\lambda$  faible (0,1) pour ne pas trop impacter la détection du mouvement.

Enfin, dans les deux méthodes, nous avons normalisé les images pour simplifier les calculs.

## 6 Conclusion

Ces deux méthodes, bien qu'assez efficaces, sont peu robustes aux occlusions. Comme se sont des approches dites par intensité, le flux est difficilement perçu sur les zones ne changeant pas d'intensité.