

# TP de Vision Robotique

Séance 2

Hugo Miquel & Louis Mennrath

November 14, 2025

# 1 Introduction

L'objectif de ce TP est de détecter les voitures en mouvement sur une séquence vidéo d'autoroute (environ 600 images). Deux approches seront utilisées : la soustraction de fond et l'analyse du flot optique.

## 2 Détection par soustraction de fond

### 2.1 Principe

Pour chaque pixel  $(i, j)$ , on calcule sa moyenne  $\mu(i, j)$  et son écart-type  $\sigma(i, j)$  sur  $N$  images en niveaux de gris :

$$\mu(i, j) = \frac{1}{N} \sum_{k=1}^N I_k(i, j) \quad ; \quad \sigma(i, j) = \sqrt{\frac{1}{N} \sum_{k=1}^N (I_k(i, j) - \mu(i, j))^2} \quad (1)$$

Les pixels du fond (route, bâtiments) ont un  $\sigma$  faible (valeur constante), tandis que les pixels où passent des véhicules ont un  $\sigma$  élevé (forte variation).

### 2.2 Résultats

Trois valeurs de  $N$  ont été testées :  $N = 5$ ,  $N = 20$  et  $N = 100$  (figures 1 et 2).



Figure 1: Images moyennes pour différentes valeurs de  $N$

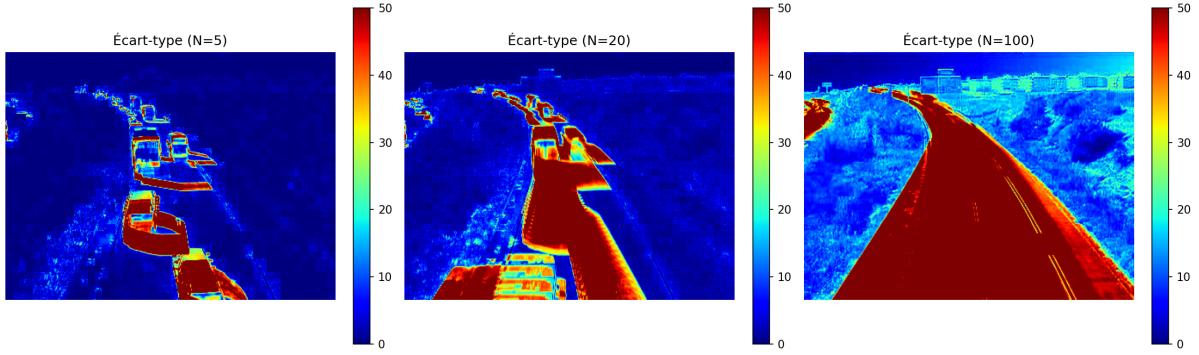


Figure 2: Images d'écart-type pour différentes valeurs de  $N$ . Les zones rouges/jaunes indiquent les pixels à forte variation (passage de véhicules).

Les images d'écart-type (figure 2) révèlent clairement les zones de mouvement.

### 2.3 Discussion

La détection des véhicules s'effectue par seuillage sur  $\sigma$  : un pixel appartient au fond si  $\sigma(i, j) < \text{seuil}$ . Le choix de  $N$  influence la qualité de la détection :

- **$N$  trop faible** (ex:  $N = 5$ ) : sensibilité au bruit, statistiques peu robustes
- **$N$  trop élevé** (ex:  $N = 100$ ) : lissage excessif, perte de précision temporelle
- **Compromis** :  $N = 20$  offre un bon équilibre entre robustesse et réactivité

Le seuil doit être ajusté pour ne conserver que les pixels présentant une variation significative, tout en éliminant le bruit de fond.

### 2.4 Application du seuillage

L'application d'un seuil  $\sigma > 30$  permet de segmenter les zones de mouvement (figure 3). Les pixels blancs représentent les zones où des véhicules ont été détectés.

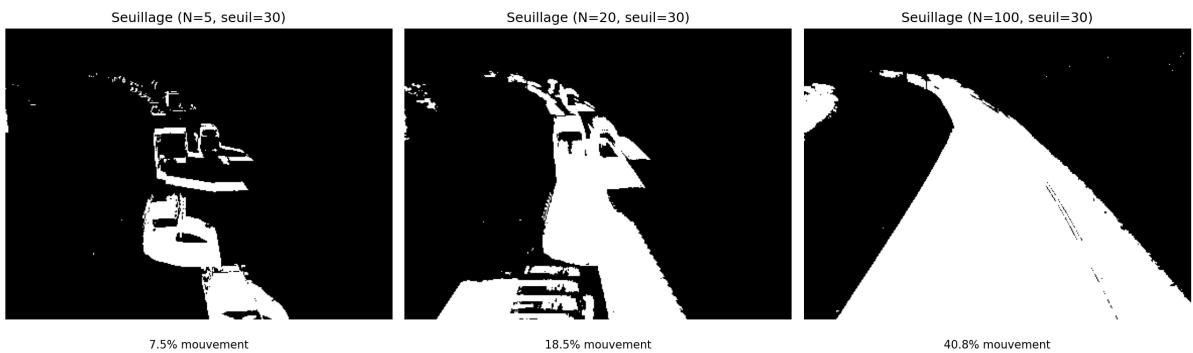


Figure 3: Résultat du seuillage sur l'écart-type ( $\sigma > 30$ ) pour les trois valeurs de  $N$ . Les zones blanches correspondent aux pixels de mouvement (voitures).

On observe que :

- Avec  $N = 5$  : présence de bruit, détection moins stable
- Avec  $N = 20$  : bon compromis, détection claire des trajectoires
- Avec  $N = 100$  : détection plus lissée, trajectoires élargies

Ces résultats confirment que le choix de  $N$  et du seuil sont cruciaux pour obtenir une détection fiable tout en limitant les faux positifs.

## 2.5 Limites de l'approche globale

En analysant l'évolution temporelle du pixel (200, 150) situé au milieu de la route (figure 4), on constate que le calcul global de  $(\mu, \sigma)$  sur toute la séquence est trompeur.

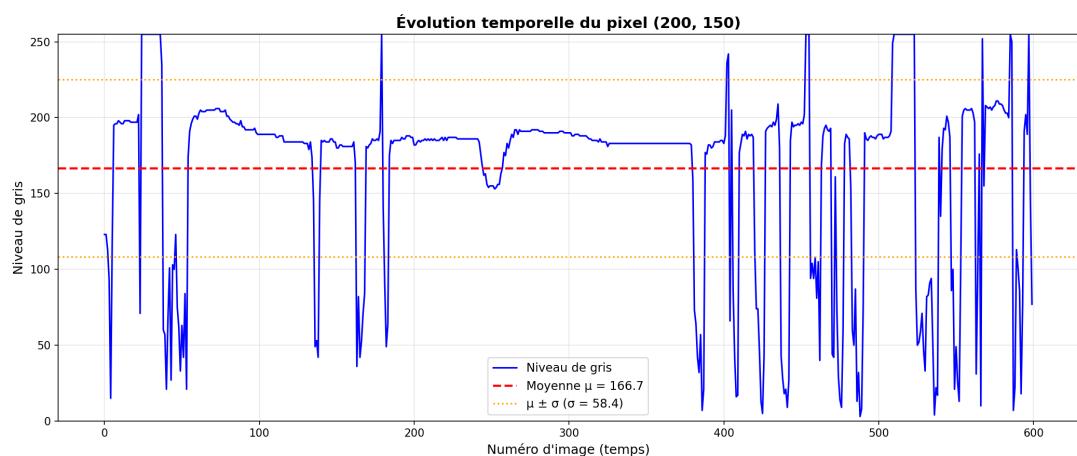


Figure 4: Évolution temporelle du niveau de gris du pixel (200, 150). Les variations brusques correspondent aux passages de véhicules.

L'écart-type calculé sur l'ensemble de la séquence est élevé, ce qui conduit à classer ce pixel comme étant en mouvement. Pourtant, entre les passages de véhicules, le pixel reste stable et correspond au fond de la route. Le calcul global mélange donc des périodes de stabilité (fond) et des transitions (passages de véhicules), ce qui fausse la détection.

## 2.6 Amélioration : détection de plateaux temporels

Pour améliorer la détection, il faut exploiter la cohérence temporelle locale. L'idée est d'identifier des **plateaux** dans le signal, c'est-à-dire des périodes où le pixel est stable, puis de calculer  $(\mu, \sigma)$  uniquement sur ces plateaux.

**Méthode proposée :**

1. Calculer les différences entre images consécutives :  $\Delta_k = |I_k(i, j) - I_{k-1}(i, j)|$
2. Identifier les plateaux : séquences où  $\Delta_k < \text{seuil}_\Delta$  (faible variation)
3. Calculer  $\mu$  et  $\sigma$  uniquement sur les valeurs appartenant aux plateaux

Cette approche permet de mieux caractériser le fond en ignorant les transitions dues aux véhicules, améliorant ainsi la robustesse de la détection.

## 2.7 Algorithme Sigma-Delta ( $\Sigma\Delta$ )

### 2.7.1 Principe

L'algorithme Sigma-Delta est une approche adaptative qui met à jour progressivement, pour chaque pixel, une estimation de la moyenne  $M$  et de la variance  $V$  du fond.

Les mises à jour se font de manière incrémentale :

- Si le pixel actuel est plus clair que  $M$ , on incrémente  $M$  de 1
- Si le pixel actuel est plus sombre que  $M$ , on décrémente  $M$  de 1
- Si la différence  $|I - M|$  est grande, on augmente  $V$ , sinon on la diminue

Un pixel est détecté comme mouvement si :  $|I(i, j) - M(i, j)| > N \cdot V(i, j)$

Le paramètre  $N$  contrôle la sensibilité de la détection.

### 2.7.2 Résultats et discussion

L'algorithme a été testé avec trois valeurs de  $N$  : 1, 2 et 4 (figure 5).

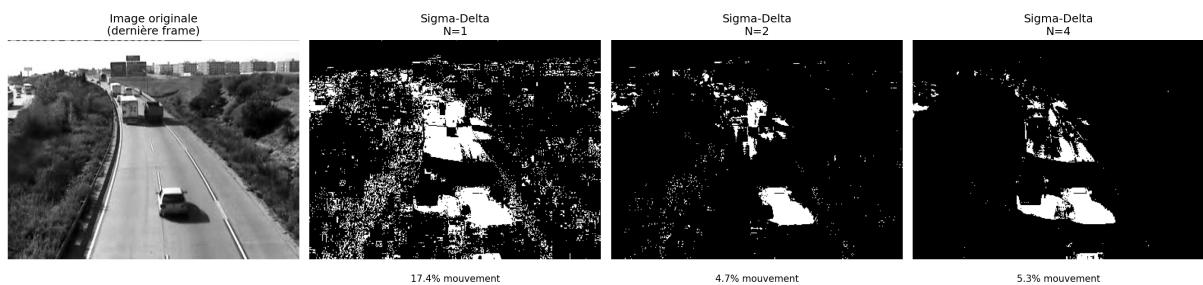


Figure 5: Résultats de l'algorithme Sigma-Delta pour différentes valeurs de  $N$ . Les zones blanches indiquent les mouvements détectés.

On constate que pour  $N = 1$ , l'algorithme est très sensible et génère de nombreux faux positifs, notamment du bruit et des ombres. Avec  $N = 2$ , on obtient un bon équilibre et une détection claire des véhicules. En revanche, pour  $N = 4$ , la détection devient moins sensible et certains véhicules peuvent ne pas être détectés.

Le choix des paramètres est donc crucial :  $N = 2$  offre le meilleur compromis pour cette séquence, tandis qu'une valeur minimale de variance  $V_{min} = 2$  permet d'éviter une sensibilité excessive au bruit.

**Avantages** : adaptation automatique au changement d'éclairage, faible coût mémoire (pas besoin de stocker plusieurs images), calculs simples et rapides.

**Limites** : sensible au choix de  $N$ , convergence lente si un véhicule reste stationnaire longtemps.

## 2.8 Utilisation d'espaces colorimétriques

L'algorithme en niveaux de gris détecte non seulement les véhicules mais aussi leurs ombres. Pour réduire cette détection parasite, on peut exploiter les images couleur dans des espaces colorimétriques différents.

L'espace HSV ne s'avère pas adapté pour filtrer les ombres. En effet, les ombres modifient non seulement la luminosité (canal V), mais aussi la teinte (H) et la saturation (S) (figure 6).

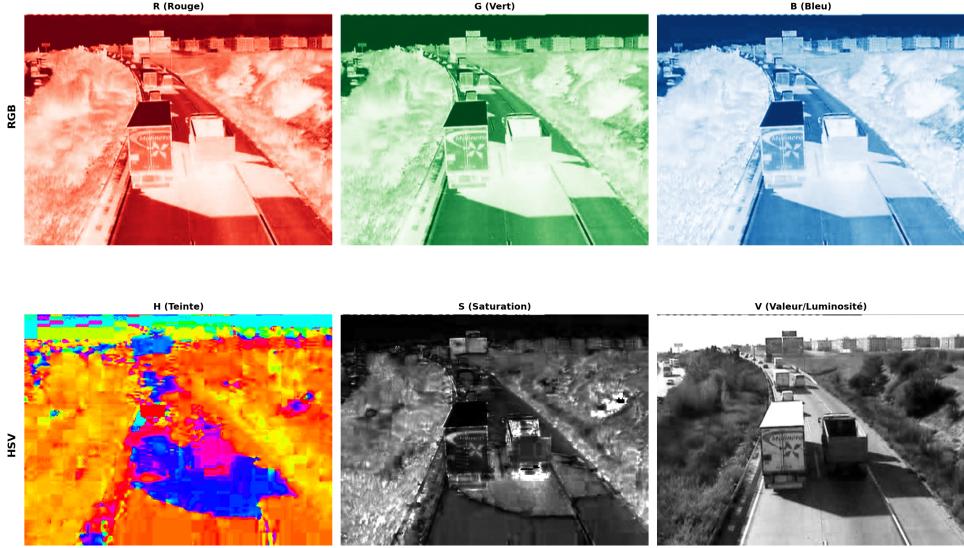
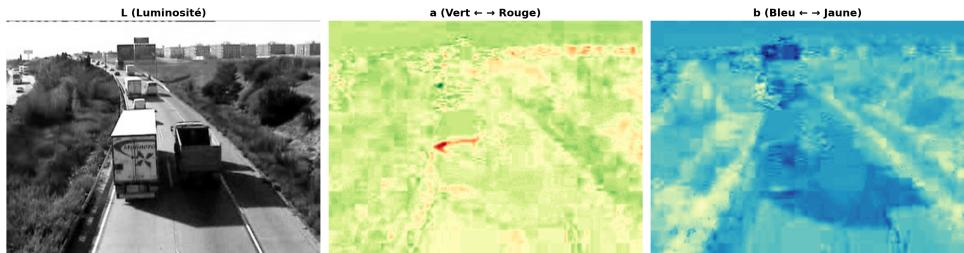


Figure 6: Décomposition en canaux RGB et HSV. Les ombres affectent tous les canaux HSV, y compris H et S.

À l'inverse, l'espace LAB se révèle bien plus efficace. Les canaux a et b représentent uniquement la chrominance, indépendante de la luminosité. Les ombres, qui affectent principalement le canal L, sont ainsi ignorées (figure 7).



**2.8.1 Application de Sigma-Delta sur LAB (a+b)**

Pour exploiter cette propriété, on applique l'algorithme Sigma-Delta directement sur les canaux a et b. On calcule d'abord la magnitude de la chrominance :  $\text{chroma} = \sqrt{a^2 + b^2}$ , puis on applique Sigma-Delta sur cette image en niveaux de gris.



Figure 8: Comparaison des masques de détection. L'approche LAB maintient la détection des véhicules.

### 3 Exercice 2 - Flot Optique

Nous allons présenter le principe fondamental de la méthode classique de Horn & Schunck pour l'estimation du flot optique entre deux images consécutives, ainsi que l'implémentation MATLAB de l'algorithme telle que demandée dans l'énoncé.

#### 3.1 Hypothèse et formulation du problème

L'article de Horn & Schunck repose sur l'hypothèse de **constance de la luminance** :

$$I(x + u, y + v, t + 1) \approx I(x, y, t),$$

où  $(u, v)$  est le déplacement entre deux images successives. Après linéarisation, on obtient :

$$E_x u + E_y v + E_t = 0,$$

où  $E_x, E_y$  sont les dérivées spatiales et  $E_t$  la dérivée temporelle.

Cette équation étant sous-déterminée, les auteurs ajoutent un terme de **régularisation** permettant d'imposer la lisibilité du champ de mouvement :

$$\mathcal{E}(u, v) = \iint (E_x u + E_y v + E_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \, dx \, dy.$$

Le paramètre  $\alpha$  contrôle la force de cette régularisation :

- $\alpha$  petit : flot plus détaillé mais sensible au bruit,
- $\alpha$  grand : flot très lissé mais perte de détails.

#### 3.2 Discrétisation des dérivées (Section 7)

Les dérivées proposées par Horn & Schunck utilisent des noyaux 2x2 symétriques :

$$k_x = \frac{1}{4} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}, \quad k_y = \frac{1}{4} \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}, \quad k_t = \frac{1}{4} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Elles sont appliquées aux deux images pour obtenir  $E_x, E_y, E_t$ .

### 3.3 Laplacien et terme de lissage (Section 8)

Horn & Schunck utilisent une moyenne pondérée des 8 voisins pour approcher le laplacien :

$$\text{avg\_k} = \begin{bmatrix} 1/12 & 1/6 & 1/12 \\ 1/6 & 0 & 1/6 \\ 1/12 & 1/6 & 1/12 \end{bmatrix}.$$

Cette moyenne correspond à l'estimation locale des valeurs  $\bar{u}$  et  $\bar{v}$  utilisées dans la mise à jour du schéma. C'est pour cela que la valeur centrale est 0 et non -1 car la valeur du pixel n'influence pas la valeur de la moyenne des voisins.

### 3.4 Schéma final d'itération (Section 12)

À chaque itération :

$$\begin{aligned} \bar{u} &= \text{avg}(u), & \bar{v} &= \text{avg}(v), \\ u &= \bar{u} - \frac{E_x(E_x\bar{u} + E_y\bar{v} + E_t)}{\alpha^2 + E_x^2 + E_y^2}, \\ v &= \bar{v} - \frac{E_y(E_x\bar{u} + E_y\bar{v} + E_t)}{\alpha^2 + E_x^2 + E_y^2}. \end{aligned}$$

Ce schéma est appliqué environ 100 à 300 fois.

### 3.5 Détection simple des objets en mouvement

Pour détecter si un pixel est en mouvement entre 2 images, on peut calculer les magnitude du flow optique, autrement dit la "force" avec laquelle le pixel bouge entre les 2 images et on peut comparer cette magnitude avec la moyenne des magnitudes de chaque pixel de l'image.

La magnitude du flot optique :

$$m = \sqrt{u^2 + v^2}$$

permet de définir un seuil :

$$m > \mu_m + 0.5\sigma_m$$

pour segmenter les zones en mouvement.

### 3.6 Filtrage Gaussien

Il est mentionné dans l'article que cet algorithme est assez sensible au bruit, il peut être intéressant de filtrer les images pour rendre la détection du flow optique plus fluide. On utilise simplement un filtre gaussien.

## 4 Résultats

## 5 Conclusion