

## Traitement Avancé des Images

### Détection du mouvement

T. Dietenbeck ([thomas.dietenbeck@sorbonne-universite.fr](mailto:thomas.dietenbeck@sorbonne-universite.fr))



## 1 Introduction

## 2 Différence entre images

## 3 Modèles statistiques

## 1 Introduction

## 2 Différence entre images

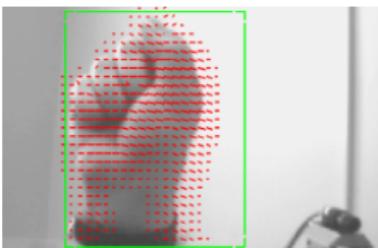
## 3 Modèles statistiques

## Problème

- On travaille sur des séquences d'images : le temps est une nouvelle dimension
- Trois problèmes fondamentaux :
  - ➊ Détection du mouvement
  - ➋ Estimation du mouvement
  - ➌ Suivi du mouvement



Détection



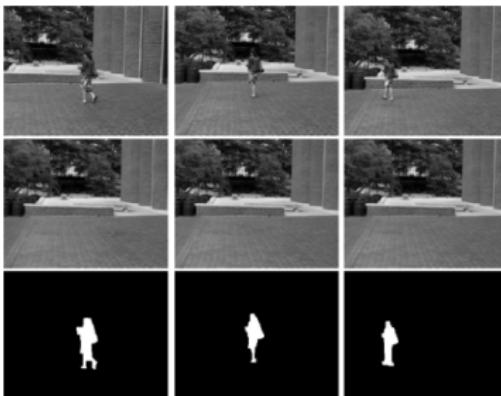
Estimation



Suivi

## Principe

- **Identifier les pixels** en mouvement (*i.e.* qui ont changé entre deux images consécutives)
  - **Extraire les objets** du reste de la scène : gain en complexité et en temps de calcul par la suite
- ⇒ Étape préliminaire fondamentale



## Applications potentielles

- Suivi d'objet(s)
- Interaction humain-ordinateur
- Reconnaissance d'action, d'activité
- Surveillance du traffic routier

## Problème

Beaucoup de difficultés inhérentes aux séquences vidéo

- Illuminations (ombres, changements, lumières artificielles)
- Bruit (capteur)
- Fond "fouillis" ou changeant (caméra mobile, objets de fond avec légers mouvements)

⇒ L'image de fond doit pouvoir s'adapter à ces changements

## Objectif

**Modéliser le fond  $B_t$  de manière à pouvoir extraire le premier plan  $F_t$  dans l'image  $I_t$**

## Notations

- $\mathbf{p} = (x, y)$  : pixel
- $I_t$  : image au temps  $t$
- $B_t$  : image de fond au temps  $t$
- $F_t$  : image de premier plan au temps  $t$
- $D_t$  : image de détection au temps  $t$

## Objectif

**Modéliser** le fond  $B_t$  de manière à pouvoir extraire le premier plan  $F_t$  dans l'image  $I_t$

## Modélisation du fond

- Niveau image ou pixel ?
- Types de modèles
  - Statistique
  - Probabiliste
  - Reposant sur la cohérence spatiale

## Objectif

**Modéliser** le fond  $B_t$  de manière à pouvoir extraire le premier plan  $F_t$  dans l'image  $I_t$

## Contraintes

L'algorithme doit être

- efficace
- rapide
- peu complexe
- peu consommateur de mémoire

## Objectif

**Modéliser le fond  $B_t$  de manière à pouvoir extraire le premier plan  $F_t$  dans l'image  $I_t$**

## Approches étudiées

- Différence de deux images
  - Fond connu
  - Fond vu comme l'image précédente
  - Fond vu comme une information statistique du passé
  - Fond modélisé par filtrage temporel
- Différence de trois images
- Modèle gaussien
- Mélange de gaussiennes
- Estimateur de densité par noyau
- Décomposition en valeurs singulières (ACP)

1 Introduction

2 Différence entre images

3 Modèles statistiques

## Hypothèse

- On dispose de l'image de fond  $B_t$

## Principe

- $F_t(\mathbf{p}) = |I_t(\mathbf{p}) - B_t(\mathbf{p})|$
- $D_t(\mathbf{p}) = \begin{cases} 1 & \text{si } F_t(\mathbf{p}) > s \\ 0 & \text{sinon} \end{cases}$

Fond  $B_t$ Image  $I_t$ Détection  $D_t$

## Limitations

- Si un objet a une **couleur similaire** au fond, il n'est pas détecté
- Si un objet entre dans la scène et s'arrête, il continue à être détecté
- **Mouvement dans le fond** (branches qui bougent, caméra mobile)



Fond



Image  $I_t$



Détection  $D_t$

## Limitations

- Si un objet a une **couleur similaire** au fond, il n'est pas détecté
- Si un objet entre dans la scène et **s'arrête**, il continue à être détecté
- **Mouvement dans le fond** (branches qui bougent, caméra mobile)



Fond

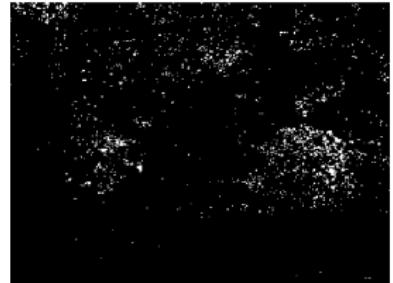
Image  $I_t$ Détection  $D_t$

## Limitations

- Si un objet a une **couleur similaire** au fond, il n'est pas détecté
- Si un objet entre dans la scène et **s'arrête**, il continue à être détecté
- **Mouvement dans le fond** (branches qui bougent, caméra mobile)



Fond

Image  $I_t$ Détection  $D_t$

## Principe

- On suppose que le fond est l'**image précédente** :  $B_t = I_{t-1}$
- $F_t(\mathbf{p}) = |I_t(\mathbf{p}) - I_{t-1}(\mathbf{p})|$
- $D_t(\mathbf{p}) = \begin{cases} 1 & \text{si } |I_t(\mathbf{p}) - I_{t-1}(\mathbf{p})| > s \\ 0 & \text{sinon} \end{cases}$

Fond  $I_{t-1}$ Image  $I_t$ Détection  $D_t$

## Avantages

- S'adapte aux petits changements
- Objets s'arrêtant ne sont plus détectés

## Inconvénients

- Ne détecte que les **contours** en mouvement, pas tout l'objet
- Dépend de l'**échelle temporelle**



# Différence de trois images

## Principe

- Considérer les images de différences dans une **fenêtre temporelle**  $[-\delta_t; +\delta_t]$  autour de  $t$
- $D_t = D_{t-\delta_t} \cdot D_{t+\delta_t}$ , avec  $\cdot$  le produit pixel à pixel et  $D_{t \pm \delta_t}$  les détections entre  $t$  et  $\pm \delta_t$

## Limitations

- Choix de  $\delta_t$
- Les silhouettes de l'objet doivent bien être séparées entre  $t$  et  $t \pm \delta_t$



Détection  $D_{t-\delta_t}$



Détection  $D_{t+\delta_t}$



Détection  $D_t$

# Fond = information statistique du passé

## Principe

- ① Le fond est la **moyenne** des  $n$  dernières images :

$$B_t(\mathbf{p}) = \frac{1}{n} \sum_{i=t-n}^t I_i(\mathbf{p})$$

- ② Le fond est la **médiane** sur les  $n$  dernières images :

$$B_t = \text{mediane}(I_{t-n}, \dots, I_t)$$

## Limitation principale

Nécessite une **grande capacité de stockage** en mémoire ( $n$  images)

# Mise-à-jour du fond par filtrage temporel

## Principe

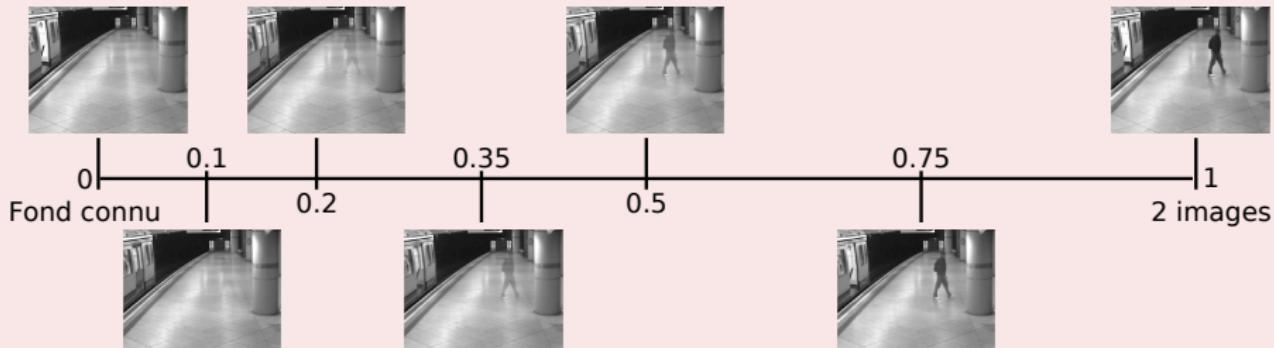
- L'image de référence est initialisée avec la première image
- Le fond est mis-à-jour en fonction de sa **précédente apparence** et de celle de **l'image en cours** :

$$\begin{cases} F_t &= |I_t - B_t| \\ B_{t+1} &= (1 - \alpha)B_t + \alpha I_t \end{cases}$$

- $\alpha$  est le paramètre d'apprentissage (généralement entre 0.01 et 0.1)

## Problème

- Quelle valeur de  $\alpha$  choisir ?



# Mise-à-jour du fond par filtrage temporel

## Avantages

- Un objet qui s'arrête disparaît progressivement (idem s'il se remet en mouvement)
- S'adapte à des changements lents d'illumination

## Inconvénients

- Un effet de rémanence est visible si l'objet se déplace lentement ou s'il est volumineux



Fond à  $t = 0$



Fond à l'instant  $t$



Détection  $D_t$

## Avantages

### Solution

- Rapide
- Souvent peu coûteuse

## Inconvénients

- Aucune valeur de seuil valable pour l'ensemble des pixels à un même instant ni au cours du temps
- Problème en cas de multiples objets ou de changement d'éclairage
- Les méthodes sont orientées pixel  $\Rightarrow$  seul l'historique récent du fond est considéré, pas le voisinage de ses pixels (correlations spatiales, etc.)

## 1 Introduction

## 2 Différence entre images

## 3 Modèles statistiques

- Modèle gaussien
- Mélange de gaussiennes
- Estimateur de densité par noyaux
- Apprentissage

## Densité de probabilité d'une gaussienne 1D

- Correspond au cas où  $I_t$  est une image en niveau de gris

$$\mathcal{N}(I_t, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(I_t - \mu)^2}{2\sigma^2}\right)$$

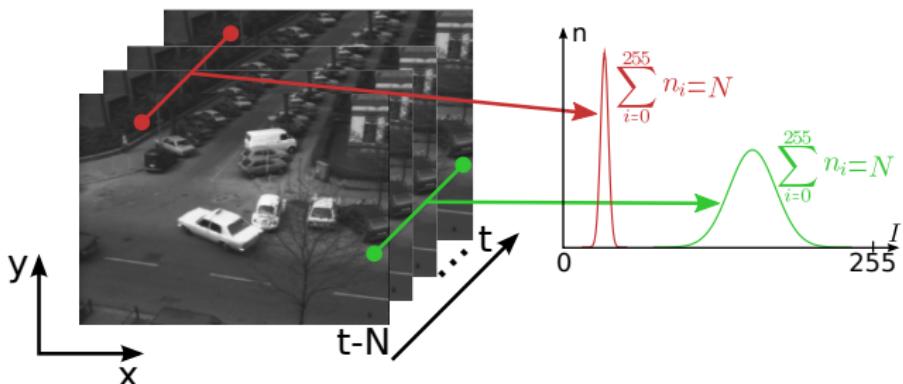
avec

- $\mathcal{N}$  : DdP de la gaussienne
- $\mu$  : moyenne de la gaussienne
- $\sigma$  : écart-type de la gaussienne

## Remarques

Une gaussienne modélise l'évolution temporelle de l'intensité d'**un seul pixel**

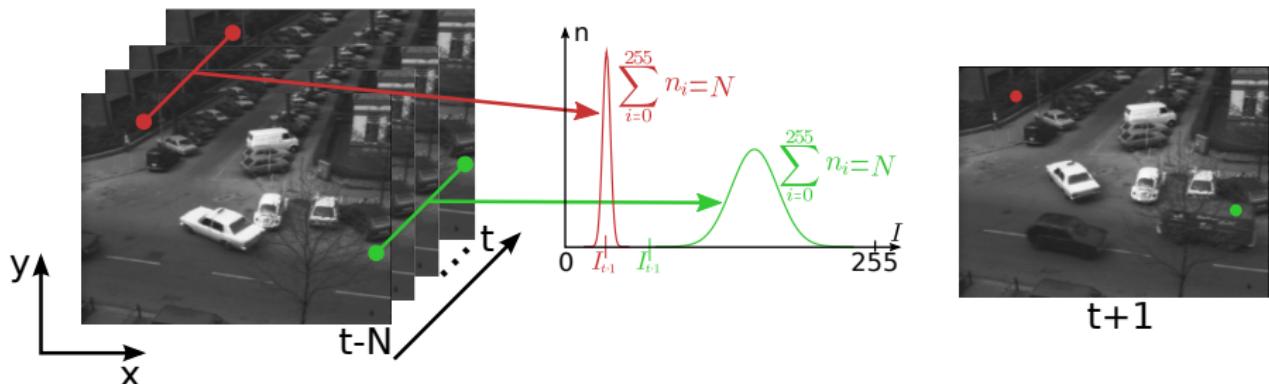
- ⇒ on aura donc autant de gaussiennes que de pixels
- ⇒  $\mu$  et  $\sigma$  sont des images de mêmes dimensions que  $I_t$



## Remarques

Une gaussienne modélise l'évolution temporelle de l'intensité d'**un seul pixel**

- ⇒ on aura donc autant de gaussiennes que de pixels
- ⇒  $\mu$  et  $\sigma$  sont des images de mêmes dimensions que  $I_t$



## Principe

- La probabilité d'observer une intensité  $I_t(\mathbf{p})$  à l'instant  $t$  peut être **modélisée par une gaussienne**  $\mathcal{N}(I_t(\mathbf{p}), \mu_t(\mathbf{p}), \sigma_t(\mathbf{p}))$
- $D_t(\mathbf{p}) = \begin{cases} 1 & \text{si } |I_t(\mathbf{p}) - \mu_t(\mathbf{p})| > k\sigma_t(\mathbf{p}) \\ 0 & \text{sinon} \end{cases}$ , où
  - $\mu_t(\mathbf{p})$  : moyenne des valeurs prises par le pixel (entre  $t$  et  $t - n$ )
  - $\sigma_t(\mathbf{p})$  : écart type des valeurs prises par le pixel (entre  $t$  et  $t - n$ )
  - en pratique,  $k = 2.5$

## Mise à jour des paramètres

La moyenne et l'écart type de chaque pixel sont mis à jour par :

- $\mu_{t+1}(\mathbf{p}) = \alpha I_t(\mathbf{p}) + (1 - \alpha) \mu_t(\mathbf{p})$
- $\sigma_{t+1}(\mathbf{p}) = \sqrt{\alpha(\mu_t(\mathbf{p}) - I_t(\mathbf{p}))^2 + (1 - \alpha)\sigma_t^2(\mathbf{p})}$

où  $\alpha$  est le facteur d'apprentissage

## Avantages

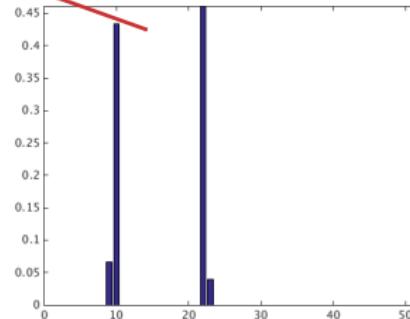
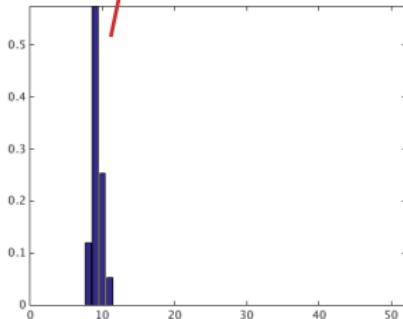
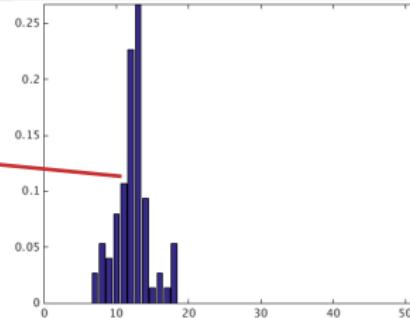
- Simple
- Peu de mémoire requise ( $\mu$  et  $\sigma$ )
- Un seuil spécifique à chaque pixel

## Question

- Est-ce que la valeur d'un pixel suit une loi gaussienne au cours du temps ?

## Question

- Est-ce que la valeur d'un pixel suit une loi gaussienne au cours du temps ?  
⇒ La valeur suit plutôt un **mélange de gaussiennes**



## Principe

- La probabilité d'observer une intensité  $I_t(\mathbf{p})$  à l'instant  $t$  peut être modélisée par un **mélange de gaussiennes** :

$$p(I_t) = \sum_{m=1}^M \omega_t^m(\mathbf{p}) \mathcal{N}^m(I_t(\mathbf{p}), \mu_t^m(\mathbf{p}), \sigma_t^m(\mathbf{p})), \text{ où}$$

- $\mu_t^m(\mathbf{p}), \sigma_t^m(\mathbf{p})$  : moyenne et écart type de la gaussienne  $m$
- $\omega_t^m(\mathbf{p})$  : poids de la gaussienne  $m$  (définit son importance dans la distribution globale), avec  $\sum_{m=1}^M \omega_t^m(\mathbf{p}) = 1$
- $M$  : nombre de gaussiennes (généralement  $M \in [3; 5]$ )

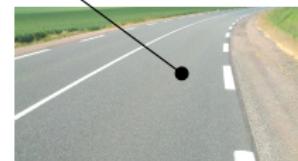
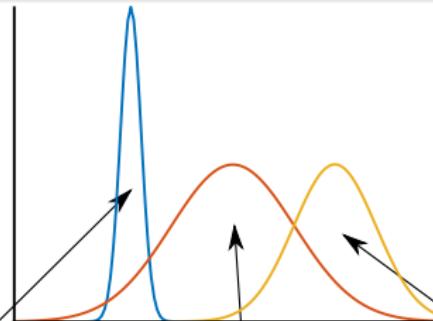
## Référence

C. Stauffer & W.E.L. Grimson (1999), *Adaptive background mixture models for real-time tracking*. In Computer Vision and Pattern Recognition (CVPR'99), vol.2, pp.246–252

## Principe

- La probabilité d'observer une intensité  $I_t(\mathbf{p})$  à l'instant  $t$  peut être modélisée par un **mélange de gaussiennes** :

$$p(I_t) = \sum_{m=1}^M \omega_t^m(\mathbf{p}) \mathcal{N}^m(I_t(\mathbf{p}), \mu_t^m(\mathbf{p}), \sigma_t^m(\mathbf{p}))$$



## Mise à jour des paramètres des gaussiennes

Si  $|I_t(\mathbf{p}) - \mu_t^m(\mathbf{p})| \leq k\sigma_t^m(\mathbf{p})$ , alors

- le pixel appartient à la gaussienne  $m$
- on met à jour les paramètres de cette gaussienne :

$$\begin{cases} \mu_{t+1}^m(\mathbf{p}) &= \rho_t(\mathbf{p})I_t(\mathbf{p}) + (1 - \rho_t(\mathbf{p}))\mu_t^m(\mathbf{p}) \\ \sigma_{t+1}^m(\mathbf{p}) &= \sqrt{\rho_t(\mathbf{p})(\mu_t^m(\mathbf{p}) - I_t(\mathbf{p}))^2 + (1 - \rho_t(\mathbf{p}))(\sigma_t^m(\mathbf{p}))^2} \\ \rho_{t+1}(\mathbf{p}) &= \alpha \mathcal{N}^m(I_t(\mathbf{p}), \mu_t^m(\mathbf{p}), \sigma_t^m(\mathbf{p})) \end{cases}$$

## Remarque

Si plusieurs gaussiennes satisfont  $|I_t(\mathbf{p}) - \mu_t^m(\mathbf{p})| \leq k\sigma_t^m(\mathbf{p})$  pour un pixel, on ne met à jour que la meilleure gaussienne.

## Mise à jour des paramètres des gaussiennes

Si le pixel **n'appartient à aucune des gaussiennes**, on remplace la moins probable par une nouvelle gaussienne de paramètres :

- $\mu_t^m(\mathbf{p}) = I_t(\mathbf{p})$
- $\sigma_t^m(\mathbf{p}) = 2 \max_k (\sigma_t^k(x, y))$  : une **grande** valeur
- $\omega_t^m(\mathbf{p}) = 0.5 \min_k (\omega_t^k(x, y))$  : une **petite** valeur

## Remarque

La gaussienne la moins probable correspond à celle dont le rapport  $\omega_t^m / \sigma_t^m$  est minimal. En effet

- $\omega_t^m$  petit  $\Rightarrow$  faible poids dans le mélange de gaussiennes
- $\sigma_t^m$  grand  $\Rightarrow$  la gaussienne tend vers une loi uniforme

## Mise à jour des poids des gaussiennes

- Les poids de l'ensemble des gaussiennes sont **mis-à-jour** :

$$\omega_{t+1}^m(\mathbf{p}) = \begin{cases} \omega_t^m(\mathbf{p}) & \text{si } \mathbf{p} \text{ appartient à la gaussienne } m \\ (1 - \alpha)\omega_t^m(\mathbf{p}) & \text{sinon} \end{cases}$$

avec  $\alpha$  le facteur d'apprentissage

- Les poids sont ensuite **normalisés** pour satisfaire  $\sum_{m=1}^M \omega_{t+1}^m(\mathbf{p}) = 1$

## Classification d'un pixel

- Trier les  $M$  gaussiennes par ordre de  $\omega_t^m(\mathbf{p})/\sigma_t^m(\mathbf{p})$  décroissant
- **Heuristique** : les  $B$  premières gaussiennes modélisent le fond, avec :

$$B = \arg \min_b \left( \sum_{i=1}^b \omega_t^i(\mathbf{p}) > T \right)$$

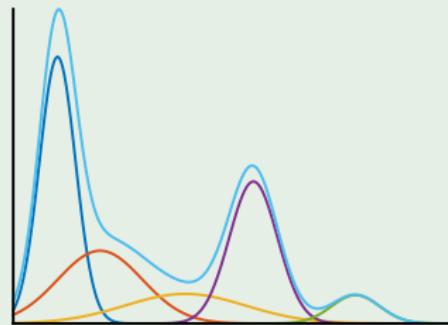
et  $T$  une proportion minimale de fond à fixer

- Si le pixel appartient à une de ces  $B$  gaussiennes, il est considéré statique ( $D_t = 0$ ) sinon en mouvement ( $D_t = 1$ )

## Exemple

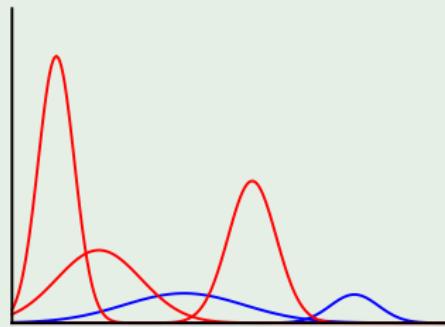
Sur l'histogramme des intensités au cours du temps pour un pixel  $p$ , on a estimé les gaussiennes suivantes :

$i$	1	2	3	4	5
$\mu_i$	25	50	100	140	200
$\sigma_i$	15	35	50	20	20
$\omega_i$	0.35	0.22	0.13	0.25	0.05



## Exemple : Séparation des gaussiennes (Fond / Objet)

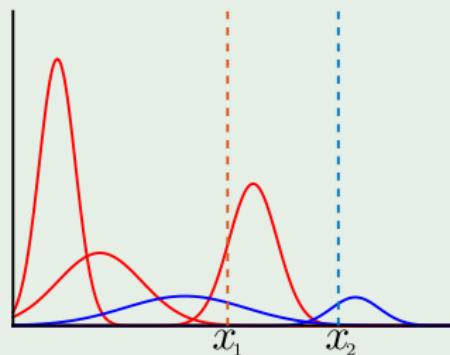
$i$	1	2	3	4	5
$\mu_i$	25	50	100	140	200
$\sigma_i$	15	35	50	20	20
$\omega_i$	0.35	0.22	0.13	0.25	0.05



- On classe les gaussiennes par ordre de  $\omega/\sigma$  décroissant :  $1 > 4 > 2 > 3 > 5$
- On cherche  $B$  tel que :  $\sum_{b=1}^B \omega_{k_b} > T$ , où  $k_b = \{1, 4, 2, 3, 5\}$
- En prenant  $T = 0.8$  (80%), on trouve  $B = 3$ .
- Les gaussiennes 1, 4 et 2 (en rouge) modélisent donc le fond et les gaussiennes 3 et 5 (en bleu) l'objet en mouvement.

## Exemple : Classification

$i$	1	2	3	4	5
$\mu_i$	25	50	100	140	200
$\sigma_i$	15	35	50	20	20
$\omega_i$	0.35	0.22	0.13	0.25	0.05



- On rappelle :  $p_i(x) = \frac{\omega_i}{\sigma_i \sqrt{2\pi}} e^{-\frac{(x - \mu_i)^2}{2\sigma_i^2}}$
- Pour un pixel  $x_1 = 125$ , on trouve  $\max(p_i(x_1)) = 4$ , donc  $x_1 \in$  Fond.
- Pour un pixel  $x_2 = 200$ , on trouve  $\max(p_i(x_2)) = 5$ , donc  $x_2 \in$  Objet.

## Avantages

- **Seuil différent** pour chaque pixel et s'adaptant au cours du temps
- Gère bien les objets qui s'arrêtent (et deviennent fond) et bougent à nouveau

## Inconvénients

- Ne gère pas les changements brutaux d'illumination
- **L'initialisation** des gaussiennes est essentielle !
- **Beaucoup de paramètres** et pas assez de données pour les ajuster correctement  
⇒ méthodes non paramétriques

# Estimateur de densité par noyaux

## Principe

- **Méthode non paramétrique** : le modèle est obtenu à partir de l'échantillon observé
- Connaissant les  $n$  dernières valeurs d'un pixel, on peut estimer la probabilité de sa valeur courante à l'aide d'un **ensemble de noyaux**

## Référence

A. Elgammal, R. Duraiswami, D. Harwood & L.S. Davis (2002), *Background and foreground modeling using nonparametric kernel density estimation for visual surveillance*. In Proceedings of the IEEE, vol.90, no.7, pp.1151–1163

# Estimateur de densité par noyaux

## Idée

- La probabilité d'observer une intensité  $I_t(\mathbf{p})$  est modélisée à l'aide de ses  $n$  plus récentes valeurs  $\{I_i\}_{i=t-n}^{t-1}$ , chacune lissée par une fonction noyau (ou kernel) :

$$p(I_t) = \frac{1}{n} \sum_{i=1}^n K_\sigma(I_t - I_{t-i})$$

où  $K_\sigma$  est la fonction noyau de support  $\sigma$

## Référence

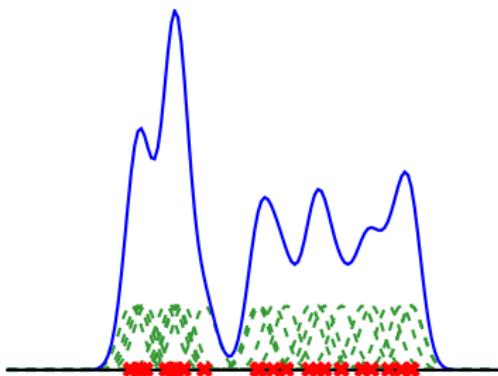
A. Elgammal, R. Duraiswami, D. Harwood & L.S. Davis (2002), *Background and foreground modeling using nonparametric kernel density estimation for visual surveillance*. In Proceedings of the IEEE, vol.90, no.7, pp.1151–1163

# Estimateur de densité par noyaux

## Estimateur de densité par noyaux

- En général, on choisit comme fonction noyau une gaussienne de moyenne nulle et d'écart type  $\sigma$  :

$$p(I_t) = \frac{1}{n} \sum_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(I_t - I_{t-i})^2}{2\sigma^2}}$$



## Estimateur de densité par noyaux

### Estimateur de densité par noyaux

- En général, on choisit comme fonction noyau une gaussienne de moyenne nulle et d'écart type  $\sigma$  :

$$p(I_t) = \frac{1}{n} \sum_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(I_t - I_{t-i})^2}{2\sigma^2}}$$

### Remarque

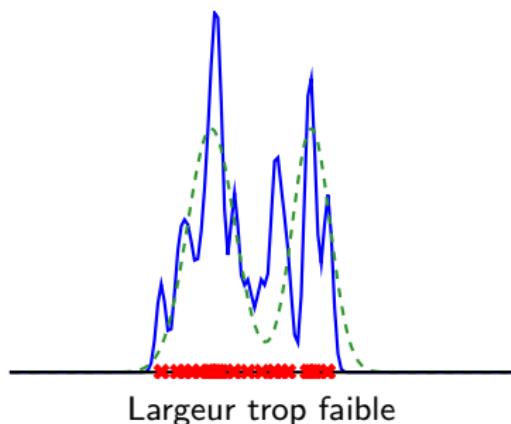
- La probabilité  $p$  modélise l'appartenance au fond.
- On considère donc plutôt :

$$p_O(I_t) = 1 - p(I_t)$$

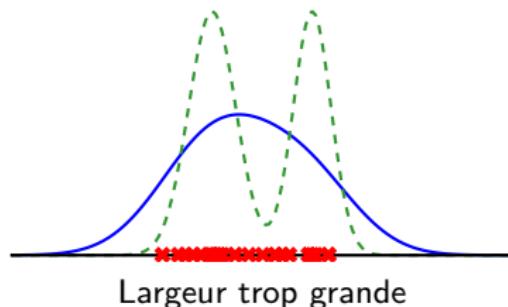
Le pixel appartient à l'objet en mouvement si  $p_O(I_t) > s$ , où  $s$  est un seuil à fixer

## Choix de la largeur du filtre

- Largeur trop faible : faux d'artefacts
- Largeur trop grande : lissage trop grand



Largeur trop faible



Largeur trop grande

## Choix de la largeur du filtre

- Largeur trop faible : faux d'artefacts
- Largeur trop grande : lissage trop grand

## Choix de $\sigma$

$$\sigma = \frac{\text{median}_i^{n-1}(|I_{i+1} - I_i|)}{0.68\sqrt{2}}$$

d'après

- A. Elgammal, D. Harwood, & L. Davis (2000). *Non-parametric Model for Background Subtraction*. In Eur. Conf. Computer Vision (ECCV'00), pp. 751–767
- A. Elgammal, R. Duraiswami, D. Harwood & L.S. Davis (2002). *Background and foreground modeling using nonparametric kernel density estimation for visual surveillance*. In Proceedings of the IEEE, vol.90, no.7, pp.1151–1163

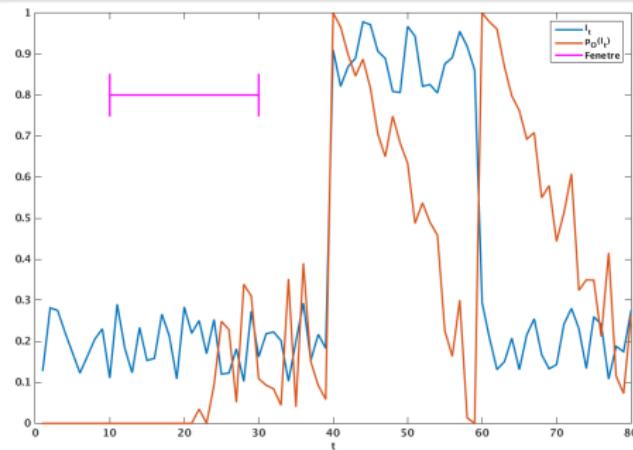
# Estimateur de densité par noyaux

## Choix de $n$

Compromis à trouver entre

- une bonne modélisation du fond
- l'inclusion de l'objet dans le fond ⇒
  - effet de latence (déttection du fond comme d'un objet)
  - non détection de l'objet
- une augmentation des temps de calcul

**Solution :** Considérer  $n$  frames parmi les  $N$  précédentes (avec  $N \gg n$ )



# Estimateur de densité par noyaux

## Résultats



# Estimateur de densité par noyaux

## Résultats



# Décomposition en valeurs singulières (ACP)

## Principe

Appliquer une **analyse en composantes principales (ACP)** sur une séquence de  $n$  images pour obtenir des *eigenbackground*

- Les parties **statiques** sont **mieux représentées** dans les sous-espaces propres
- Objets mobiles : faible proportion de pixels et ne couvrant pas les mêmes pixels sur les  $n$  images
  - ⇒ peu représentés dans le sous-espace propre

## Référence

N. Oliver, B. Rosario & A. Pentland (2000), *A Bayesian computer vision system for modeling human interactions*. IEEE Trans. Pattern Analysis and Machine Intelligence, vol.22, no.8, pp.831–843

# Décomposition en valeurs singulières (ACP)

## Apprentissage

- ① Soit un ensemble d'image  $\{I_1, \dots, I_n\}$ , avec  $\mu_B$  l'image moyenne
- ② Soit  $A$  la matrice dont **chaque colonne est une image centrée**  $\tilde{I}_i = I_i - \mu_B$
- ③ Calculer la **matrice de covariance**  $C_B = AA^T$
- ④ Décomposition en valeurs singulières :  $L_B = \Phi_B C_B \Phi_B^T$   
avec  $L_B$  la matrice diagonale des **valeurs propres** et  $\Phi_B$  la matrice contenant les **vecteurs propres** (*eigenbackground*) de  $C_B$
- ⑤ Garder les  $M$  principaux *eigenbackgrounds* (*i.e.* associés aux  $M$  **plus grandes valeurs propres**) :  $\Phi_M$  est la **matrice de projection** dans le sous-espace propre

# Décomposition en valeurs singulières (ACP)

## Choix de $M$

- On associe à chaque valeur propre une énergie (quantité d'information qu'elle représente) :  $E_i = \lambda_i^2$
- On classe les valeurs propres par ordre décroissant d'énergie :  $\lambda_1^2 > \lambda_2^2 > \dots > \lambda_m^2$ , où  $m$  est le nombre de pixels dans l'image
- On cherche à conserver  $s\%$  de l'énergie totale  $E = \sum_{i=1}^m \lambda_i^2$
- $M$  est le premier entier tel que :  $\sum_{i=1}^M \lambda_i^2 > sE$

# Décomposition en valeurs singulières (ACP)

Détection des pixels du fond d'une nouvelle image  $I_t$

- ① **Projeter**  $I_t$  dans le sous-espace de représentation :  $B_t = \Phi_M^T(I_t - \mu_B)$
- ② **Reconstruire** l'image projetée :  $B_t^r = \mu_B + \Phi_M B_t$
- ③ **Calculer la distance** entre l'image reconstruite et l'originale  $|I_t - B_t|$  (distance euclidienne)
- ④ Les pixels des objets sont tels que  $|I_t - B_t| > S$  ( $S$  à fixer)

# Décomposition en valeurs singulières (ACP)

## Remarque

- L'apprentissage peut être très lourd en terme de mémoire et temps de calcul.
- Par exemple, pour 100 images de  $512 \times 512$  (soit 262144 éléments), il faut :
  - ➊ faire le produit d'une matrice  $262144 \times 100$  avec sa transposée
  - ➋ décomposer en valeurs et vecteurs propres une matrice de  $262144 \times 262144$
- En théorie, il faudrait refaire cette étape à chaque instant  $t$  pour tenir compte de la nouvelle image (mouvement de la caméra, changement d'illumination, ...)

# Décomposition en valeurs singulières (ACP)

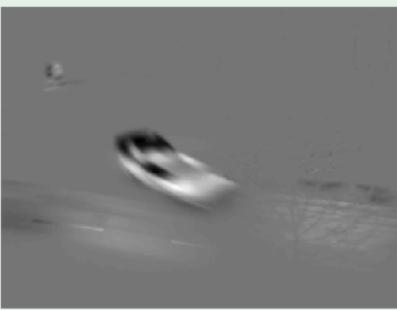
## Exemple

- On considère toute la séquence pour l'apprentissage (41 images)
- On conserve les  $M$  eigenbackgrounds correspondant à 90% de l'énergie  $E$   
 $\Rightarrow M = 2$

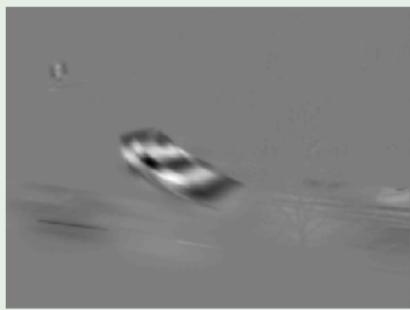
## Apprentissage



Image moyenne



Eigenbackground ( $\lambda_1$ )



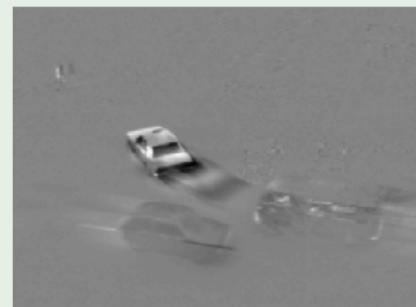
Eigenbackground ( $\lambda_2$ )

# Décomposition en valeurs singulières (ACP)

## Exemple

- On considère toute la séquence pour l'apprentissage (41 images)
- On conserve les  $M$  eigenbackgrounds correspondant à 90% de l'énergie  $E$   
 $\Rightarrow M = 2$
- On reprojette les images de la séquence sur l'espace appris.

## Détection

 $I_t$  $B_t^r$  $I_t - B_t^r$

# Décomposition en valeurs singulières (ACP)

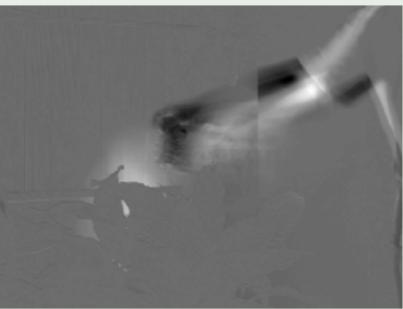
## Exemple

- On considère les 100 premières images d'une séquence pour l'apprentissage
- On conserve les  $M$  eigenbackgrounds correspondant à 90% de l'énergie  $E$   
 $\Rightarrow M = 2$

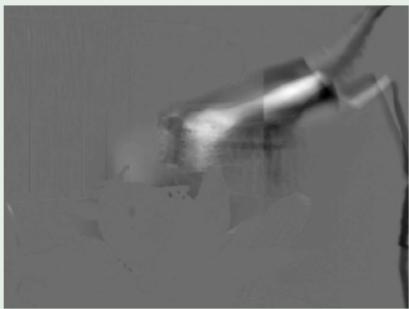
## Apprentissage



Image moyenne



Eigenbackground ( $\lambda_1$ )



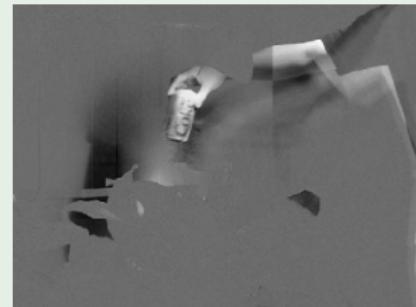
Eigenbackground ( $\lambda_2$ )

# Décomposition en valeurs singulières (ACP)

## Exemple

- On considère les 100 premières images d'une séquence pour l'apprentissage
- On conserve les  $M$  eigenbackgrounds correspondant à 90% de l'énergie  $E$   
 $\Rightarrow M = 2$
- On projette les images suivantes sur l'espace appris.

## Détection

 $I_t$  $B_t^r$  $I_t - B_t^r$

## Conclusions

- De nombreuses méthodes, mais problème encore ouvert
- Etape très importante : une mauvaise détection entraîne un mauvais suivi

## Limitations

- Un post-traitement est souvent nécessaire pour enlever le bruit (morphologie mathématique, ...)
- Problème des ombres portées (détectées avec l'objet)



Post-traitement



Ombre portée

