

# Détection et suivi d'objets en mouvement dans des scènes complexes : Application à la surveillance des conducteurs

Aurélie Bugeau

Équipe Vista, IRISA / INRIA

20 décembre 2007

# Introduction générale

## Contexte

ACI (Action concertée Initiative) Behaviour sur la surveillance du conducteur pour la détection de l'endormissement

- Laboratoire HEUDIASYC de l'UTC Compiègne :  
Suivi de visage et analyse du comportement
- Équipe VISTA :  
Détection et suivi simultanés de parties du corps pour l'analyse des trajectoires

### Difficulté rencontrée

Complexité des vidéos prises dans des conditions réelles



# Vidéos complexes

## Difficultés des vidéos

- Fond dynamique
- Changements d'illumination
- Faibles contrastes (couleur et mouvement)
- Caméra mobile
- Vidéos en couleur ou en niveau de gris

Difficultés génériques : nous ne nous concentrons pas exclusivement sur les vidéos de conducteurs  
⇒ Pas d'*a priori* sur les objets à détecter et à suivre



# Problématique et plan de la présentation

Comment détecter et suivre les objets en mouvement ?

## Algorithme proposé

- 1 Détection des objets en mouvement  
repose sur le partitionnement des points en régions cohérentes (clusters)
- 2 Suivi des objets en mouvement

# Problématique et plan de la présentation

Comment détecter et suivre les objets en mouvement ?

## Algorithme proposé

- 1 Détection des objets en mouvement  
repose sur le partitionnement des points en régions cohérentes (clusters)
- 2 Suivi des objets en mouvement

## Plan de la présentation

- 1 Partitionnement des points en régions cohérentes
- 2 Détection des objets en mouvement
- 3 Suivi des objets en mouvement

## Première partie

# Partitionnement des points en régions cohérentes

## Mean shift

- Clustering de points pour des données de grandes dimensions : problème ouvert en vision par ordinateur
- Nombre et forme des clusters inconnus → mean shift

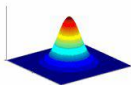
Principe :

- Trouver les modes de la densité  $f$  (inconnue) d'un nuage de points  $\{\mathbf{x}^{(i)}\}_{i=1\dots n}$
- Estimation de la densité avec un estimateur à noyau

$$\hat{f}(\mathbf{x}) = \frac{1}{n|\mathbf{H}|^{1/2}} \sum_{i=1}^n \underbrace{K}_{\text{Noyau}}(\mathbf{H}^{-1/2}(\mathbf{x} - \mathbf{x}^{(i)})) \quad \text{avec } (\mathbf{H} : \text{"Taille" du noyau})$$

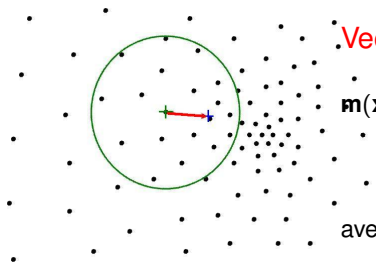
- Exemple de noyau : noyau gaussien

$$K(\mathbf{x}) = ce^{(-\|\mathbf{x}\|^2/2)}$$



# Mean shift

- Principe : trouver les modes (maxima locaux)
- $\nabla \hat{f}(\mathbf{x}) = \mathbf{H}^{-1} \hat{f}(\mathbf{x}) \mathbf{m}(\mathbf{x})$
- Montée de gradient en ne calculant que  $\mathbf{m}(\mathbf{x})$



Vecteur mean shift :

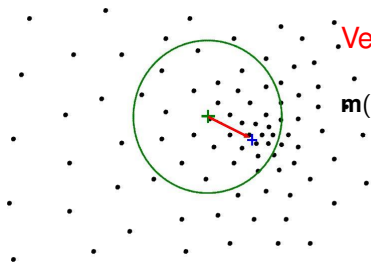
$$\mathbf{m}(\mathbf{x}) = \frac{\sum_{i=1}^n \mathbf{x}^{(i)} k(\|\mathbf{H}^{-1/2}(\mathbf{x} - \mathbf{x}^{(i)})\|^2)}{\underbrace{\sum_{i=1}^n k(\|\mathbf{H}^{-1/2}(\mathbf{x} - \mathbf{x}^{(i)})\|^2)}_{\text{Point de trajectoire } \mathbf{y}^{(1)}}} - \mathbf{x}$$

avec  $K(x) = c_k k(\|x\|^2)$

# Mean shift

Principe : montée de gradient itérative.

$\mathbf{y}^{(1)}$  devient le nouveau point d'estimation



Vecteur mean shift :

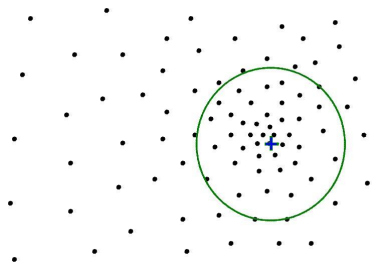
$$\mathbf{m}(\mathbf{y}^{(1)}) = \frac{\sum_{i=1}^n \mathbf{x}^{(i)} k(\|\mathbf{H}^{-1/2}(\mathbf{y}^{(1)} - \mathbf{x}^{(i)})\|^2)}{\underbrace{\sum_{i=1}^n k(\|\mathbf{H}^{-1/2}(\mathbf{y}^{(1)} - \mathbf{x}^{(i)})\|^2)}_{\text{Point de trajectoire } \mathbf{y}^{(2)}}} - \mathbf{y}^{(1)}$$

# Mean shift

Principe : montée de gradient itérative.

Continuer jusqu'à convergence, c'est-à-dire

$$\mathbf{y}^{(k)} = \mathbf{y}^{(k-1)}$$

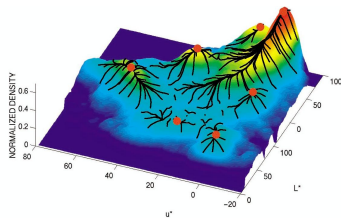
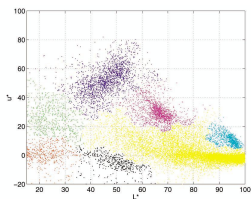
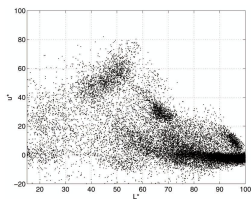


$\mathbf{y}^{(k)}$  est le mode local  
de la densité

- └ Partitionnement des points en régions cohérentes
- └ Principe du clustering mean shift

## Clustering mean shift

Regrouper les points ayant convergé vers le même mode.



[Comaniciu'02]

Permet de diviser les données en plusieurs clusters.

- └ Partitionnement des points en régions cohérentes
- └ Mean shift avec des estimateurs à noyau variable

## Estimateurs à noyau variable

Noyau fixe pas adapté à des données hétérogènes.

⇒ Estimateur à noyau variable

### 1 Estimateur "sample point"

$$\hat{f}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{|\mathbf{H}(\mathbf{x}^{(i)})|^{1/2}} K(\mathbf{H}(\mathbf{x}^{(i)})^{-1/2}(\mathbf{x} - \mathbf{x}^{(i)}))$$

Application au clustering mean shift :

"Variable bandwidth mean shift" [Comaniciu'01]

- └ Partitionnement des points en régions cohérentes
- └ Mean shift avec des estimateurs à noyau variable

## Estimateurs à noyau variable

### 2 Estimateur "balloon"

$$\hat{f}(\mathbf{x}) = \frac{1}{n|\mathbf{H}(\mathbf{x})|^{1/2}} \sum_{i=1}^n K(\mathbf{H}(\mathbf{x})^{-1/2}(\mathbf{x} - \mathbf{x}^{(i)}))$$

Meilleurs résultats pour des données de dimension supérieure à 3 [Terrell'92].

## Pseudo balloon mean shift

- Problème :  $\nabla \hat{f}(\mathbf{x})$  dépend de  $(\mathbf{x} - \mathbf{x}^{(i)})^2$  et  $\mathbf{H}'(\mathbf{x})$
- Sous les hypothèses :
  - $\forall i = 1..n \quad \mathbf{H}'(\mathbf{x}^{(i)}) = 0$
  - $\mathbf{H}(\mathbf{y}^{(j)}) = \mathbf{H}(\mathbf{x}^{(i)})$  (constant pour tous les points de trajectoire démarrant au point  $\{\mathbf{x}^{(i)}\}, i = 1..n$ )
- Un vecteur mean shift peut être obtenu :

$$\mathbf{m}(\mathbf{x}) = \frac{\sum_{i=1}^n \mathbf{x}^{(i)} k(\|(\mathbf{H}(\mathbf{x}))^{-1/2}(\mathbf{x} - \mathbf{x}^{(i)})\|^2)}{\sum_{i=1}^n k(\|(\mathbf{H}(\mathbf{x}))^{-1/2}(\mathbf{x} - \mathbf{x}^{(i)})\|^2)} - \mathbf{x}$$

- Comment définir le noyau permettant de respecter les hypothèses ?

## Taille du noyau

- Méthodes existantes :
  - Méthodes statistiques
  - Méthodes basées sur la stabilité des clusters
- Ne sont pas adaptées à des données multidimensionnelles hétérogènes

*i.e.*  $\mathbf{x}^{(i)}$  de dimension  $d \rightarrow \mathbf{x}^{(i)} = (\mathbf{x}_1^{(i)}, \dots, \mathbf{x}_P^{(i)})$ ,  
*e.g.* position, couleur, mouvement ...

- Méthodes sur la stabilité : tous les points d'un même voisinage seront encouragés à être associés à la même taille de noyau

⇒ Méthode itérative, inspirée de [Comaniciu'03], basée sur la stabilité des clusters

## Sélection de la taille du noyau

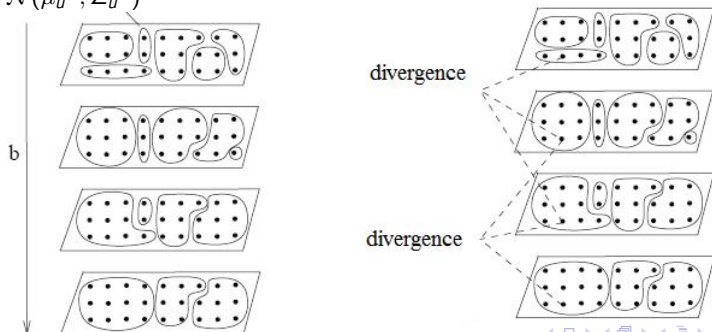
Pour chaque point  $i = 1 \dots n$  et chaque espace  $p = 1, \dots, P$

Choisir le noyau parmi un ensemble prédéfini  $\mathbf{H}_p = \{\mathbf{H}_p^{(b)}\}_{b=1, \dots, B_p}$

Meilleur noyau  $\Upsilon^{(i)} =$  noyau donnant les clusters les plus stables

2 étapes pour chaque espace  $p = 1, \dots, P$

$$\mathcal{N}(\mu_u^{(1)}, \Sigma_u^{(1)})$$



## Sélection itérative

- Méthode appliquée successivement à chaque espace de caractéristiques ( $p = 1, \dots, P$ )
- pour trouver  $\Upsilon_p$  en chaque point
- $\forall b = 1, \dots, B_p$ , le noyau prédéfini utilisé est :

$$\text{diag}[\Upsilon_1(\mathbf{x}^{(i)}), \dots, \Upsilon_{p-1}(\mathbf{x}^{(i)}), \mathbf{H}_p^{(b)}, \tilde{\mathbf{H}}_{p+1}, \dots, \tilde{\mathbf{H}}_P]$$

$$\text{avec } \tilde{\mathbf{H}}_q = \frac{1}{B_q} \sum_{b=1}^{B_q} \mathbf{H}_q^{(b)}, q > p$$

## Validation de l'approche itérative

Pour tous les espaces, 9 noyaux prédéfinis, entre 10 et 30.

Sélection non itérative : 1 espace ( $P = 1$ ) de dimension 5

Sélection itérative : 5 espaces ( $P = 5$ ) de dimension 1

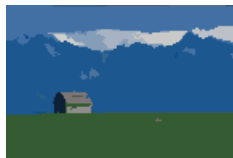


Image originale

Sélection non itérative

Sélection itérative

- └ Partitionnement des points en régions cohérentes
- └ Application à la segmentation d'images couleur

## Validation du clustering pseudo balloon mean shift

Sélection itérative des noyaux puis clustering mean shift

avec l'estimateur sample point [Comaniciu'01]

avec l'estimateur balloon



Image originale

Sample point

Balloon

## Conclusion sur le partitionnement

### Conclusion

- Nouveau mean shift basé sur l'estimateur balloon
- Algorithme itératif de sélection de la taille du noyau
- Validation qualitative dans le contexte de la segmentation d'images couleur

### Perspectives

- Étude théorique approfondie de la convergence du pseudo-balloon mean shift
- Étude théorique de la validité des hypothèses avec les noyaux calculés par la méthode itérative proposée
- Meilleure représentation des clusters pour le calcul de la divergence dans l'algorithme de sélection.

## Deuxième partie

# Détection des objets en mouvement

# Présentation de l'approche proposée

## 4 catégories pour la détection/segmentation de mouvement

- Détection basée sur la différence inter-images  
⇒ caméra fixe ou compensation de mouvement, fond fixe, peu de changements d'illumination, masque binaire
- Modélisation du fond  
⇒ caméra fixe, fond peu dynamique, masque binaire
- Extraction de couches de mouvement  
⇒ régions caractérisées par un modèle paramétrique de mouvement
- Détection de mouvement basée sur la cohérence des trajectoires  
⇒ objets se déplacent dans une direction pendant plusieurs pas de temps

# Présentation de l'approche proposée

Points importants issus de l'analyse des méthodes existantes :

- Combiner des informations de mouvement et de photométrie [Mittal'04].
- Possible d'utiliser un nombre restreint de points.  
Regrouper les points en régions cohérentes avec une méthode de clustering.

# Présentation de l'approche proposée

## Algorithme proposé

- 1 Sélection et description de points en mouvement  
(pour travailler sur un nombre restreint de points)
- 2 Partitionnement des points en régions cohérentes  
(cohérence dépendant du mouvement et de la couleur)
- 3 Segmentation finale de tous les objets mobiles  
(obtenir le masque complet de chaque objet)

## Détection des pixels en mouvement

- Relativement au mouvement de la caméra
- Mouvement de la caméra = mouvement dominant (affine)  
estimé avec la méthode de [Odobez'95]  
carte  $W_t$  : participation de chaque pixel à l'estimation



Image 107



Image 108

 $W_{108}$ 

- $M_t$  : carte finale binaire des pixels en mouvement

 $W_{107}$  $W_{108}$  $M_{108}$

## Grille de points

Pas d'*a priori* sur la forme ou la texture des objets :  
→ grille de pixels en mouvement uniformément répartis.

$$G = \left\{ s = \left( \frac{k \cdot w}{N_G}, \frac{l \cdot h}{N_G} \right), k = 0 \cdots N_G, l = 0 \cdots N_G \mid M_t(s) = 1 \right\},$$

avec :

- $w$  : largeur de l'image
- $h$  : hauteur
- $N_G$  : paramètre définissant la taille de la grille  
calculé en fonction du nombre de pixels en mouvement dans l'image et  
du nombre de points voulu (500).

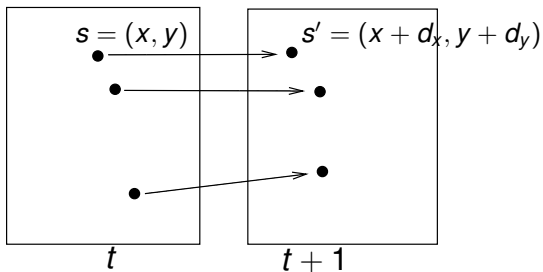
## Description des points

### Objet en mouvement

Région de l'image ayant une couleur et un mouvement quasi constants

⇒ Caractéristiques : position, mouvement, photométrie

# Notations



$$\mathbf{z}_t(\mathbf{s}) = \left( \underbrace{\mathbf{z}_t^{(G)}(\mathbf{s})}_{\text{intensité}}, \underbrace{\mathbf{z}_t^{(C)}(\mathbf{s})}_{\text{couleur}}, \underbrace{\mathbf{z}_t^{(M)}(\mathbf{s})}_{\text{mouvement}} \right)$$

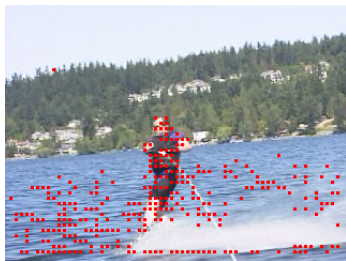
## Caractéristiques de mouvement

- Vecteurs de mouvement  $\mathbf{z}_t^{(M)}(s) = (d_x, d_y)$  calculés par flot optique [Lucas'81]
- Problèmes :
  - Le calcul du flot optique se base sur 2 hypothèses :
    - conservation de la luminance
    - mouvement quasi constant autour d'un point
 Validation des vecteurs avec un **test statistique** (p-value)
  - Résolution de l'équation du flot optique n'est possible que pour des voisinages de luminance non constante

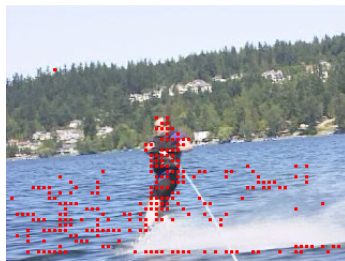
- $\Rightarrow$  nouvelle grille :

$$G = \left\{ s = \left( \frac{k \cdot w}{N_G}, \frac{l \cdot h}{N_G} \right) \mid M_t(s) = 1 \ \& \ \exists (x_i, y_i) \in V(s), |\nabla I(x_i, y_i, t)| \neq 0 \right. \\ \left. \ \& \ \text{p-value}(s, s') < 0.05 \right\}.$$

## Exemple de grille



Grille initiale



Validation des vecteurs de mouvement

## Caractéristiques photométriques

- Espace colorimétrique YUV :  $\mathbf{z}_t^{(C)}(s) = (Y_t(s), U_t(s), V_t(s))$
- moyenne sur le voisinage du point :  $\bar{\mathbf{z}}_t^{(C)}(s)$
- cohérence temporelle :  $\bar{\mathbf{z}}_{t+1}^{(C)}(s')$

⇒ Descripteur en chaque point d'indice  $i$  de la grille :

$$\mathbf{x}^{(i)} = (\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \mathbf{x}_3^{(i)}),$$

avec

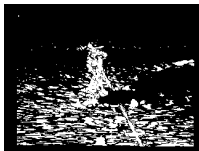
$$\mathbf{x}_1^{(i)} = (x, y), \quad \mathbf{x}_2^{(i)} = (d_x, d_y), \quad \mathbf{x}_3^{(i)} = (\bar{\mathbf{z}}_t^{(C)}(s), \bar{\mathbf{z}}_{t+1}^{(C)}(s'))$$

## Partitionnement des points

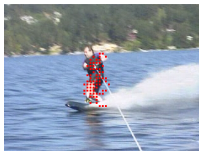
- Pseudo balloon mean shift
- Sélection automatique de la taille du noyau  
les noyaux prédéfinis sont calculés automatiquement en fonction des caractéristiques de l'image
- ⇒ **Clusters en mouvement**, noté  $\mathcal{C}_{u,t}$
- Ne garder que les clusters contenant plus de 15 pixels



Image



Pixels en mouvement



Clusters en mouvement

## Segmentation des objets à partir des clusters

- Utilisation d'une méthode de coupe dans un graphe sur tous les pixels de l'image

Avantages des Graph Cuts (coupe minimale/flot maximal) :

- Permettent de trouver le minimum global de certaines fonctions d'énergie
  - Faible coût de calcul
- Segmentation indépendante de chaque objet  $u$   
*i.e.* une fonction d'énergie par cluster
  - Étiquetage de chaque pixel  $s$  de l'image :  
 $I_s \in \{\text{fond, objet}\}, L = \{I_s, s \in \mathcal{P}\}$
  - Validation des clusters :  
si la segmentation finale est identique au cluster, le cluster est non valide

$$E_t(L) = \underbrace{-\gamma_c \sum_{s \in \mathcal{P} \setminus \mathcal{C}_{u,t}} \ln(p(\mathbf{z}_t^{(C)}(s)|I_s)) - \gamma_m \sum_{s \in \mathcal{G} \setminus \mathcal{C}_{u,t}} \ln(p(\mathbf{z}_t^{(M)}(s)|I_s))}_{\text{Terme d'attache aux données (GMM sur la couleur et le mouvement)}}$$

Terme d'attache aux données (GMM sur la couleur et le mouvement)

$$- \underbrace{\sum_{s \in \mathcal{C}_{u,t}} \left(1 + \max_{s' \in \mathcal{P}} \lambda \sum_{r | (s', r) \in \mathcal{E}_{\mathcal{P}}} B_{\{s', r\}}\right) \delta(I_s, \text{"objet"})}_{\text{Forcer les points appartenant à un cluster à appartenir à l'objet}}$$

Forcer les points appartenant à un cluster à appartenir à l'objet

$$+ \underbrace{\lambda \sum_{(s,r) \in \mathcal{E}_{\mathcal{P}}} B_{\{s,r\}}(I_s, I_r) \delta(I_s, I_r)}_{\text{Terme de lissage}}$$

Terme de lissage  $B_{\{s,r\}}(I_s, I_r) = \frac{1}{\|s-r\|} \exp\left(-\frac{\|\mathbf{z}_t^{(G)}(s) - \mathbf{z}_t^{(G)}(r)\|^2}{\sigma^2}\right)$  [Boykov'01]

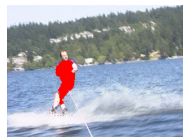
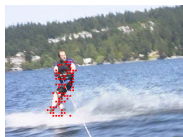
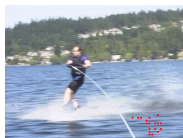
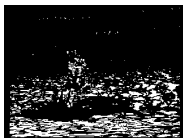
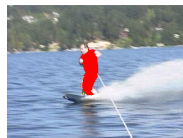
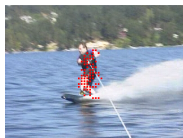
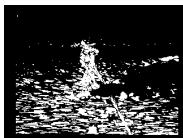
avec

$$\delta(I_s, I_r) = \begin{cases} 1 & \text{si } I_s \neq I_r \\ 0 & \text{sinon} \end{cases}$$

Énergie minimisée avec un algorithme de coupe minimale/flot maximal

[Boykov'04]

## Résultats : séquence de ski nautique



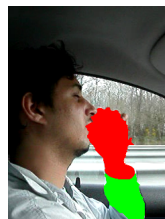
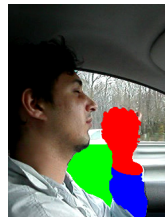
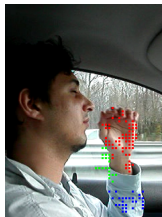
Images

Pixels en mouvement

Clusters en mouvement

Objets détectés

## Résultats : séquence de conducteur



Images

Pixels en mouvement

Clusters en mouvement

Objets détectés

## Comparaison avec d'autres méthodes



Images originales



Modélisation de fond  
(mélange de gaussiennes) [Grimson'98]



Modélisation du fond  
(non paramétrique) [Elgammal'00]



Pixels en mouvement



Notre méthode

# Conclusion sur la détection d'objets

## Conclusion :

- Détection et segmentation d'objets en mouvement dans des scènes complexes
- Détection sur un nombre restreint de points
- Informations de couleur et de mouvement

## Limites et perspectives :

- Étude quantitative de l'influence de la p-value
- Énergie multi-étiquettes pour la segmentation
- Détecter les petits objets
- Détecter les objets appartenant au mouvement dominant
- Introduire de la cohérence temporelle

## Troisième partie

# Suivi des objets en mouvement

## Méthodes existantes

3 catégories pour le suivi d'objets :

- Suivi par appariement de détection
  - + gèrent l'entrée de nouveaux objets
  - dépendent de la qualité des détections
- Suivi déterministe par détection séquentielle
  - + suivi robuste avec un faible coût de calcul
  - ne gèrent pas l'entrée de nouveaux objets
- Segmentation dynamique
  - + segmentation
  - ne gèrent pas l'entrée de nouveaux objets

# Contraintes

- Segmentation et suivi temporel des objets en mouvement
- Inclure des observations venant d'un module extérieur
- Gérer l'entrée et la sortie de nouveaux objets du champ de la caméra

# Observations

Masques de pixels :

- Obtenus par soustraction de fond



Image de référence

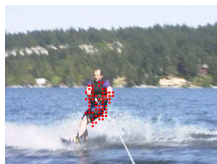


Image courante

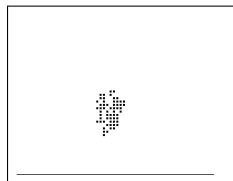


Observations

- Clusters en mouvement



Clusters détectés



Observations

## 2 étapes

- 1 Suivi indépendant de chaque objet
- 2 Séparation des objets ayant fusionné

Reposent sur des minimisations d'énergies par Graph Cuts

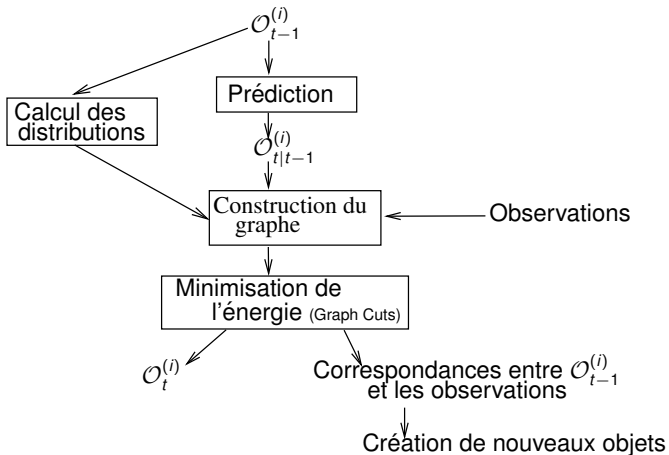
## Principe de la méthode de suivi

- Chaque objet est suivi indépendamment  
(un graphe et une fonction d'énergie par objet)
- Suivi et segmentation de l'objet  $i$  ( $i = 1 \dots k_t$ ), noté  $\mathcal{O}_t^{(i)}$ , repose sur
  - des distributions (de couleur et de mouvement)
  - la prédiction de l'objet :

$$\mathcal{O}_{t|t-1}^{(i)} = \left\{ \mathbf{s} + \frac{\sum_{r \in \mathcal{O}_{t-1}^{(i)}} \mathbf{z}_{t-1}^{(M)}(r)}{|\mathcal{O}_{t-1}^{(i)}|}, \mathbf{s} \in \mathcal{O}_{t-1}^{(i)} \right\}$$

- $m_t$  observations externes  
 $j^{\text{eme}}$  observation notée  $\mathcal{M}_t^{(j)}$
- Assigner une étiquette ("objet" ou "fond") à tous les pixels

## Principe de la méthode de suivi

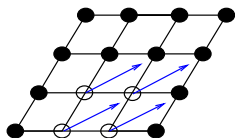


# Graphe

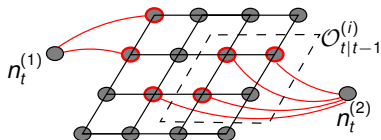
Graphe non orienté  $G_t = (\mathcal{V}_t, \mathcal{E}_t)$

- $\mathcal{V}_t = \mathcal{P} \cup_{j=1}^{m_t} n_t^{(j)}$  : ensemble de nœuds
  - $\mathcal{P}$  grille de pixels  
appelé **nœud pixel**
  - $n_t^{(j)}$  : nœud associé à l'observation  $\mathcal{M}_t^{(j)}$   
appelé **nœud d'observation**
- $\mathcal{E}_t = \mathcal{E}_{\mathcal{P}} \cup_{j=1}^{m_t} \mathcal{E}^{(j)}$  : ensemble d'arcs (ou arêtes)
  - $\mathcal{E}_{\mathcal{P}}$  : paires  $\{s, r\}$  de pixels voisins de  $\mathcal{P}$
  - $\mathcal{E}^{(j)}$  : paires  $\{s, n_t^{(j)}\}$ , avec  $s \in \mathcal{M}_t^{(j)}$

## Grappe



Objet  $i$  à l'instant  $t - 1$



Grappe pour l'objet  $i$  à l'instant  $t$

Un objet peut correspondre à plusieurs observations

Une observation peut correspondre à plusieurs objets

- ↳ Suivi des objets en mouvement

- ↳ Suivi indépendant de chaque objet

## Fonction d'énergie

### Étiquettes

- Assigner une étiquette ("fond" ou "objet") à chaque nœud :
- *i.e.* assigner
  - $I_{s,t}^{(i)}$  en chaque pixel  $s$
  - $I_{j,t}^{(i)}$  pour chaque nœud d'observation  $n_t^{(j)}$
- *i.e.* trouver  $L_t^{(i)} = \{I_{s,t}^{(i)}, \forall s \in \mathcal{V}_t\}$

### Fonction d'énergie

En omettant les indices  $i$  et  $t$  :

$$E(L) = \underbrace{\sum_{s \in \mathcal{V}} R(s, I_s)}_{\text{terme d'attache aux données}} + \underbrace{\sum_{\{s,r\} \in \mathcal{E}} B(s,r) \delta(I_s, I_r)}_{\text{terme de lissage : pénalise les discontinuités}}$$

## Terme d'attache aux données

$$\sum_{s \in \mathcal{V}} R(s, l_s) = \sum_{s \in \mathcal{O}_{t|t-1}^{(i)}} -\ln(p_1(s, l_s)) + \alpha \sum_{j=1}^{m_t} d_2(j, l_{j,t})$$

### Nœuds correspondant à la prédiction

- $\mathcal{O}_t^{(i)}$  doit être similaire à  $\mathcal{O}_{t-1}^{(i)}$  pour la couleur et le mouvement
- $p_1$  : distribution de couleur et mouvement dépendant de l'étiquette
  - GMM calculé sur  $\mathcal{O}_{t-1}$  pour  $l =$  "objet"
  - GMM calculé sur  $\mathcal{P} \setminus \mathcal{O}_{t-1}^{(i)}$  pour  $l =$  "fond"
  - Mélanges de gaussiennes calculés avec un algorithme EM
  - Remis à jour à chaque instant

## Terme d'attache aux données

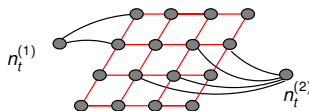
$$\sum_{s \in \mathcal{V}} R(s, l_s) = \sum_{s \in \mathcal{O}_{t|t-1}^{(i)}} -\ln(p_1(s, l_s)) + \alpha \sum_{j=1}^{m_t} d_2(j, l_{j,t})$$

### Nœuds d'observation

- Observation utilisée que si elle est similaire à  $\mathcal{O}_{t-1}^{(i)}$
- $d_2(s, l)$  : distance de Kullback-Leibler entre
  - GMM de l'objet et GMM de l'observation si  $l = \text{"objet"}$
  - GMM du fond et GMM de l'observation si  $l = \text{"fond"}$
- GMM de l'observation,  $\rho_t^{(j)}$ , calculé sur  $\{\mathbf{z}_t(s)\}_{s \in \mathcal{M}_t^{(j)}}$

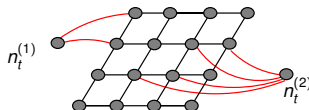
## Terme de lissage

- Arcs entre pixels, paires  $\{s, r\}$  de pixels voisins de  $\mathcal{P}$  :  
Basé sur les gradients de couleur [Boykov'01]



$$B_{\{s,r\}} = \lambda_1 \frac{1}{\|s - r\|} e^{-\frac{\|z_t^{(C)}(s) - z_t^{(C)}(r)\|^2}{\sigma_T^2}}$$

- Arcs d'observation, paires  $\{s, n_t^{(j)}\}$ , avec  $s \in \mathcal{M}_t^{(j)}$  :



$$B_{\{s, n_t^{(j)}\}} = \lambda_2 \rho_t^{(j)}(z_t^{(C)}(s))$$

## Minimisation et création de nouveaux objets

Énergie minimisée avec un algorithme de coupe minimale/flot maximal

[Boykov'04]

Étiquetage final :  $\hat{L}_t^{(i)} = \arg \min_{L_t^{(i)}} E_t^{(i)}(L_t^{(i)})$ .

Mise à jour des objets :  $\mathcal{O}_t^{(i)} = \{s \in \mathcal{P} : \hat{l}_{s,t}^{(i)} = \text{"objet"}\}$ .

### Création de nouveaux objets :

- Une observation  $j$  est associée à  $\mathcal{O}_{t-1}^{(i)}$  si  $\hat{l}_{t,j}^{(i)} = \text{"objet"}$
- Une nouvelle piste de suivi est initialisée pour l'observation  $j$  si  $\forall i = 1 \dots k_t, \hat{l}_{t,j}^{(i)} = \text{"bg"}$   
en définissant

$$\mathcal{O}_t^{(k_t)} = \mathcal{M}_t^{(j)}$$

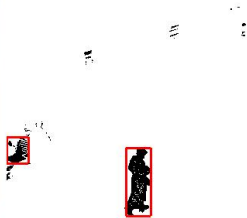
En pratique, un objet n'est créé que s'il a été associé à au moins une observation à l'instant suivant.

- └ Suivi des objets en mouvement
- └ Suivi indépendant de chaque objet

## Soustraction de fond



Images originales



Observations

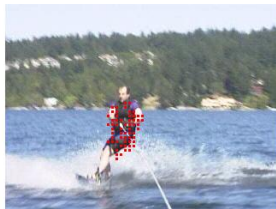


Résultats du suivi

## Clusters en mouvement



Images originales



Observations



Résultats du suivi

## Fusion d'objets

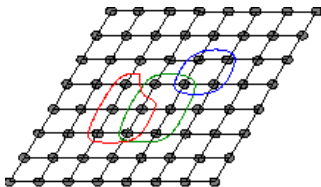


- si  $\cap_{i \in \mathcal{F}} \mathcal{O}_t^{(i)} \neq \emptyset$ ,  
avec  $\mathcal{F}$  : ensemble des indices des objets suivis
- Chaque pixel ne doit appartenir qu'à un seul objet.
- Solution proposée : deuxième minimisation d'énergie

## Deuxième fonction d'énergie

Graphe  $\tilde{G}_t = (\tilde{\mathcal{V}}_t, \tilde{\mathcal{E}}_t)$

- nœuds : pixels des objets ayant fusionné,  $\tilde{\mathcal{V}}_t = \cup_{i \in \mathcal{F}} \mathcal{O}_t^{(i)}$
- $\tilde{\mathcal{E}}_t$  : paires de nœuds voisins de  $\tilde{\mathcal{V}}_t$



Résultat du suivi  
3 objets ont fusionné



Forme du graphe correspondant

- Assigner à chaque nœud  $s \in \tilde{\mathcal{V}}_t$  une étiquette  $\psi_s \in \mathcal{F}$ ,

$$\tilde{\mathcal{L}} = \{\psi_s, s \in \tilde{\mathcal{V}}_t\}$$

# Énergie multi-étiquettes

$$\tilde{E}_t(\tilde{L}) = \sum_{s \in \tilde{\mathcal{V}}_t} -\ln(p_3(s, \psi_s)) + \lambda_3 \sum_{\{s, r\} \in \tilde{\mathcal{E}}_t} \frac{1}{\|s - r\|} e^{-\frac{\|z_t^{(C)}(s) - z_t^{(C)}(r)\|^2}{\sigma_3^2}} \delta(\psi_s, \psi_r)$$

## Distribution :

- La couleur et le mouvement ne sont pas suffisants pour distinguer les objets
- $\Rightarrow$  Faire confiance à la prédiction

$$\text{si } s \in \mathcal{O}_{t|t-1}^{(\psi)}, p_3(s, \psi) = 1$$



# Soustraction de fond

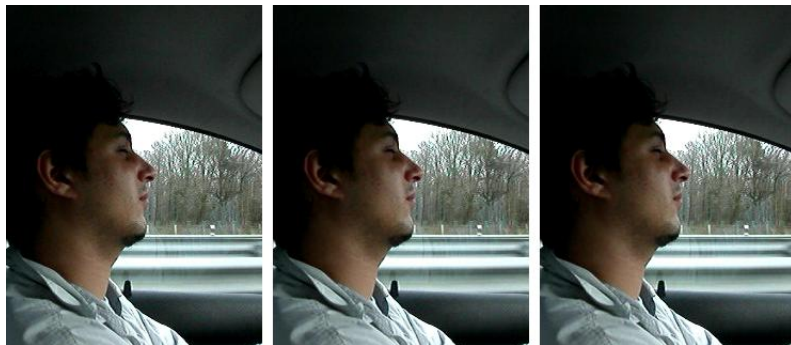


- └ Suivi des objets en mouvement
- └ Séparation des objets fusionnés

## Clusters en mouvement



## Clusters en mouvement



## Conclusion du suivi

### Conclusion :

- Segmentation et suivi simultanés d'objets
- Association des objets suivis avec les observations
- Second graphe : suivi indépendant de chaque objet
- Permet de segmenter et suivre les clusters en mouvement détectés dans les scènes complexes

### Perspectives :

- Regrouper les deux fonctions d'énergie
- Réglage automatique des paramètres
- Ajout d'autres caractéristiques (*e.g.* la forme)

# Conclusion générale

# Synthèse

- Détection des objets en mouvement
  - Sélection et description d'un nombre restreint de points
  - Partitionnement avec un nouvel algorithme mean shift et une sélection automatique de la taille du noyau
  - Segmentation par Graph Cuts
- Suivi des objets par Graph Cuts
  - en combinant la prédiction, des distributions de couleur et de mouvement et des observations extérieures
  - association automatique pistes/observations
- Bons résultats dans des vidéos complexes

## Nombreuses perspectives

- Validation quantitative de toutes les méthodes proposées
- Nouvel algorithme mean shift
  - Problèmes théoriques de convergence
- Détection des objets
  - Détection des petits objets / mouvement dominant
- Suivi des objets
  - Regrouper les deux fonctions d'énergie
- Graph Cuts
  - Réglage automatique des paramètres
  - Ajout de termes globaux dans la fonction d'énergie