

Analyse de la vidéo

Chapitre 4.2 - Suivi d'objet

10 mars 2015

Plan de la présentation

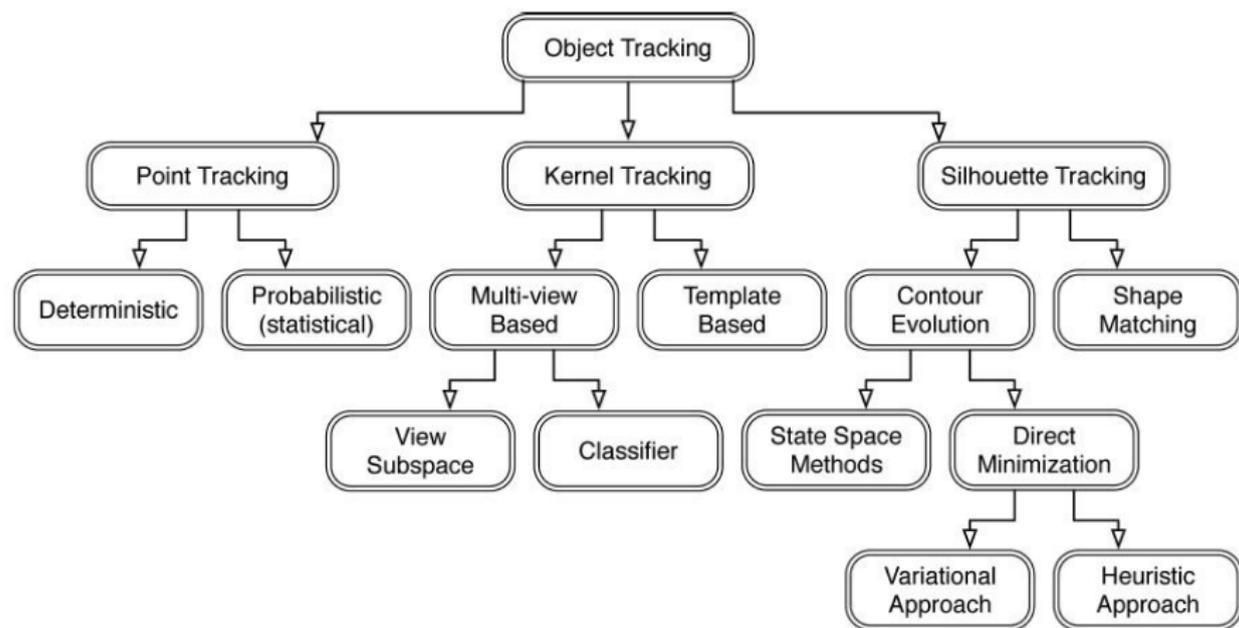
- 1 Suivi d'objet
 - Catégories des suivis d'objet
 - Suivi basé sur les points d'intérêt
 - Suivi basé sur les descripteurs locaux (topologie)
 - Suivi basé sur les noyaux (mean-shift)
 - Suivi basé sur les prédiction (Kalman)
 - Suivi d'objet basé sur les contours/silhouettes

Catégories de suivi d'objet

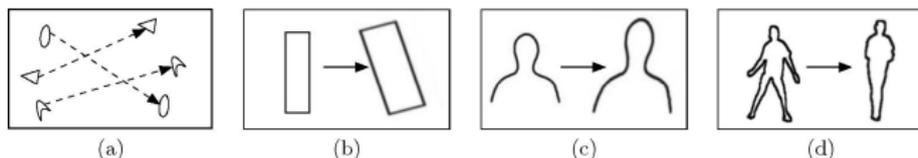
La tâche de détecter les objets et d'établir la correspondance entre les instances d'objet à chaque image peut être effectuée ensemble ou séparément.

- **Séparément** : Les régions possibles de présence d'objet à chaque image sont obtenues au moyen d'un algorithme de détection d'objet, puis un *tracker* fait correspondre les objets à travers les images.
- **Conjointement** : Les régions et correspondances d'objet sont estimés à chaque image en mettant à jour la position de l'objet ainsi que son descripteur de contenu.

Catégories de suivi d'objet

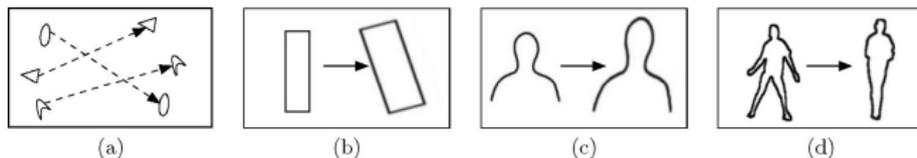


Catégories de suivi d'objet



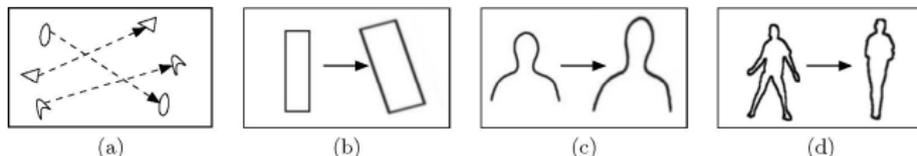
- **(a)** : Les objets détectés dans des images consécutives sont représentés par des points que l'on met en correspondance. Cette approche nécessite d'abord une détection de points et d'objets dans chaque image.

Catégories de suivi d'objet



- **(b)** : Kernel = Suivi du noyau, modèle de forme et d'apparence de l'objet. Par exemple, noyau = un modèle rectangulaire ou elliptique avec un histogramme associé. Les objets sont suivis par le calcul du mouvement (translation, rotation, affine) du noyau entre images consécutives.

Catégories de suivi d'objet



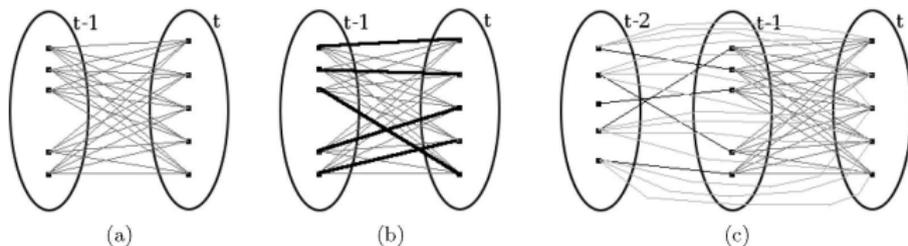
- **(c) (d)** : Suivi de silhouette. Ces méthodes utilisent les informations codées à l'intérieur de la zone de l'objet (apparence et forme). Compte tenu des modèles d'objet, les silhouettes sont suivies soit par appariement de formes (c) ou par évolution de contour (d). Ce dernier peut être apparenté aux segmentations d'objet dans le domaine temporel, en utilisant les probabilités *a priori* générées à partir des images précédentes.

Plan de la présentation

- 1 Suivi d'objet
 - Catégories des suivis d'objet
 - **Suivi basé sur les points d'intérêt**
 - Suivi basé sur les descripteurs locaux (topologie)
 - Suivi basé sur les noyaux (mean-shift)
 - Suivi basé sur les prédiction (Kalman)
 - Suivi d'objet basé sur les contours/silhouettes

Suivi de points

En général, après avoir modélisé nos objets avec des points d'intérêt (Harris, SIFT, etc), on les compare entre eux à l'aide d'une mesure de similarité de leur descripteur.



On teste la correspondance entre chaque point, on conserve la meilleure pour chaque point.

Suivi de points - RANSAC

Lorsqu'on a une série de point où la mise en correspondance a été effectuée, on peut utiliser un algorithme permettant d'extrapoler le déplacement de l'objet (RANSAC, etc) (Chapitre 2.3).

Un avantage de RANSAC est sa capacité à calculer de manière robuste les paramètres du modèle de transformation, même s'il y a beaucoup d'outliers.

Un inconvénient de RANSAC est qu'il n'y a pas de limite supérieure sur le temps de calcul de ces paramètres.

Suivi de points - Iterative closest point (ICP)

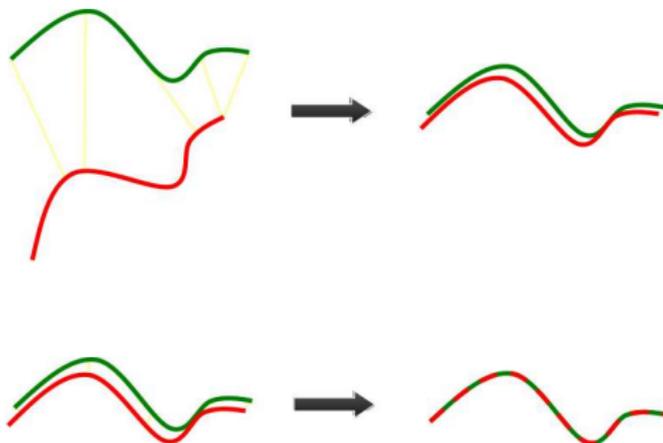
L'algorithme d'Iterative Closest Point est utilisé afin de faire correspondre deux nuages de points ensemble.

La méthode consiste à relier tous les points d'un modèle sur un nuage de point, en modifiant la position des point par une transformation globale afin de minimiser une mesure de distance basée sur tous les points. Les étapes principales de l'algorithme sont :

- 1 Association des points par les critères du plus proche voisin.
- 2 Estimation des paramètres de transformation utilisant une fonction de coût quadratique moyenne.
- 3 Transformer les points en utilisant les paramètres estimés.
- 4 Itération (ré-associer les points etc).

Suivi de points - Iterative closest point (ICP)

En assumant que **le point le plus près constitue un match**, on itère en transformant le nuage de point pour s'approcher progressivement d'une solution optimale.



Suivi de points - Iterative closest point (ICP)

L'algorithme va converger si :

- Le modèle et la cible sont déjà presque alignés (petits mouvements)
- Il n'y a pas trop d'outliers.

Pour optimiser l'algorithme, on peut travailler sur plusieurs points :

- 1 La sélection initiales des points (réduire les outliers) ;
- 2 La mesure de similarité des points ;
- 3 L'algorithme de minimisation de la mesure de similarité.

Suivi de points - Iterative closest point (ICP)

1 : La sélection initiales des points

On peut :

- contrôler la densité de points à utiliser afin de réduire le temps de calcul.
- contrôler la qualité des points obtenus (SIFT vs Harris)

Suivi de points - Iterative closest point (ICP)

2 : La mesure de similarité des points

On peut ajouter une contrainte de similarité sur les points (au lieu de prendre le plus près).

Ex. : Si le point le plus près est trop différent du point du modèle, alors on l'exclue du calcul.

Suivi de points - Iterative closest point (ICP)

3 : L'algorithme de minimisation de la mesure de similarité

La plus grande perte de temps de l'algorithme est dans la recherche du point le plus près.

On peut d'abord filtrer les points de la scène qui ne sont pas dans une zone de mouvement, puis aligner le centroïde du modèle avec celui de la scène, puis lancer notre algorithme de recherche de transformation.

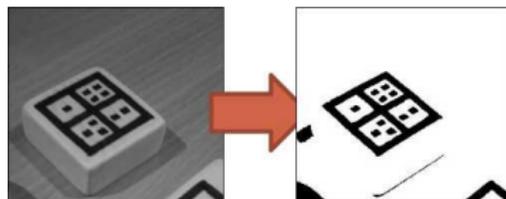
Pour trouver une transformation potentielle, on peut utiliser RANSAC.

Plan de la présentation

- 1 Suivi d'objet
 - Catégories des suivis d'objet
 - Suivi basé sur les points d'intérêt
 - **Suivi basé sur les descripteurs locaux (topologie)**
 - Suivi basé sur les noyaux (mean-shift)
 - Suivi basé sur les prédiction (Kalman)
 - Suivi d'objet basé sur les contours/silhouettes

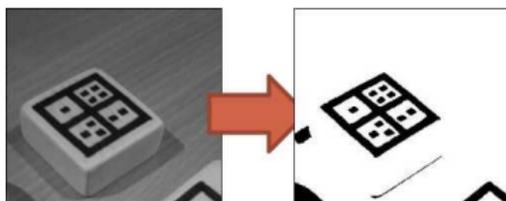
Recherche par topologie

Si on a accès à une segmentation binaire (ou à plusieurs classes) de la scène et de notre objet, on peut utiliser des méthodes à recherche par topologie (IMN638).



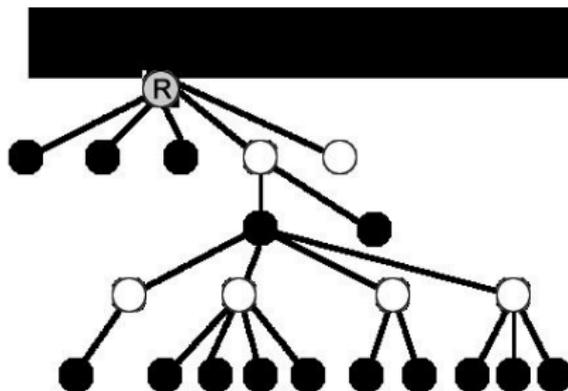
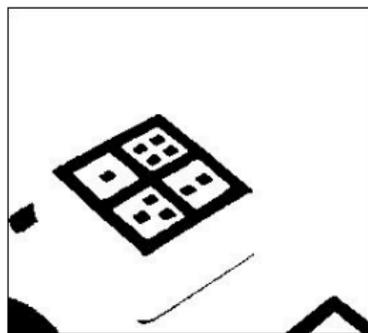
Recherche par topologie

La relation «contenant/contenu» permet de créer une hiérarchie qu'il est possible de représenter sous la forme d'un arbre. Cet arbre est appelé « arbre d'adjacence » car il permet de décrire l'adjacence (la relation contenant/contenu) entre les différentes régions du marqueur.



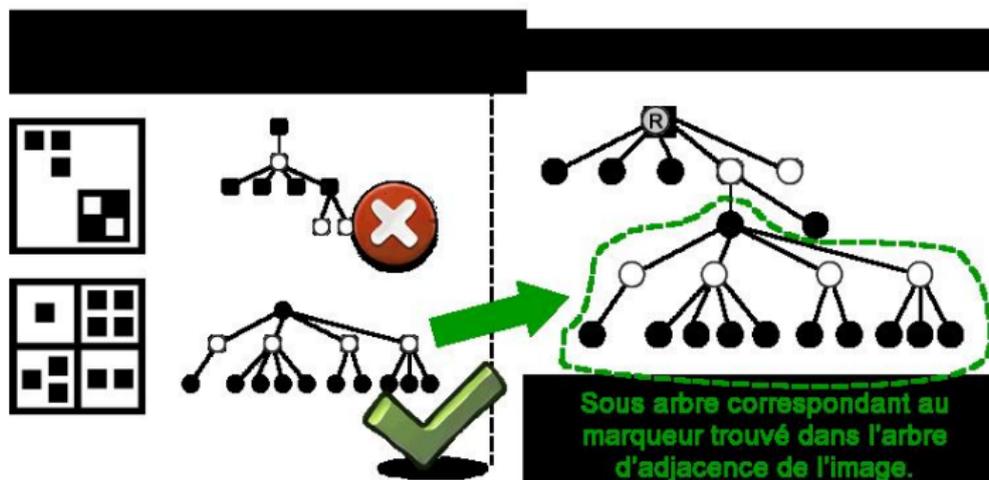
Recherche par topologie

Une fois la segmentation deux classe effectuée, l'étape suivante consiste à prendre notre image binaire puis générer un arbre d'adjacence de la totalité de l'image.



Recherche par topologie

Notre arbre d'adjacence généré pour la totalité de l'image, la recherche du marqueur topologique en soit peut commencer. Le tout s'effectue en recherchant un sous-arbre dans l'arbre généré à partir de l'image qui correspond à celui d'un des marqueurs dans notre librairie.



Recherche par topologie - translation

Pour trouver notre translation, on peut simplement utiliser le barycentre de la racine du marqueur, le centre de son aire englobant ou la moyenne de position des barycentres de ses régions pour déterminer sa position sur le plan image.

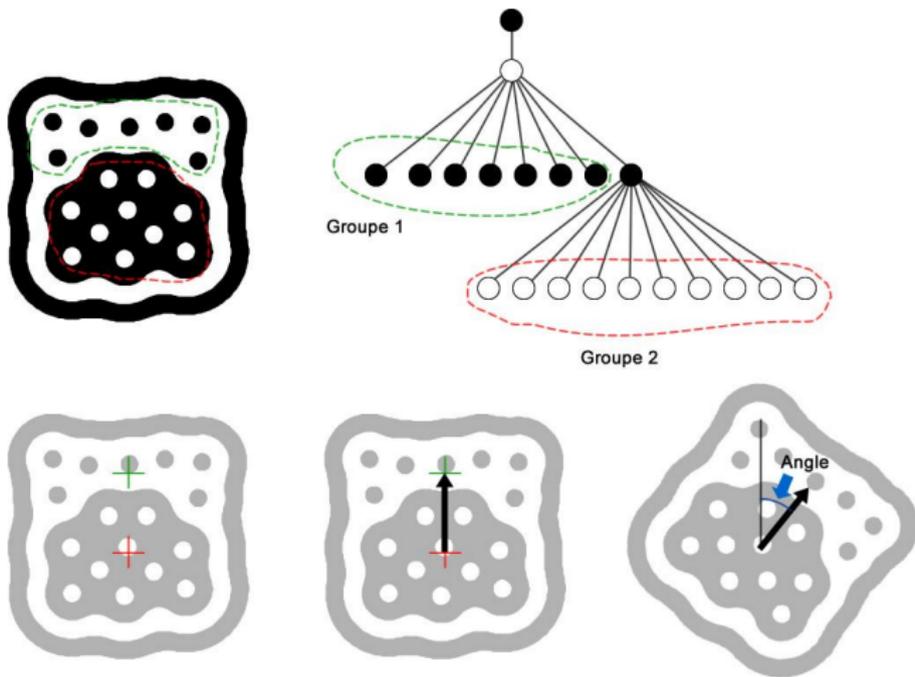
Par contre, nous sommes limités à une translation seulement.

Recherche par topologie - rotation ou autre

Rotation : L'arbre des marqueurs peut être divisé en deux groupes distincts. La position d'un des groupes par rapport à l'autre est calculée à l'aide des deux barycentres qui permettent de construire un vecteur. L'orientation de ce vecteur dans l'espace fenêtre donne la rotation (Le vecteur doit être perpendiculaire au vecteur de vision de la caméra pour que ceci fonctionne.)

Matrice de transformation : L'arbre des marqueurs peut être divisé en quatre sections différentes. Le centre de chaque section devient équivalent à un « coin » d'un marqueur planaire et peut être utilisé pour trouver la matrice de transformation avec les méthodes précédemment vu.

Recherche par topologie - rotation ou autre



Plan de la présentation

- 1 Suivi d'objet
 - Catégories des suivis d'objet
 - Suivi basé sur les points d'intérêt
 - Suivi basé sur les descripteurs locaux (topologie)
 - **Suivi basé sur les noyaux (mean-shift)**
 - Suivi basé sur les prédiction (Kalman)
 - Suivi d'objet basé sur les contours/silhouettes

Suivi de noyau

En général, la méthode “recherche brute” pour le suivi de noyau d'objet consiste à :

- 1 Définir une zone de recherche ;
- 2 Placer le modèle défini à partir de l'image précédente **à chaque position** de la zone de recherche et calculer une mesure de similarité entre le modèle et le modèle-cible (candidat) ;
- 3 Choisir le meilleur candidat maximisant une mesure de similarité.

Bien sûr, comme la recherche serait beaucoup trop longue pour toute l'image, on limite à l'entourage de l'objet.

Suivi de noyau

En général, on peut soit comparer directement les noyaux “modèle (m)” et “cible (c)”, soit comparer leur distribution. Pour ce faire, on peut définir des mesures :

- Mesures directes :

- $d_{L1}(m, c) = \sum_{i \in \text{noyau}} |m(i) - c(i)|$

- $d_{LSQ}(m, c) = \sum_{i \in \text{noyau}} (m(i) - c(i))^2$

- $d_{corr}(m, c) = \sum_{i \in \text{noyau}} m(i) \cdot c(i)$

Suivi de noyau

En général, on peut soit comparer directement les noyaux “modèle (m)” et “cible (c)”, soit comparer leur distribution. Pour ce faire, on peut définir des mesures :

- Mesures basées sur la distribution (histogramme) :
 - $d_{EMD}(m, c) = \text{EarthMoverDistance}(H_m, H_c)$
 - $d_{Inter}(m, c) = \text{IntersectionHisto}(H_m, H_c)$
 - $d_{Hellinger}(m, c) = \sqrt{1 - \text{Bhattacharyya}(m, c)}$

Suivi de noyau

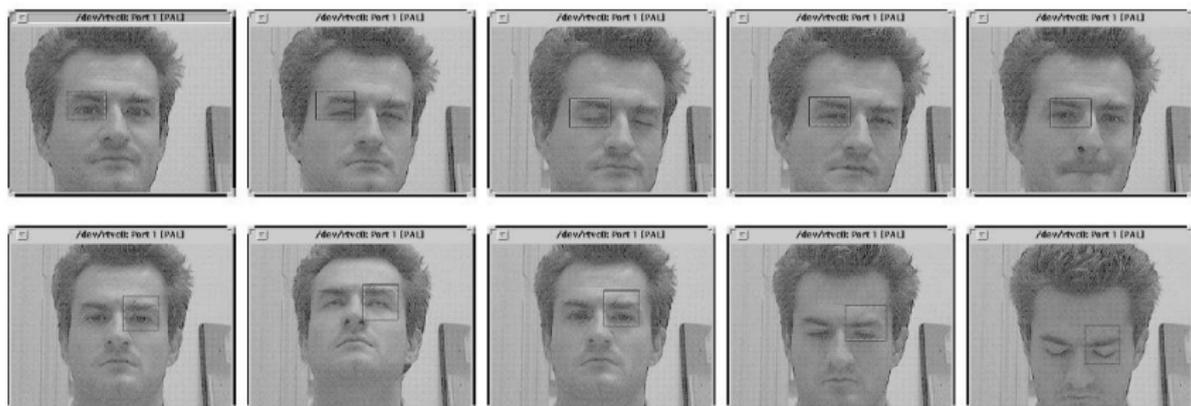
- $d_{\text{Hellinger}}(m, c) = \sqrt{1 - \text{Bhattacharyya}(m, c)}$

Le coefficient de Bhattacharyya permet d'évaluer l'information **communes** à deux distributions discrètes sous une forme d'intégrale rudimentaire. elle est définie comme :

$$\text{Bhattacharyya}(m, c) = \sum_{i \in \text{Histo}} \sqrt{H_m(i) \cdot H_c(i)}. \quad (1)$$

Suivi de noyau

Exemple de suivi de noyau par comparaison directe.



Suivi de noyau

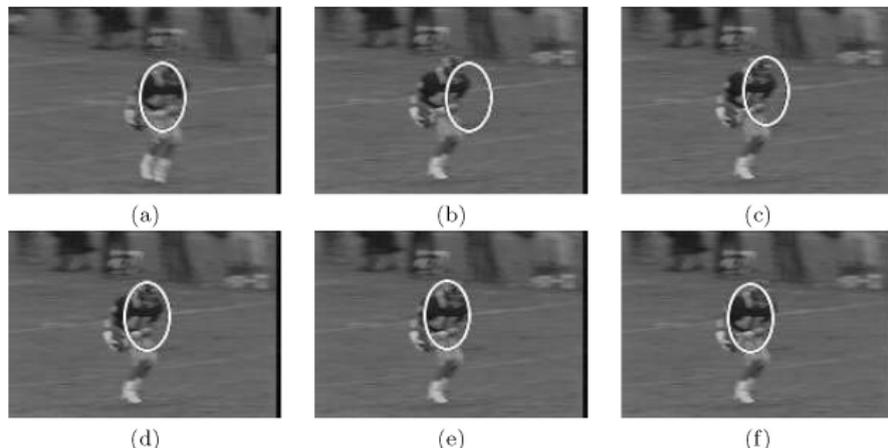
Exemple de suivi de noyau par distribution (couleur et gradient)



$$\mathbf{s}^* = \arg \max_{\mathbf{s}_i \in S} \{ \bar{\phi}_g(\mathbf{s}_i) + \bar{\phi}_c(\mathbf{s}_i) \}$$

Mean Shift : fonction de densité

L'algorithme de suivi de noyau le plus utilisé est le *mean shift*, qui utilise une fonction de densité des histogrammes de couleur pour apparié le modèle et la cible :

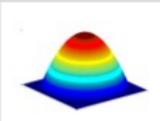
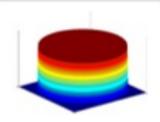
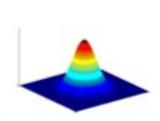


Mean Shift : fonction de densité

Fonction de densité appliquée à l'histogramme binarisé

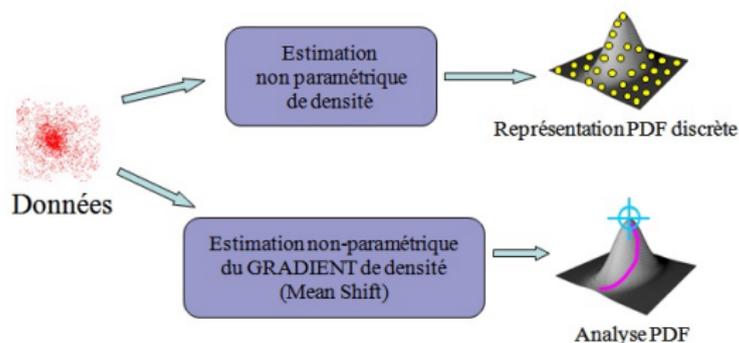
Plutôt que de prendre les données directement pour former l'histogramme de l'objet p et les cibles $p(y)$, on pondère à l'aide d'une fonction de densité.

Mean-shift utilise un noyau Epanechnikov (quadratique), car on utilisera sa forme dérivée (noyau uniforme).

• Noyau Epanechnikov	$K_E(\mathbf{x}) = \begin{cases} c(1 - \ \mathbf{x}\ ^2) & \ \mathbf{x}\ \leq 1 \\ 0 & \text{otherwise} \end{cases}$	
• Noyau Uniforme	$K_U(\mathbf{x}) = \begin{cases} c & \ \mathbf{x}\ \leq 1 \\ 0 & \text{otherwise} \end{cases}$	
• Noyau Normal	$K_N(\mathbf{x}) = c \cdot \exp\left(-\frac{1}{2}\ \mathbf{x}\ ^2\right)$	

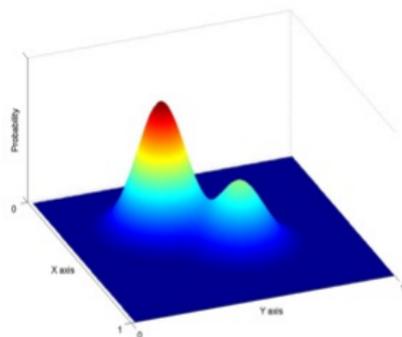
Mean Shift : fonction de densité

Cette représentation pondérée par la fonction de densité (noyau) permet d'avoir une distribution qui sera très **dense** (poids important) au centre de l'objet, et qui sera de moins en moins dense en s'éloignant (poids moins important). Ainsi, on s'assure de pouvoir sortir une direction de changement valide afin de maximiser la comparaison des histogrammes.

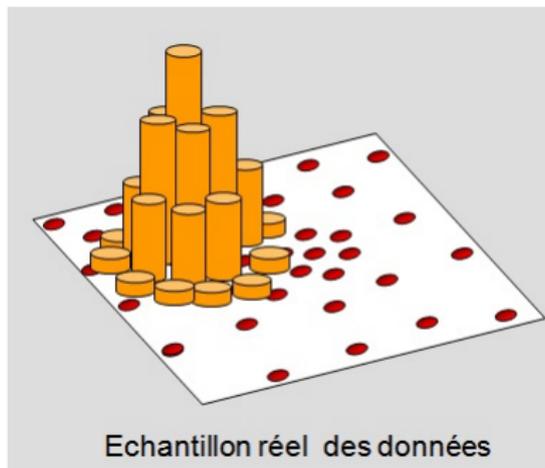


Mean Shift : fonction de densité

Le **mean shift** consiste à trouver, de façon itérative, le déplacement d'un objet en mettant en correspondance leurs histogrammes pondérés par une fonction de densité.

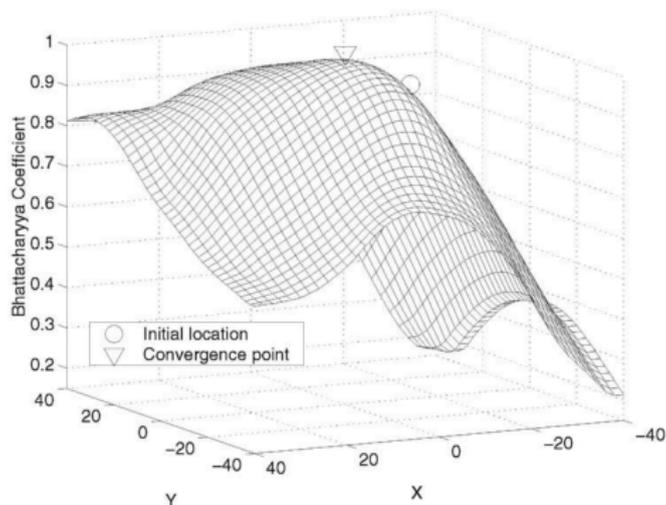


PDF Supposée



Echantillon réel des données

Mean Shift : fonction de densité



L'utilisation du noyau permet de “lisser” la fonction de similarité (le coefficient de bhattacharyya) et d'extrapoler une direction de changement.

Mean shift - Justification mathématique

Objet référence

Le mean-shift suit le même principe qu'une descente de gradient, mais en se basant sur l'estimation des fonctions de densité (non-paramétrique) :

- q l'objet à suivre (histogramme pondéré)
- $p(y_j)$ un objet candidat à la position y et à l'itération j (histogramme pondéré)
- $1..m$ est l'indice d'une bin d'histogramme ($q_u =$ valeur de l'histogramme de l'objet)
- x_i la position d'un pixel relatif au centre de la cible
- K_e représente le noyau epanechnikov
- K_n le noyau uniforme (la dérivée de K_e)
- w_i Une pondération du pixel x_i
- $b(x_j)$ Bin associée à la valeur du pixel x_j

Construction des histos pondérés :

$$\mathbf{q}_u = \sum_{i \in \text{cible}} K_e(\|\mathbf{y}_0 - x_i\|) \cdot \delta[b(x_i) - u] \quad (2)$$

$$\mathbf{p}_u(y_j) = \sum_{i \in \text{cible}} K_e(\|\mathbf{y}_j - x_i\|) \cdot \delta[b(x_i) - u] \quad (3)$$

Mean shift - Justification mathématique

Objet référence

Le mean-shift suit le même principe qu'une descente de gradient, mais en se basant sur l'estimation des fonctions de densité (non-paramétrique) :

q l'objet à suivre (histogramme pondéré)

$\rho(y_j)$ un objet candidat à la position y et à l'itération j (histogramme pondéré)

$u \in 1..m$ est l'indice d'un bin d'histogramme ($q_u =$ valeur de l'histogramme de l'objet)

x_j la position d'un pixel relatif au centre de la cible

K_e représente le noyau epanechnikov

K_n le noyau uniforme (la dérivée de K_e)

w_i Une pondération du pixel x_i

$b(x_i)$ Bin associée à la valeur du pixel x_i

Équation du mean-shift :

$$\mathbf{y}_1 = \frac{\sum_{i \in \text{cible}} x_i \cdot w_i \cdot K_n(\|\mathbf{y}_0 - x_i\|)}{\sum_{i \in \text{cible}} w_i \cdot K_n(\|\mathbf{y}_0 - x_i\|)} \quad (4)$$

Mean shift - Justification mathématique

Objet référence

Le mean-shift suit le même principe qu'une descente de gradient, mais en se basant sur l'estimation des fonctions de densité (non-paramétrique) :

q l'objet à suivre (histogramme pondéré)

$p(y_j)$ un objet candidat à la position y et à l'itération j (histogramme pondéré)

$u \in 1..m$ est l'indice d'un bin d'histogramme ($q_u =$ valeur de l'histogramme de l'objet)

x_i la position d'un pixel relatif au centre de la cible

K_e représente le noyau epanechnikov

K_n le noyau uniforme (la dérivée de K_e)

w_i Une pondération du pixel x_i

$b(x_i)$ Bin associée à la valeur du pixel x_i

Pondération w_i , à calculer pour chaque pixel :

$$w_i = \sum_{u \in 1..m} \sqrt{\frac{q_u}{p_u(y_0)}} \cdot \delta[b(x_i) - u] \quad (5)$$

où δ est la fonction dirac.

Mean shift - Suivi de l'objet

Algorithme de suivi par le mean shift

- 1 Calculer q et $\rho(y_0)$ selon Eq.(2) et Eq.(3)
- 2 Évaluer $\rho(y_0) = \sum_{u \in 1..m} \sqrt{q_u \cdot p_u(y_0)}$ (ρ = Coefficient de Bhattacharyya)
- 3 Évaluer les poids w_i pour chaque pixel selon Eq.(5)
- 4 Trouver le prochain déplacement (y_1) en utilisant Eq.(4).
- 5 Calculer $\rho(y_1)$ selon Eq.(3) et évaluer $\rho(y_1) = \sum_{u \in 1..m} \sqrt{q_u \cdot p_u(y_1)}$
- 6 **Tant que** $(\rho(y_1) < \rho(y_0))$ **faire** :
 - $y_1 \leftarrow \frac{(y_1) + (y_0)}{2}$
 - Re-évaluer $\rho(y_1)$.
- 7 **Si** $(\|(y_0) - (y_1)\| < \epsilon)$ **Fin** de l'algorithme, **sinon** $(y_0) \leftarrow (y_1)$ et aller à l'étape 3.

Mean shift - Amélioration

Amélioration : la mise à échelle (Camshift)

- On ajoute un paramètre h qui permet d'ajuster la taille de l'objet cible afin de mieux représenter l'objet en mouvement.
- La mesure de similarité (coefficient de Bhattachariia) est **invariante au changement d'échelle.**

Équation du cam-shift :

$$\mathbf{y}_1 = \frac{\sum_{i \in \text{cible}} x_i \cdot w_i \cdot K_n\left(\left\|\frac{\mathbf{y}_0 - x_i}{h}\right\|\right)}{\sum_{i \in \text{cible}} w_i \cdot K_n\left(\left\|\frac{\mathbf{y}_0 - x_i}{h}\right\|\right)} \quad (6)$$

Mean shift - cam-shift

Après convergence du mean-shift, on effectue la comparaison **3 fois**, avec comme valeur de h , où h_{prev} définit la valeur de h retenue à l'itération précédente :

① $h = h_{prev}$

② $h = h_{prev} + \Delta h$

③ $h = h_{prev} - \Delta h$

$$\Delta = 0.1 h_{prev}$$

On prend le meilleur résultat h_{opt} . Pour être moins sensible au changement d'échelle, on se définit la fonction suivante :

$$h = \alpha h_{opt} + (1 - \alpha) h_{prev}$$

où $\alpha = 0.1$

Mean shift - Améliorations

COMANICIU ET AL.: KERNEL-BASED OBJECT TRACKING



Fig. 1. Football sequence, tracking player number 75. The frames 30, 75, 105, 140, and 150 are shown.

Mean shift - Améliorations

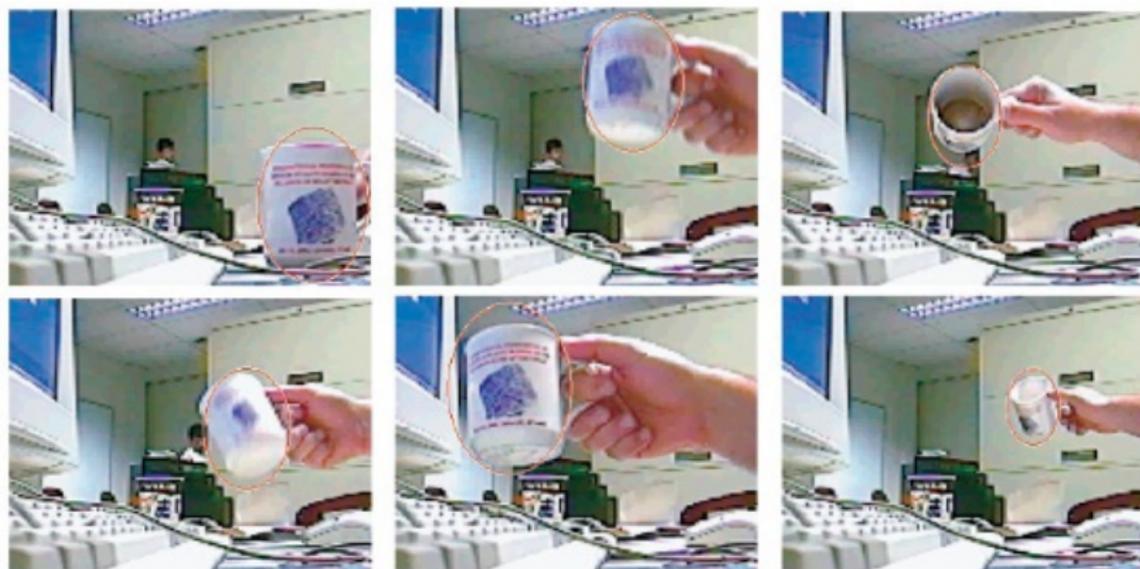


Fig. 7. *Mug* sequence. The frames 60, 150, 240, 270, 360, and 960 are shown.

Mean shift - Amélioration

Apprentissage de l'arrière-plan

En apprenant l'arrière-plan, on peut masquer les valeurs d'intensité reliées à l'arrière-plan.

De cette façon, le suivi ne pourra être confondu avec des objets stables de l'arrière-plan.

Mean shift - Améliorations

Filtre de Kalman

En intégrant les **filtres de Kalman**, on augmente la robustesse de l'estimation du déplacement en deux étapes :

- Prédiction de l'état suivant ;
- Mise-à-jour de l'état

NOTE : Nous aborderons ce thème au prochain sous-chapitre...

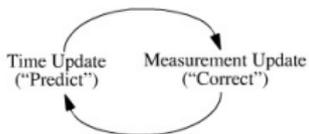
Plan de la présentation

- 1 Suivi d'objet
 - Catégories des suivis d'objet
 - Suivi basé sur les points d'intérêt
 - Suivi basé sur les descripteurs locaux (topologie)
 - Suivi basé sur les noyaux (mean-shift)
 - **Suivi basé sur les prédiction (Kalman)**
 - Suivi d'objet basé sur les contours/silhouettes

Introduction sur les filtres de Kalman

Le filtre de Kalman : Ceci permet d'avoir un suivi facilité par le fait que l'on **prend en compte les précédents déplacements pour essayer de deviner où se trouve l'objet à l'instant t** .

Il faut initier le filtre, puis, en corrigeant à chaque étape sa prédiction avec la position réelle, l'estimation du déplacement **est de plus en plus précise**.



Introduction sur les filtres de Kalman

- **Estimation *a posteriori*** (\hat{x}_t^+) : Estimation du déplacement à l'instant t basée sur l'estimation *a priori* du déplacement à l'instant t .
- **Estimation *a priori*** (\hat{x}_t^-) : Estimation du déplacement calculée à partir des estimations *a posteriori* aux instants précédents ($t-1$, $t-2$, ...).

Équation du système

L'équation régissant un lien linéaire stochastique entre **l'état de l'objet** et la **mesure de ce déplacement** est :

$$\begin{aligned} x_t &= A \cdot x_{t-1} + B \cdot u_t + w_{t-1} && \rightarrow \text{État} \\ z_t &= H \cdot x_t + v_k && \rightarrow \text{Mesure} \end{aligned} \quad (7)$$

Où w_k et v_k représentent le **bruit (incertitude) d'état** et le **bruit de la mesure**. Ils sont indépendants et régis selon une loi de probabilité *normale* :

$$p(w) \sim N(0, Q) \quad (8)$$

$$p(v) \sim N(0, R) \quad (9)$$

La **covariance du bruit de déplacement Q** et la **covariance du bruit de mesure R** sont assumées constantes même s'ils peuvent changer en pratique. Ils sont estimés dans la phase d'apprentissage.

Équation du système

$$\begin{array}{lcl} x_t = & A \cdot x_{t-1} + B \cdot u_k + w_{t-1} & \rightarrow \text{Déplacement} \\ z_t = & H \cdot x_t + v_t & \rightarrow \text{Mesure} \end{array}$$

- $A_{n \times n}$ fait le lien entre l'état du déplacement du temps $t-1$ avec le temps t .
- $B_{n \times l}$ fait le lien entre l'option de contrôle optionnel $u_k \in \mathbb{R}^l$ avec l'état x_k .
- $H_{m \times n}$ fait le lien l'état x_k et la mesure $z_k \in \mathbb{R}^m$.

En pratique, A , B , H varient en fonction du temps, mais elles sont déterminées en phase d'apprentissage et supposées constantes.

Équation de l'estimation

On a définie l'estimation *a posteriori* \hat{x}_t^+ comme étant une **combinaison linéaire** entre l'estimation *a priori* \hat{x}_t^- et une **différence pondérée** entre une mesure z_k et l'estimation *a priori* de cette mesure $H \cdot \hat{x}_t^-$ (appelée résidu :

$$\hat{x}_t^+ = \hat{x}_t^- + K \cdot (z_k - H \cdot \hat{x}_t^-) \quad (10)$$

Un résidu de 0 signifie une prédiction exacte de la mesure.

Équation de l'erreur de prédiction

On peut exprimer l'erreur d'état *a priori* de l'étape t comme étant la **différence entre l'état et l'estimation *a priori* de l'état** ($\hat{x}_t^- \in \mathbb{R}^n$).

On peut exprimer l'erreur d'état *a posteriori* de l'étape t comme étant la **différence entre l'état et l'estimation *a posteriori* de l'état** ($\hat{x}_t^+ \in \mathbb{R}^n$).

$$\begin{aligned}e_t^- &= x_t - \hat{x}_t^- \\e_t^+ &= x_t - \hat{x}_t^+\end{aligned}$$

Équation de l'erreur de prédiction

D'un point de vue statistique, on peut exprimer la **covariance de l'erreur d'estimation *a priori*** et la **covariance de l'erreur d'estimation *a posteriori***, qui permettent d'évaluer la dépendance entre l'erreur de prédiction \hat{x} avec l'état x , comme étant :

$$\begin{aligned} P_t^- &= E[e_t^- \cdot e_t^{-T}] = E[(x_t - \hat{x}_t^-)(x_t - \hat{x}_t^-)^T] \\ P_t^+ &= E[e_t^+ \cdot e_t^{+T}] = E[(x_t - \hat{x}_t^+)(x_t - \hat{x}_t^+)^T] \end{aligned} \quad (11)$$

Équation du filtre de Kalman

$$\begin{aligned}
 P_t^- &= E[e_t^- \cdot e_t^{-T}] && \rightarrow \text{Eq.11-} \\
 P_t^+ &= E[e_t^+ \cdot e_t^{+T}] && \rightarrow \text{Eq.11+} \\
 \hat{x}_t^+ &= \hat{x}_t^- + K \cdot (z_k - H \cdot \hat{x}_t^-) && \rightarrow \text{Eq.10}
 \end{aligned}$$

$K_{n \times m}$ de l'Eq.10 représente le **facteur de gain (ou de mélange)** minimisant l'Eq.11+.

On obtient sa valeur en :

- 1 Substituant **Eq.10** dans l'équation de l'erreur e_t^+ ;
- 2 Substituant le résultat dans l'équation **Eq.11+** ;
- 3 Calculant l'esperance $E[e_t^+ \cdot e_t^{+T}]$;
- 4 Extrayant la dérivée de la trace du résultat, en égalant à 0 et en resolvant l'équation pour K .

Équation du filtre de Kalman

On obtient alors que le gain K peut être déterminé par l'équation suivante :

$$\begin{aligned} K_t &= P_t^- H^T \cdot (HP_t^- H^T + R)^{-1} \\ &= \frac{P_t^- H^T}{HP_t^- H^T + R} \end{aligned} \tag{12}$$

Équation du filtre de Kalman

En regardant l'**Eq.12**, on peut voir que **plus la covariance de l'erreur de mesure (R) approche 0, plus le gain K donne une plus grande pondération au résidu** :

$$\lim_{R \rightarrow 0} K_t = \frac{1}{H}$$

On constate aussi que **plus l'erreur d'estimation d'état *a priori* (P_t^-) approche 0, plus le gain K donne une plus petite pondération au résidu** :

$$\lim_{P_t^- \rightarrow 0} K_t = 0$$

Équation du filtre de Kalman

$$\begin{aligned}\lim_{R \rightarrow 0} K_t &= \frac{1}{H} \\ \lim_{P_t^- \rightarrow 0} K_t &= 0\end{aligned}$$

De ces équations, on peut tirer les évidences suivantes :

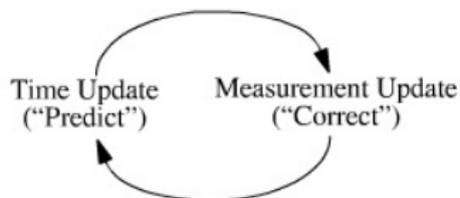
- Plus la covariance de l'erreur de mesure R approche de 0, plus on fait confiance à la mesure actuelle (z_k) et moins on fait confiance à la prédiction de mesure (l'*a priori* $H\hat{x}_t^-$).
- Plus la covariance d'erreur de prédiction P_t^- approche 0, moins on fait confiance à la mesure actuelle (z_k) et plus on fait confiance à la prédiction de mesure (l'*a priori* $H\hat{x}_t^-$).

Kalman : Cas discret

Forme générale

Le filtre de Kalman estime un déplacement en utilisant un contrôle issu du retour de résultat : Le filtre **estime l'état du système** à une moment et puis obtient un **résultat de mesure bruité** .

On sépare donc les équations du filtre en deux différentes catégories : ***Mise à jour du temps (MaJT)*** et ***Mise à jour de la mesure (MaJM)***.



Kalman : Cas discret

Forme générale

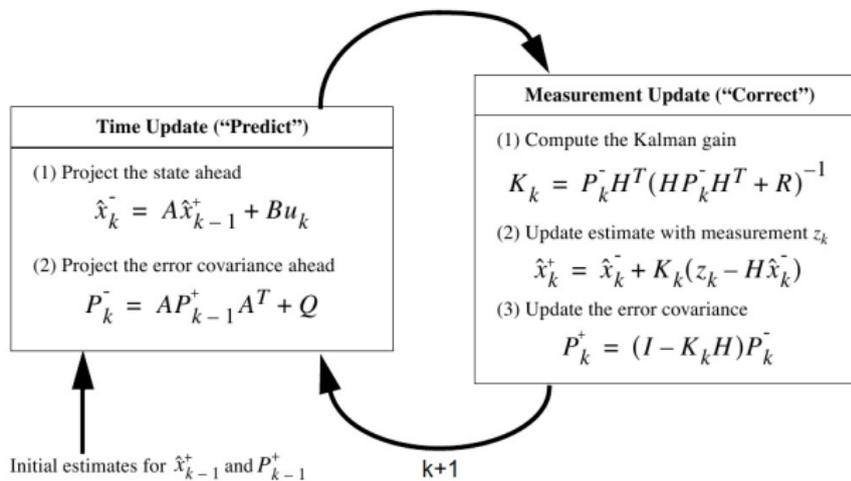
La **MaJT** s'occupe de la **prédiction** en calculant l'estimation de la mesure et la covariance de son erreur (\hat{x}_t^- et P_t^-).

La **MaJM** s'occupe quant à elle d'ajouter une mesure z et de corriger l'estimation *a priori* (calcul de l'estimation a posteriori \hat{x}_t^+ ainsi que la covariance de son erreur P_t^+).

Kalman : Cas discret

Forme générale

Au lieu de garder toutes les informations des anciens déplacements (mesures), le filtre de Karman conditionne sa prédiction sur mesures et prédictions précédentes. Le cycle complet de l'algorithme :



Kalman : Cas discret

Condition optimale

Le filtre de Kalman offrira des résultats optimaux dans le cas où :

- Le bruit d'état \mathbf{Q} et de mesure \mathbf{R} sont indépendants ;
- Le bruit d'état \mathbf{Q} et de mesure \mathbf{R} sont indépendants sont parfaitement représentés par un bruit de distribution normale.
- Le système est connu ;
- Le système est linéaire.

Kalman : Cas discret

Application au suivi d'image

Il faut comprendre que pour avoir une bonne variable d'état, il faut **bien modéliser notre système**.

Prenons l'exemple d'un cas de la robotique. Supposons $([x], [y])$ les coordonnées du centre de l'objet (le robot), $([vx], [vy])$ les vitesses de l'objet (le robot) et $([ax], [ay])$ l'accélération des objets. L'état x_t au temps t est donné par :

$$x_t = \begin{bmatrix} X_{[x],t} \\ X_{[y],t} \\ X_{[vx],t} \\ X_{[vy],t} \\ X_{[ax],t} \\ X_{[ay],t} \end{bmatrix} \quad (13)$$

Kalman : Cas discret

Application au suivi d'image

Les lois de la physique suivent le modèle de mouvement suivant :

$$\begin{array}{rclclcl}
 x_{[x],t+1} & = & x_{[x],t} & + & x_{[vx],t} \cdot \Delta t & + & \frac{x_{[ax],t} \cdot \Delta t^2}{2} \\
 x_{[y],t+1} & = & x_{[y],t} & + & x_{[vy],t} \cdot \Delta t & + & \frac{x_{[ay],t} \cdot \Delta t^2}{2} \\
 x_{[vx],t+1} & = & 0 & + & x_{[vx],t} & + & x_{[ax],t} \cdot \Delta t \\
 x_{[vy],t+1} & = & 0 & + & x_{[vy],t} & + & x_{[ay],t} \cdot \Delta t \\
 x_{[ax],t+1} & = & 0 & + & 0 & + & x_{[ax],t} \\
 x_{[ay],t+1} & = & 0 & + & 0 & + & x_{[ay],t}
 \end{array}$$

Donc, en supposant $\Delta t = 1$, la matrice A , constante, permettant de passer de l'état x_t à l'état x_{t+1} est donnée par :

$$A_{n \times n} = \begin{bmatrix} 1 & 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 1 & 0 & 1 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

Kalman : Cas discret

Application au suivi d'image

$$\begin{aligned} x_t &= A \cdot x_{t-1} + B \cdot u_k + w_{t-1} && \rightarrow \text{Déplacement} \\ z_t &= H \cdot x_t + v_t && \rightarrow \text{Mesure} \end{aligned}$$

$$\begin{aligned} p(w) &\sim N(0, Q) \\ p(v) &\sim N(0, R) \end{aligned}$$

Pour faire le lien avec l'état du système et la mesure, on a besoin de la matrice H . Comme nous ne voulons que le positionnement, suivant la même logique que pour la matrice A , la matrice H sera :

$$H_{n \times n} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (15)$$

Kalman : Cas discret

Application au suivi d'image

$$\begin{aligned} x_t &= A \cdot x_{t-1} + B \cdot u_k + w_{t-1} && \rightarrow \text{Déplacement} \\ z_t &= H \cdot x_t + v_t && \rightarrow \text{Mesure} \end{aligned}$$

$$\begin{aligned} p(w) &\sim N(0, Q) \\ p(v) &\sim N(0, R) \end{aligned}$$

Comment estime-t-on les covariances de bruit d'état $Q_{6 \times 6}$ et de bruit de mesure $R_{2 \times 2}$?

- Hypothèse 1 : Aucun bruit d'état ou de mesure $\rightarrow R = 0, Q = 0$
- Hypothèse 2 : Bruit de positionnement seulement $\rightarrow R = 0, Q = 0$ sauf pour Q_{11} et Q_{22}
- **Hypothèse 3 : Bruit de positionnement, de vitesse, d'accélération, de mesure.**

Kalman : Cas discret

Application au suivi d'image

Hypothèse 3 : Bruit d'état et de mesure.

Le bruit de la mesure R n'est pas disponible si on n'a accès qu'aux images de la vidéo. Cependant, si on connaît les imprécisions du système d'acquisition, celles-ci seraient intégrées à la matrice R . Les deux possibilités d'estimation de la matrice R sont :

- Mettre R à 0 ; → Donne de moins bons résultats (aucune imprécision).
- Mettre R à identité. → Permet d'estimer l'imprécision à une loi normale (gaussienne).

Kalman : Cas discret

Application au suivi d'image

Hypothèse 3 : Bruit d'état et de mesure.

Le bruit de positionnement, vitesse et accélération doit être déterminé expérimentalement. Il est régi par le comportement de l'objet : **Plus l'objet a un mouvement chaotique, moins l'hypothèse de constance de la matrice Q est vérifiée.** Pour estimer la matrice Q :

- 1 Faire le tracking et enregistrer l'estimation *a posteriori* d'états estimés \hat{x}_t^+
- 2 Au temps $t+1$, déterminer l'erreur d'état en suivant l'équation : *a priori* $w_t = \hat{x}_t^+ - A \cdot \hat{x}_{t-1}^+$.
- 3 Calculer $Q = cov(w_t)$ et itérer jusqu'à convergence de Q

Kalman : Cas discret

Application au suivi d'image

Plan de la présentation

- 1 Suivi d'objet
 - Catégories des suivis d'objet
 - Suivi basé sur les points d'intérêt
 - Suivi basé sur les descripteurs locaux (topologie)
 - Suivi basé sur les noyaux (mean-shift)
 - Suivi basé sur les prédiction (Kalman)
 - **Suivi d'objet basé sur les contours/silhouettes**

Suivi d'objet basé sur les contours/silhouettes

La définition des objets qui ont une forme complexe (humain, main, bras, etc) en utilisant des volumes englobant n'est parfois pas assez précises pour l'application.

Une représentation par contour ou silhouette permet d'avoir des application plus poussées qu'un simple suivi d'objet.

Suivi d'objet basé sur les contours/silhouettes

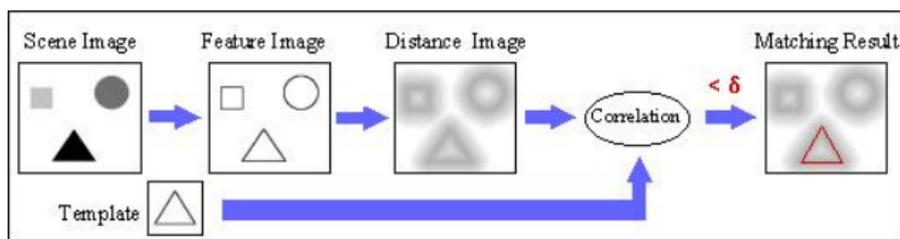
- **Détection d'action** : On recherche une silhouette dans le frame courant et on la compare avec une banque de silhouette (actions)
- **Évolution de contour** : On initialise un contour à la forme et position du précédent et minimise une fonction d'énergie pour la faire évoluer.

Suivi d'objet basé sur les contours/silhouettes

Reconnaissance par transformation de distance

On veut transformer nos images modèles et notre scène dans un espace où des primitives sont facilement extraites et associables.

On utilise un détecteur de contour (Canny), suivi d'une transformation de distance, où on calcule pour chaque point la distance vers le contour le plus près.

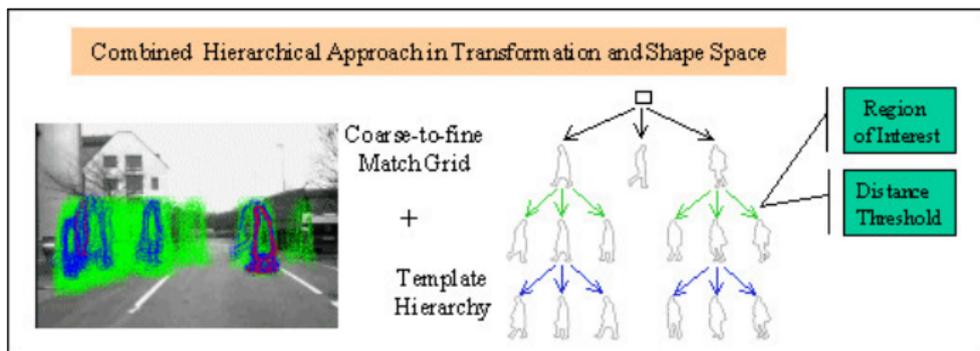


Suivi d'objet basé sur les contours/silhouettes

Reconnaissance par transformation de distance Par corrélation, on

trouve la translation en trouvant le modèle qui est le plus près possible de notre objet.

On doit donc avoir plusieurs poses et plusieurs résolution pour un objet. On peut organiser nos images de façon hiérarchique afin de faciliter la recherche : par taille, par pose, par type d'objet, etc.



Suivi d'objet basé sur les contours/silhouettes

Évolution de contour actif

On veut itérativement faire évoluer un contour initialisé à partir des frames précédents.

Cela implique donc que le contour actuel doit partiellement toucher au contour précédent afin de ne pas perdre l'objet.

Suivi d'objet basé sur les contours/silhouettes

Évolution de contour actif : Modèle général

Un **contour actif** (v) est représenté par une courbe v , fermée ou non, évoluant au cours du temps. Cette évolution s'effectue selon la minimisation d'une fonction d'énergie $E(v)$:

$$E_v = \int_0^1 \left[\alpha_{int} E_{int}(v(s)) + \alpha_{ext} E_{ext}(v(s)) \right] ds \quad (16)$$

où E_{int} et E_{ext} représentent les énergies internes et externes.

E_{int} Propriétés physiques intrinsèques du contour (contour globalement homogène et de forme simple) ;

E_{ext} Terme d'attache aux données qui permettra au contour actif de se fixer sur les limites des objets suivis.

Suivi d'objet basé sur les contours/silhouettes

Évolution de contour actif : Modèle discret

Une implémentation pratique (Modèle discret) consiste à discrétiser le contour. :

$$E_V = \alpha_{int} E_{int}(V) + \alpha_{ext} E_{ext}(V) \quad (17)$$

où V représente le contour actif discret formé par m points et $V(i)$ son i ème point. La déformation est effectuée itérativement et localement en chaque point du contour.

Suivi d'objet basé sur les contours/silhouettes

Évolution de contour actif : Modèle discret

On définit E_{int} comme :

$$E_{int} = \alpha_{con} E_{con}(V) + \alpha_{cou} E_{cou}(V) + \alpha_{bal} E_{bal}(V) \quad (18)$$

E_{con} E_{cou} E_{bal} sont les énergies associés aux forces de continuité, courbure et ballonnement.

E_{con} Détermine la continuité de la courbe en vérifiant que les points sont le plus équidistants possible. $E_{con} V^t(i)$ est la différence entre la distance du point $V^t(i)$ puis suivant $V^t(i+1)$ et $V^t(i)$ puis le même de l'itération précédente $V^{t-1}(i)$.

Suivi d'objet basé sur les contours/silhouettes

Évolution de contour actif : Modèle discret

On définit E_{int} comme :

$$E_{int} = \alpha_{con} E_{con}(V) + \alpha_{cou} E_{cou}(V) + \alpha_{bal} E_{bal}(V) \quad (19)$$

E_{con} E_{cou} E_{bal} sont les énergies associés aux forces de continuité, courbure et ballonnement.

E_{cou} Dérivée discrète d'ordre deux, s'assure qu'il n'y ait pas de points isolés.

Suivi d'objet basé sur les contours/silhouettes

Évolution de contour actif : Modèle discret

On définit E_{int} comme :

$$E_{int} = \alpha_{con} E_{con}(V) + \alpha_{cou} E_{cou}(V) + \alpha_{bal} E_{bal}(V) \quad (20)$$

E_{con} E_{cou} E_{bal} sont les énergies associés aux forces de continuité, courbure et ballonnement.

E_{bal} Permet au contour de se réduire ou grossir, dépendamment du signe du coefficient. Si on utilise un volume englobant, le coefficient sera négatif. C'est le produit scalaire entre le vecteur normal à $V(i)$ et le vecteur formé par $V^t(i+1)$ et $V^t(i)$

Suivi d'objet basé sur les contours/silhouettes

Évolution de contour actif : Modèle discret

On définit E_{int} comme :

$$E_{ext} = \alpha_{grad} E_{grad}(V) + \alpha_{int} E_{int}(V) \quad (21)$$

E_{grad} E_{int} sont les énergies associés au gradient et à l'intensité de l'image.

E_{grad} On estime le gradient à l'intérieur du volume englobant (Sobel, Canny). Un point non situé sur un contour sera alors pénalisé.

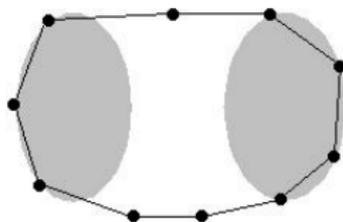
E_{int} Permet d'insérer un *a priori* sur la couleur de l'objet à segmenter.

Suivi d'objet basé sur les contours/silhouettes

Évolution de contour actif

On utilise un algorithme itératif afin de faire varier les points avec un nombre d'étape fixe (50).

Les algorithmes peuvent comporter des étapes supplémentaires pour gérer les occlusions ou les croisements, tel une opération de *fusion* et *scision*



Suivi d'objet basé sur les contours/silhouettes

