

MASTER  
INFORMATIQUE



Université  
Paris Cité

Nicolas Loménie

# Vision par Ordinateur 3D en Mouvement

MASTER  
INFORMATIQUE



Université  
Paris Cité

Nicolas Loménie

Analyse de Séquences Vidéo

TRACKING  
TRACKING

# Techniques de Matching

Il s'agit d'estimer le champ de déplacement en certains points correspondant à des primitives seulement.

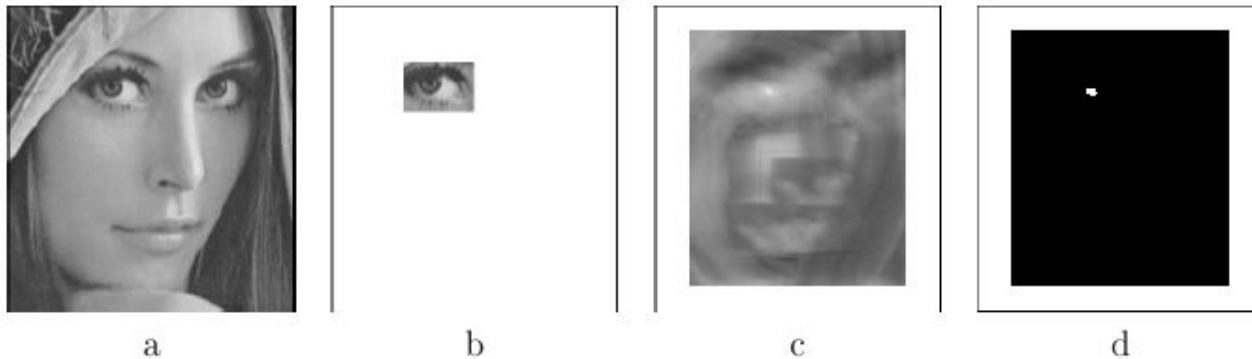
Le résultat est un champ de déplacement épars.

On illustrera deux cas :

1. L'analyse frame-à-frame dans le cas de 2 frames seulement, ce qui revient à trouver les disparités entre deux frames consécutives.
2. Le tracking du déplacement d'une primitive à travers une séquence d'images plus longue, et qui peut améliorer la robustesse du matching frame-à-frame.

# Techniques de Matching par autocorrélation

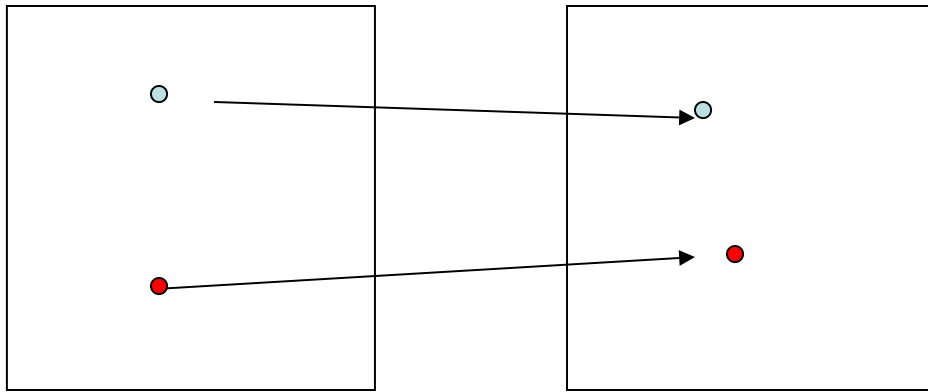
$$N(i, j) = \frac{\sum_{x,y \in \mathcal{P}} T(x, y) \cdot I(x - i, y - j)}{\sqrt{\sum_{x,y \in \mathcal{P}} T(x, y)^2 \cdot \sum_{x,y \in \mathcal{P}} I(x - i, y - j)^2}} \quad (2.3)$$



**FIG. 2.1:** a) Image originale b) noyau choisi c) autocorrélation d) autocorrélation seuillée. On retrouve bien l'emplacement du noyau recherché

## Matching de 2 frames : Tracking avec jetons

Etant donné deux frames capturées à différents instants et  $m$  points dans chaque image, le problème de mec revient à mapper un point d'une frame à un point de l'autre frame de sorte que deux points ne se mappent pas en un même point.



Le problème en stéréo est évidemment simplifié puisque les correspondants possibles se trouvent le long de la droite épipolaire.

Dans le cas d'objets se déplaçant et de frames prises à intervalle quelconque, les matches possibles peuvent se trouver quasiment n'importe où dans la frame suivante, contrairement à la stéréo où les frames sont séparés dans l'espace et non dans le temps.

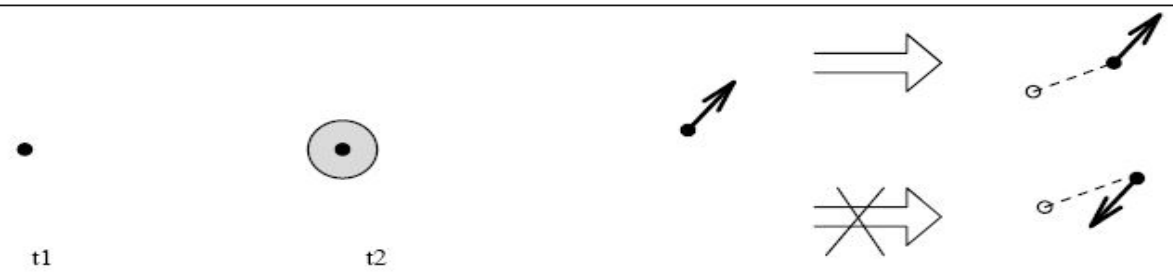
Le problème de mec est ainsi extrêmement combinatoire et lourd à traiter de façon brutale. Pour 2 frames et 5 points dans chaque frame, le nombre de mappings possible est  $5!=120$ . Par ailleurs, même si on peut éventuellement construire ces 120 mappings, comment choisir le meilleur ?

Pour réduire la complexité du problème, on utilise des contraintes physiques ou heuristiques.

Smoothness



Rigidity



(a)

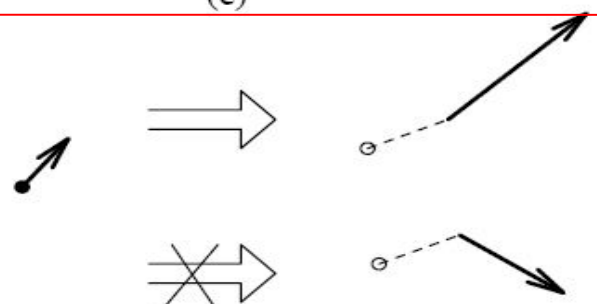
(b)



(c)



(d)



(f)

(e)

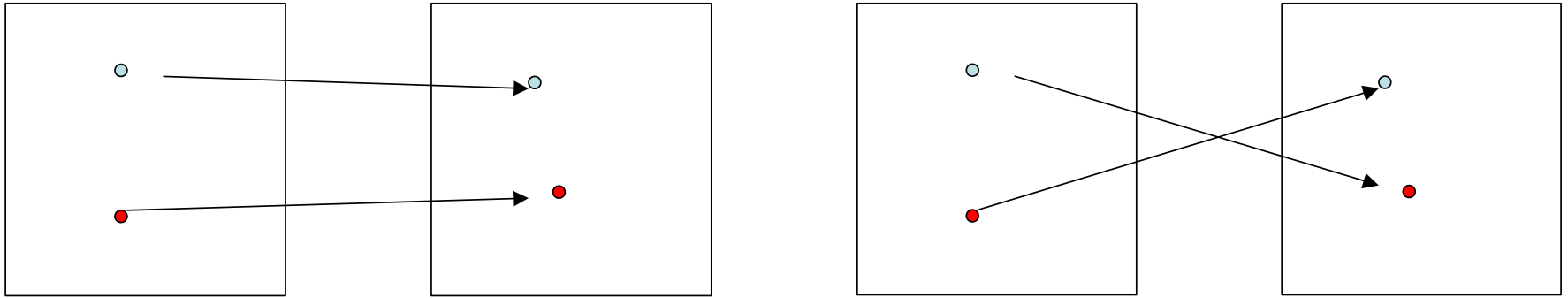
Techniques différentielles



(See M. Shah book)

Constraints used in motion correspondence. (a) Maximum velocity, (b) Small velocity changes, (c) common motion, (d) consistent match, (e) model, (f) proximal uniformity constraint.

La difficulté est de traduire ces heuristiques qualitatives en expressions quantitatives afin de définir des fonctions de coûts. (Techniques d'optimisation de type Programmation Dynamique utilisées en Reconnaissance de Formes structurelles).



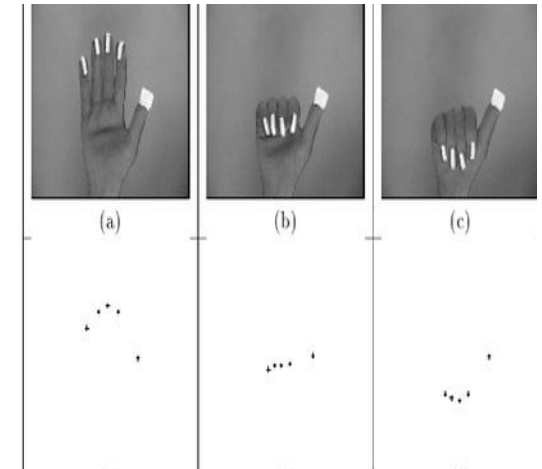
Le problème revient alors à déterminer le mapping optimal en fonction de ces fonctions de coût.

L'énumération de toutes les configurations possibles n'est en général pas possible. Il s'agit donc de trouver un bon algorithme d'approximation pour obtenir une solution sub-optimal très proche de la solution optimale.



# Tracking avec jetons : Algorithme itératif

S. T. Barnard and W. B. Thompson, "Disparity analysis in images," IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-2, pp. 333- 340, 1980.



Il s'agit d'un algorithme itératif pour calculer le flot optique pour  $m$  jetons.

Ils utilisent une mesure de confiance  $P_{ij}$  qui code la probabilité d'un jeton  $i$  du frame  $f^1$  à être associée au jeton  $j$  du frame  $f^2$  :

$$P_{ij} = P( (i,1) \leftrightarrow (j,2) )$$

La probabilité initiale  $P_{ij}^0$  sont calculées en utilisant des différences de niveaux de gris dans des petites fenêtres autour des jetons se correspondants :

$$P_{ij}^0 = \frac{1}{c + w_{ij}}$$

avec

$$w_{ij} = \sum_{dy=-w}^{dy=w} \sum_{dx=-w}^{dx=w} \left( f^1(x_i + dx, y_i + dy) - f^2(x_j + dx, y_j + dy) \right)^2$$

A chaque itération, on calcule :

$$P_{ij}^n = \frac{\tilde{P}_{ij}^n}{\sum_k \tilde{P}_{ik}^n}$$

Avec :  $\tilde{P}_{ij}^n = P_{ij}^{n-1} \left( A + B q_{ij}^{n-1} \right)$

et

$$q_{ij}^{n-1} = \sum_k \sum_l P_{kl}^{n-1}$$

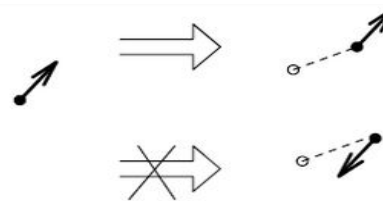
Avec  $k$  voisin de  $i$  et  $l$  voisin de  $j$ , tels que :

$$\| (x_i, y_i) - (x_k, y_k) \| \leq D_{\max}$$



et

$$\| V_{ij} - V_{kl} \| \leq V_{\max}$$



où  $V_{ij}$  est le flot optique du jeton  $i$  s'il est matché avec le jeton  $j$  dans l'autre frame

# Algorithme ITER\_TOKEN

**INPUT** : Une séquence temporelle de n images  $f_1, \dots, f_n$  avec le même nombre de primitives ou jetons à apparier dans chaque frame.

Pour chaque paire de frames consécutives, et pour chaque paire de jetons i et j d'une frame à l'autre, calculer jusqu'à stabilisation

$$P_{ij}^0 = \frac{1}{c + w_{ij}}$$

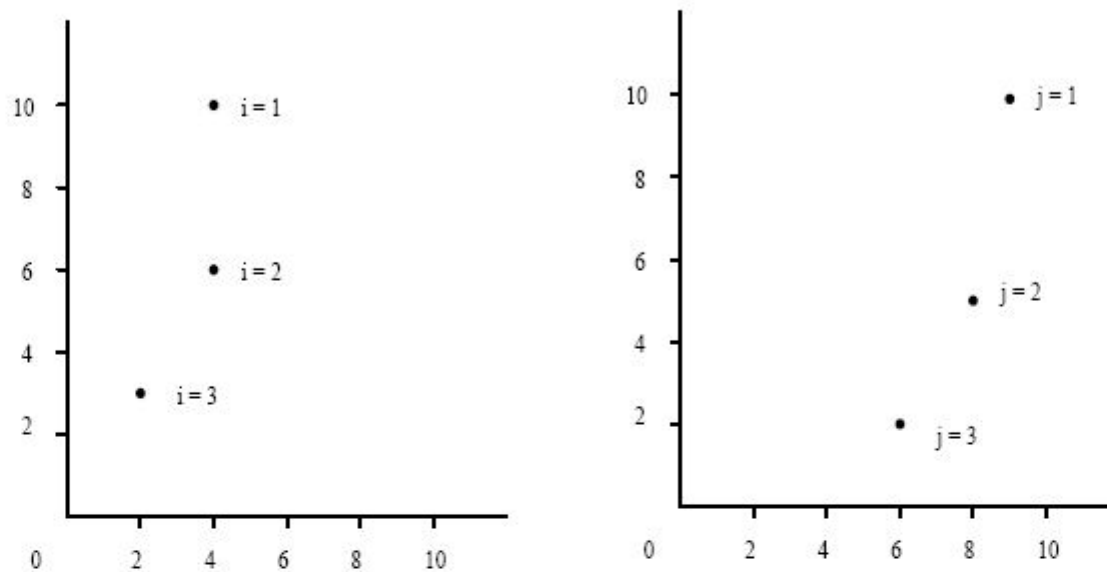
$$P_{ij}^n = \frac{\tilde{P}_{ij}^n}{\sum_k \tilde{P}_{ik}^n}$$

$$\text{avec } \tilde{P}_{ij}^n = P_{ij}^{n-1} (A + Bq_{ij}^{n-1})$$

et

$$q_{ij}^{n-1} = \sum_k \sum_l P_{kl}^{n-1}$$

**OUTPUT** : le flot optique ou la correspondance de mouvement associé à l'ensemble des jetons de frame en frame.



(a)

$$\begin{array}{l}
 ((4,10) \quad ((5,0),.7)((4,-5),.25)((2,-8),.05)) \\
 ((4,6) \quad ((5,4),.5)((4,-1),.3)((2,-4),.2)) \\
 ((2,3) \quad ((7,7),.3)((6,2),.35)((4,-1),.2))
 \end{array}$$

(b)

$$q_{ij}^0 = \begin{bmatrix} .3 & .2 & 0 \\ .35 & .9 & .25 \\ 0 & .5 & .3 \end{bmatrix}, \quad \tilde{P}_{ij}^1 = \begin{bmatrix} .84 & .225 & .015 \\ .675 & .9 & .21 \\ .09 & .63 & .24 \end{bmatrix}, \quad P_{ij}^1 = \begin{bmatrix} .76 & .2 & .013 \\ .378 & .5 & .11 \\ .09 & .65 & .25 \end{bmatrix}.$$

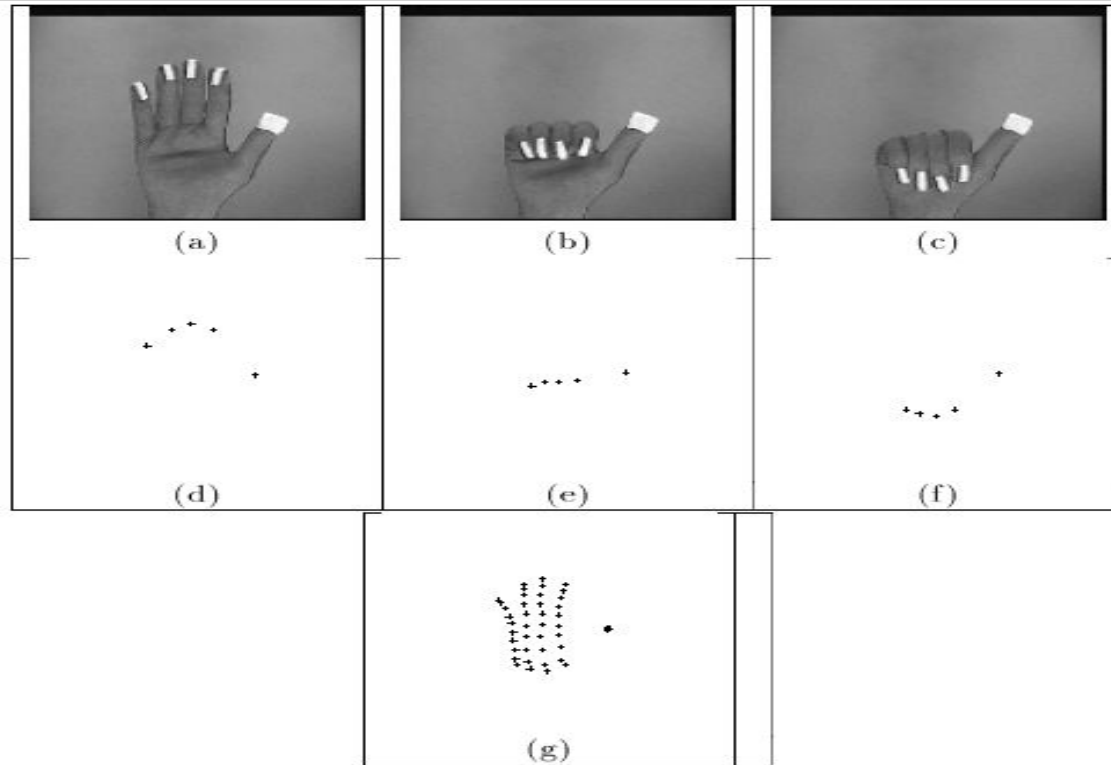
(c)

**Illustration de la méthode de Barnard et Thompson avec  $A=.3$ ,  $B=3$  et  $V_{\max} = 2$ , et  $D_{\max}=4$**

# Matching multi-frames : Tracking avec jetons :

## Algorithme non-itératif **déterministe** (voir le livre en ligne de M. Shah)

Il s'agit de suivre des jetons dans plusieurs frames successives et d'utiliser un mapping initial entre les deux premières frames pour propager cette information et adapter le mappings suivants en fonction de cette information *a priori*.



Trajectory generation. (a)-(c) Image sequence – selected images shown. (d)-(f) Respective fingertip points for images. (g) Fingertip trajectories.

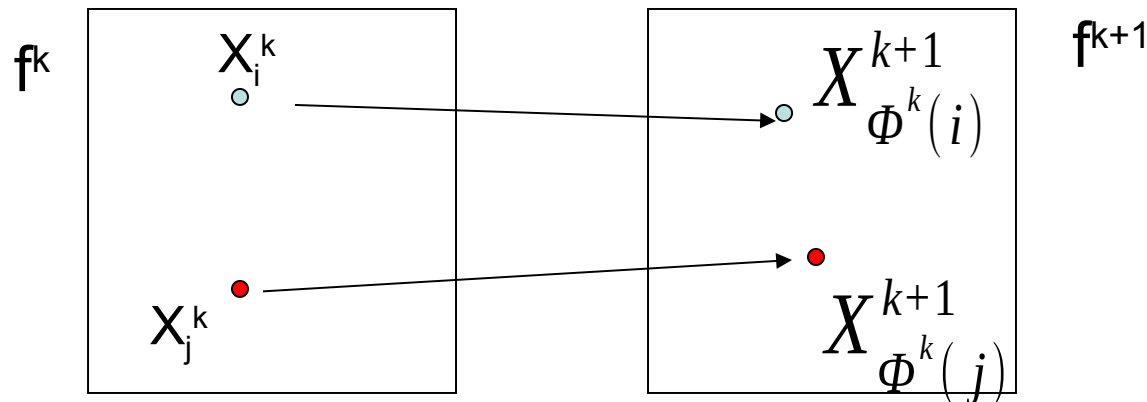
Soit une séquence de  $n$  frames  $f^1, f^2, \dots, f^n$ .

On suppose que les  $m$  jetons d'intérêt (détecteur de Harris pour les jonctions par exemple) ont été détectés.

Ainsi, chaque frame  $f^i$  est réduit à un ensemble de  $m$  points.

$X_i^j$  est le vecteur de coordonnées 2D du  $i^{\text{ème}}$  points dans la  $j^{\text{ème}}$  frame.

Le but est de trouver une correspondance bi-univoque  $\Phi^k$  entre les points de la frame  $k$  et ceux de la frame  $k+1$ .



Il n'est pas irréaliste de supposer que dans l'espace et sur des intervalles de temps petits, les objets se déplacent sur de petites distances. Et que leur mouvement correspondant est régulier ou uniforme : contrainte (a).

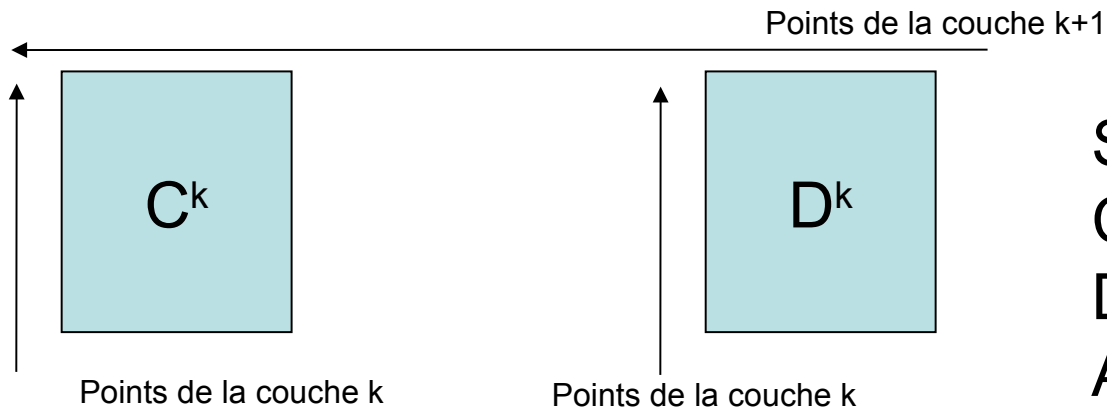
La régularité du mouvement implique un changement minime de la vitesse du point. C'est à dire que le point ne peut changer sa direction et sa vitesse de déplacement simultanément : contrainte (b).

Ainsi, les points vont suivre un certain chemin dit "*proximal uniform path*". La découverte de ce chemin régulier optimal va utiliser la fonction de coût ("*proximal uniformity function*")  $\delta$  définie dans ce qui suit.

Soit les matrices des Changements Relatifs de la frame  $k-1$  à  $k+1$   $C_r^k \in M_{m \times m}$  et des Déplacements Relatifs de la frame  $k$  à la frame  $k+1$   $D_r^k \in M_{m \times m}$ .

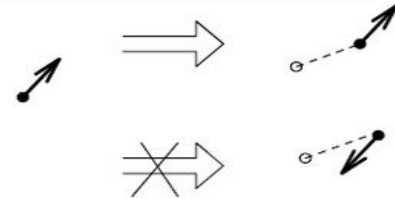
On placera en ligne les points de la frame de départ et en colonne les points de la frame d'arrivée.





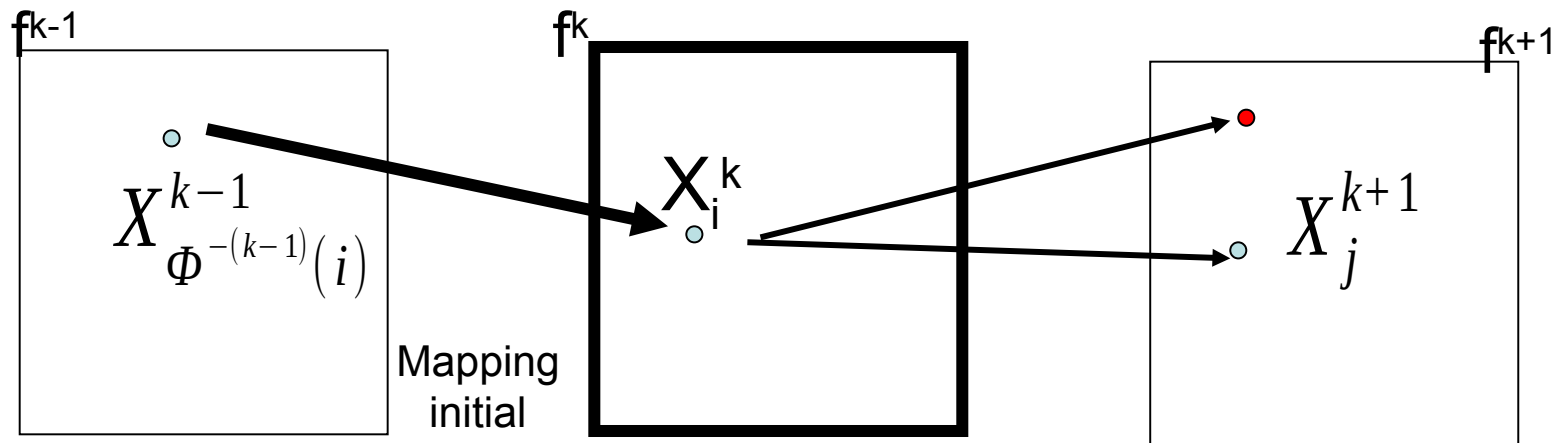
Soit les matrices de  
Changement C et  
Déplacement D  
Absolus

$$C^k[i,j] = \|\overline{X_{\Phi^{-(k-1)}(i)}^{k-1}} X_i^k - \overline{X_i^k X_j^{k+1}}\|$$



et

$$D^k[i,j] = \|\overline{X_i^k X_j^{k+1}}\|$$



Soit

$$C_{r^k}[i,j] = \frac{C^k[i,j]}{\sum_u \sum_v C^k[u,v]}$$

*et*

$$D_{r^k}[i,j] = \frac{D^k[i,j]}{\sum_u \sum_v D^k[u,v]}$$

Soit  $M \in M_{m \times m}$  la matrice résultante :

$$M^k [i,j] = C_{r^k} [i,j] + D_{r^k} [i,j]$$

Trajectoire lisse et uniforme :  
Contrôle des Changements du vecteur vitesse

Proximal Match

Cette matrice code la fonction de coût  $\delta$  pour une association  $(i,k)$  avec  $(j,k+1)$  sachant l'association optimale précédente  $(\Phi^{-(k-1)}(i), k-1)$  avec  $(i,k)$  :

$$M^k [i,j] = \delta \left( X_{\Phi^{-(k-1)}(i)}^{k-1}, X_i^k, X_j^{k+1} \right)$$

En estimant quantitativement les critères de régularisation suivants :

- La vitesse ne change pas beaucoup entre deux frames successives;
- La direction ne change pas beaucoup entre deux frames successives;
- Le déplacement d'un point entre deux frames successives tend à être petit;

Dans cette formulation, on suppose qu'un premier mapping  $\Phi^1$  entre la frame 1 et la frame 2 est disponible. On peut utiliser l'algorithme à jeton itératif précédent, par exemple. Ensuite l'algorithme va prolonger la trajectoire de frame en frame en utilisant ces contraintes de régularités initiales.

Le mapping  $\Phi^k$  est déterminé en fonction des mappings précédents de sorte à minimiser la fonction de coût globale sur toute la trajectoire :

$$C = \sum_k \delta(X_p^{k-1}, X_q^k, X_r^{k+1})$$

# Algorithme GREEDY\_TOKEN

## INPUT :

- Une séquence temporelle de  $n$  images  $f_1, f_2, \dots, f_n$ .
- Un ensemble de  $m$  jetons dans chaque frame.
- Une correspondance initiale entre les  $m$  jetons de la frame 1 et les  $m$  jetons de la frame 2 donnée par l'algorithme ITER\_TOKEN par exemple.

For  $k = 2$  to  $n - 1$  do

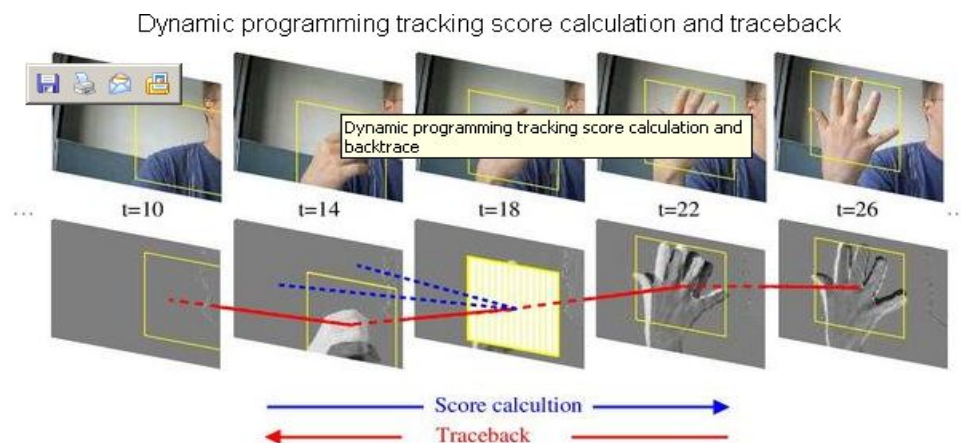
- (a) Construct  $M$  an  $(m \times m)$  matrix, with the points from  $k$ th frame along the rows and points from  $(k + 1)$ th frame along the columns. Let  $M[i, j] = \delta(X_p^{k-1} X_i^k X_j^{k+1})$ , when  $\Phi^{k-1}(p) = i$ .
- (b) for  $a = 1$  to  $m$  do
  - i. Identify the minimum element  $[i, l_i]$  in each row  $i$  of  $M$ .
  - ii. Compute *priority matrix*  $B$ , such that  $B[i, l_i] = \sum_{j=1, j \neq l_i}^m M[i, j] + \sum_{k=1, k \neq i}^m M[k, l_i]$  for each  $i$ .
  - iii. Select  $[i, l_i]$  pair with highest *priority* value  $B[i, l_i]$ , and make  $\Phi^k(i) = l_i$ .
  - iv. Mask row  $i$  and column  $l_i$  from  $M$ .

## OUTPUT : les trajectoires de la séquence d'images

This algorithm has the nice property that it will pick the *least cost assignment* if there are just two points in the frame. Consider the matrix  $M$ , with  $M[1,1] = 0.6$ ,  $M[1,2] = 0.3$ ,  $M[2,1] = 0.7$ , and  $M[2,2] = 0.2$ . Minimum along row 1 is element  $[1,2]$  with value 0.3, while the minimum along row 2 is element  $[2,2]$  with value 0.2. Therefore,  $B[1,2] = (0.6 + 0.2) = 0.8$ , and  $B[2,2] = (0.7 + 0.3) = 1.0$ . Now,  $B[2,2] > B[1,2]$ , hence we choose correspondence  $(2,2)$  first. Then, mask row 2 and column 1 with a high value. Next we pick the only assignment possible  $[1,1]$ . For this assignment  $\delta = M[1,1] + M[2,2] = 0.6 + 0.2 = 0.8$ , which is the least possible for this configuration.

Quand le nombre de jetons varie, on adapte ce cadre algorithmique à la programmation dynamique (cf. reconnaissance de la parole et de l'écrit manuscrit)  
[https://helios2.mi.parisdescartes.fr/~lomn/Cours/CV/SeqVideo/Articles/Tracking\\_Using\\_Dynamic\\_Programming\\_for\\_A.pdf](https://helios2.mi.parisdescartes.fr/~lomn/Cours/CV/SeqVideo/Articles/Tracking_Using_Dynamic_Programming_for_A.pdf)

- Ces techniques séparent complètement le processus de segmentation/détection des primitives de leur suivi.



Remarque : Ces conclusions intermédiaires sont tirées de la thèse de Auguste Genovesio traitant du tracking de spot fluorescents en imagerie médicale disponible en ligne et d'un chapitre en particulier qui propose un bon bilan des méthodes de tracking ponctuel ou de forme (les références se trouvent dans le document)

<https://helios2.mi.parisdescartes.fr/~lomn/Cours/CV/SeqVideo/CoursComplements/GenovesioThesis.pdf>

## **Avantages**

Ces méthodes présentent des avantages qui ont déjà permis leur implémentation au sein de logiciels commerciaux.

1. Elle est efficace sur des flots de points bien détectés.
2. Elle offre une relative simplicité de mise en oeuvre

## **Inconvénients**

Ces méthodes présentent aussi deux inconvénients pour notre application :

1. Elle repose sur une hypothèse de régularité mal adaptée à de multiples objets dont le mouvement varie entre eux et dans le temps
2. Elle ne prend pas en compte les fusions ou séparations d'objets en cas de sur ou sous détection.
3. Elle est sensible aux fausses détections

# Matching multi-frames : Tracking : Filtrage de Kalman

## Définition :

Il s'agit de mettre en correspondance des primitives sur une séquence d'images longue en associant des incertitudes..

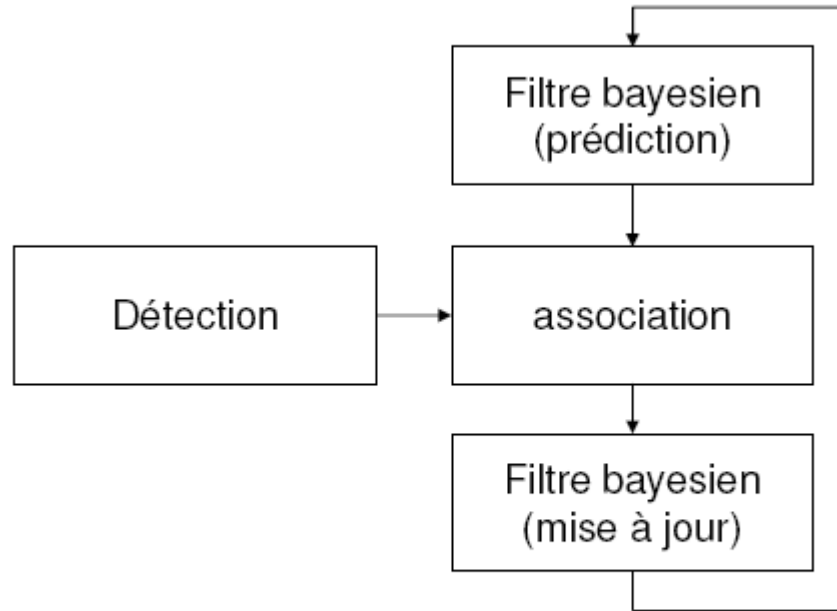
Il s'agit comme précédemment d'utiliser le travail de mise en correspondance passé pour prédire la mise en correspondance à venir dans l'hypothèse de trajectoires continues.

Pour cela, il existe un cadre théorique bien établi venant des techniques d'optimisation ; le filtre de Kalman.

Dans nos problématiques de vision, un filtre de Kalman est mis en place sous la forme d'un algorithme récursif qui estime la position et l'incertitude d'un point caractéristique mobile dans la frame suivante à partir des frames précédentes. En d'autres termes, on cherche la primitive dans cette frame et la taille de la région autour de la primitive prédite pour être sûre de trouver la dite primitive avec une certaine confiance.



## Cadre de modélisation bayésienne



**FIG. 2.4:** Schema general du suivi bayésien



Math

Voir document [\*kalma.HMM.ps\*](#)

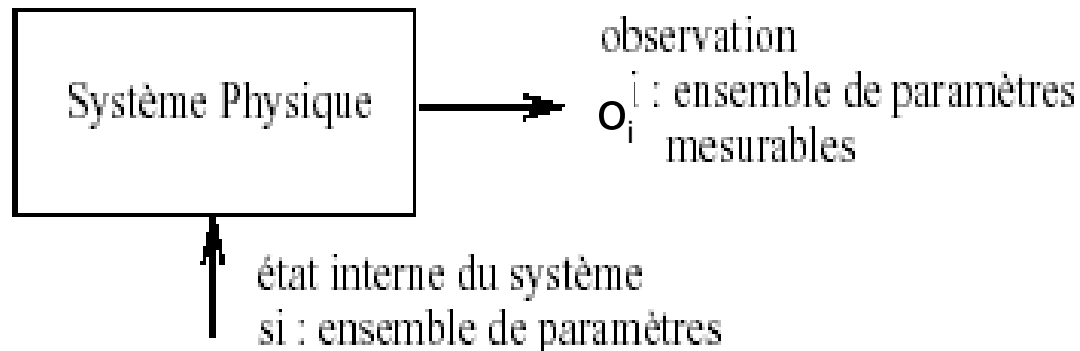
L'estimée résultat est *optimal au sens statistique* : sur un grand nombre d'expériences, le filtre de Kalman, **s'il est bien conçu**, serait meilleur, en moyenne, que les estimées résultats de tout autre filtre de prédiction sous l'hypothèse d'un système linéaire et de bruit blanc gaussien.

Si le bruit est non gaussien, le filtre de Kalman est encore le meilleur filtre linéaire non biaisé.

**[P.S. Maybeck, *Stochastic Models, Estimation, and Control, Vol I. Academic Press, New York, 1979*]**

# Principe

- Un système physique.
- **Distinction** entre paramètres:
  - qui modélisent le système;
  - que l'on peut mesurer.
- Estimer les premiers à partir des seconds.



# Prise en compte des incertitudes

- Les mesures sont *imprécises/incertaines*
- Le modèle est *simpliste donc ....*
- Dans ce formalisme statistique, on fait une estimation  $\hat{s}_i$  de l'état courant  $s_i$  à partir des observations  $o_i$ .
- On veut savoir la *fiabilité* d'une estimation  $\hat{s}_i$  de l'état courant  $s_i$ .

→ ce que le filtre de Kalman permet.

# Formalisation: mesure et vecteur d'états

- On a accès a:  $\hat{o}_i = o_i + \eta_i$
- $\eta_i$  bruit blanc additif et:

$$R_i = E[\eta_i \cdot \eta_i^t]$$

Les notes de Boutayna le mois i

Etat de concentration du professeur pour le mois i

- Evolution du modèle:

$$s_{i+1} = h_i(s_i) + n_i$$

Etat de connaissance de Boutayna le mois i+1

- $n_i$  bruit blanc additif et :

$$Q_i = E[n_i \cdot n_i^t]$$

Etat de concentration de Boutayna pour le mois i

# Formalisation: mesure et vecteur d'états

- Filtrage de Kalman: processus itératif d'estimation du vecteur  $s_i$
- Estimation à l'instant  $i$ :

$$\hat{s}_i \text{ associée à } P_i = E[(\hat{s}_i - s_i)(\hat{s}_i - s_i)^t]$$

- $P_i$  traduit la *confiance* que l'on a en l'estimation
- Si on connaît  $\hat{s}_{i-1}$ , statistiquement parlant la meilleure estimation possible de  $\hat{s}_i$  est:

$$\hat{s}_{i/i-1} = h_i(\hat{s}_{i-1})$$

# Formalisation: les équations de mesures, ou de quoi dispose-t-on en pratique ?

- du Lien entre les observations et le vecteur d'état:

$$f_i(o_i, s_i) = 0$$

$$f_i: \mathbb{R}^{m_i} \times \mathbb{R}^n \rightarrow \mathbb{R}^{p_i}$$

- Cas linéaire:  $\hat{o}_i = F_i \cdot s_i + \eta_i$

# Formalisation: les équations de mesures

- Dans le cas non-linéaire, on s'y ramène: *développement de Taylor* à l'ordre 1 de  $f_i$  :

$$f_i(\hat{o}_i, s_i) = f_i(o_i, \hat{s}_{i/i-1}) + \frac{\partial f_i}{\partial o_i} \cdot (\hat{o}_i - o_i) + \frac{\partial f_i}{\partial s_i} \cdot (s_i - \hat{s}_{i/i-1}) + \dots$$

- et on a bien:

$$z_i = M_i \cdot s_i + w_i$$

avec

$$\hat{o}_i = F_i \cdot s_i + \eta_i$$

en lieu et place de

$$z_i = -f_i(o_i, \hat{s}_{i/i-1}) + \frac{\partial f_i}{\partial s_i}(o_i, \hat{s}_{i/i-1}) \cdot \hat{s}_{i/i-1}$$

$$M_i = \frac{\partial f_i}{\partial s_i}(o_i, \hat{s}_{i/i-1})$$

$w_i$ : mesure l'incertitude sur la mesure

$$w_i = \frac{\partial f_i}{\partial o_i}(o_i, \hat{s}_{i/i-1}) \cdot (\hat{o}_i - o_i) = \frac{\partial f_i}{\partial o_i}(o_i, \hat{s}_{i/i-1}) \cdot \eta_i$$



# L'algorithme

3 étapes, propagation des incertitudes.

• Initialisation:  $(\hat{s}_0, P_0)$

• Prédiction (grâce au modèle système) :

$$\begin{aligned}\hat{s}_{i/i-1} &= h_i(\hat{s}_{i-1}) \\ P_{i/i-1} &= \frac{\partial h_i}{\partial s_i}(\hat{s}_{i-1}) \cdot P_{i-1} \cdot \frac{\partial h_i}{\partial s_i}(\hat{s}_{i-1})^t + Q_{i-1}\end{aligned}$$

• Calcul du gain de Kalman:

$$K_i = P_{i/i-1} \cdot M_i^t \cdot (M_i \cdot P_{i/i-1} \cdot M_i^t + W_i^t)^{-1}$$

• Mise à jour (grâce aux mesures) :

$$\begin{aligned}\hat{s}_i &= \hat{s}_{i/i-1} + K_i \cdot (z_i - M_i \cdot \hat{s}_{i/i-1}) \\ P_i &= (I - K_i \cdot M_i) \cdot P_{i/i-1}\end{aligned}$$



Un pas du filtre.

# L'algorithme : interprétation

- Compromis entre la contribution de la prédiction et de la mesure:

$$\hat{s}_i = \overbrace{\hat{s}_{i/i-1}}^{\text{prédiction}} + K_i \cdot (\overbrace{z_i - M_i \hat{s}_{i/i-1}}^{\text{inoxation}})$$

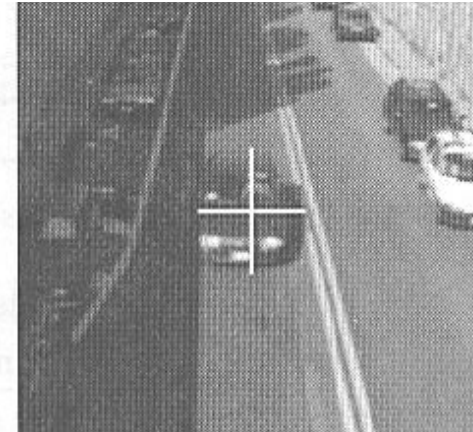
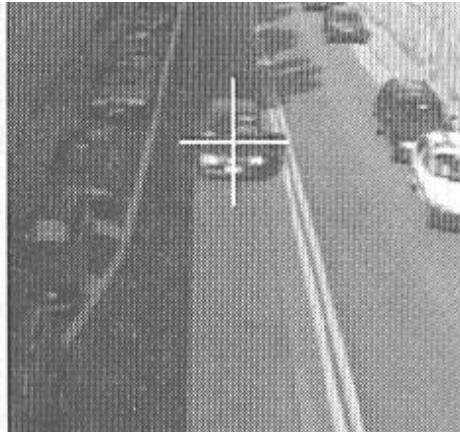
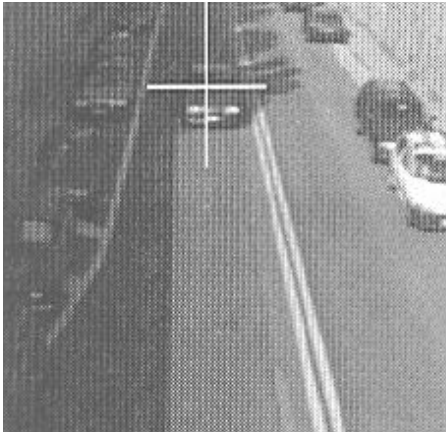
- On peut réécrire:  $K_i = P_i \cdot M_i^t \cdot W_i^{-1}$
- Grande incertitude sur le vecteur d'état donc  $P_i$  « grande »,  $K_i$  « grand » donc innovation favorisée.
- Grande incertitude sur la mesure donc  $W_i$  « grande »,  $K_i$  « petit », donc prédiction favorisée.

Formalisons dans ce cadre le problème du tracking.

Une nouvelle frame de la séquence d'image est acquise et traitée à chaque instant  $t_k = t_0 + k$  où  $k$  est un entier naturel.

$k$  est suffisamment petit pour considérer que le mouvement est linéaire d'une frame à l'autre.

On considère seulement un point primitive,  $p_k = [x_k, y_k]^T$ , dans la frame acquise à l'instant  $t_k$ , se déplaçant avec la vitesse  $v_k = [v_{x,k}, v_{y,k}]^T$ .



## Le modèle du système : ou prédiction du vecteur d'état

On décrit le mouvement dans le plan image avec un vecteur d'état  $\mathbf{s}_k = [x_k, y_k, v_{x,k}, v_{y,k}]^T$ .

En supposant un intervalle d'échantillonnage suffisamment petit (et donc une vélocité de la primitive constante entre 2 frames), le modèle système s'écrit dans le cadre du filtre de Kalman :

$$p_k = p_{k-1} + v_{k-1} + \xi_{k-1}$$

$$v_k = v_{k-1} + \zeta_{k-1}$$

où  $\xi_{k-1}$  et  $\zeta_{k-1}$  sont des bruits blancs gaussiens modélisant le bruit du système .

En terme de vecteurs d'état  $\mathbf{s}_k$  cela s'écrit :

$$\mathbf{s}_k = \mathbf{H}_{k-1} \mathbf{s}_{k-1} + \mathbf{n}_{k-1}$$

$$\text{avec } \mathbf{H}_{k-1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ et } \mathbf{n}_{k-1} = \begin{bmatrix} \xi_{k-1} \\ \zeta_{k-1} \end{bmatrix}$$

## Le modèle de mesures

En terme de mesures, on suppose qu'un extracteur rapide de primitive est disponible et estime  $z_k$ , la position du point primitive  $p_k$  pour chaque frame de la séquence. Ainsi le modèle de mesure du filtre de Kalman devient :

$$z_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_k \\ v_k \end{bmatrix} + w_k = Ms_k + w_k$$

avec  $w_k$  bruit blanc Gaussien centré  
modélisant le bruit de mesure

Remarque : dans le cas linéaire :  $R_k = W_k$ ,  $M_k = F_k$  et  $z_k = \hat{o}_k$

### Hypothèses et Positions du problème

Sous les hypothèses du filtre linéaire de Kalman, et étant données les observations bruitées  $z_k$ , calculer la meilleure estimée de la position de la primitive et de sa vitesse à l'instant  $t_k$  ainsi que leurs **incertitudes** associées.

# Algorithme KALMAN\_TRACKING

**INPUT :** L'entrée est formée, à chaque instant  $t_k$ , des deux matrices de covariance des bruits du système et de mesure à l'instant  $t_{k-1}$ ,  $Q_{k-1}$  et  $W_{k-1}$  respectivement, plus les deux matrices invariantes temporellement d'état d'une part,  $H$ , et de mesure,  $M$ , d'autre part, ainsi enfin que des mesures de position à l'instant  $t_k$ ,  $z_k$ . Les entrées de  $P_0$  sont fixées à de très fortes valeurs arbitraires.

$$\hat{s}_{k/k-1} = H_{k-1} \hat{s}_{k-1}$$

Prédiction modèle

$$P_{k/k-1} = H_{k-1} P_{k-1} H_{k-1}^T + Q_{k-1}$$

$$K_k = P_{k/k-1} M_k^T (M_k P_{k/k-1} M_k^T + W_k)^{-1}$$

Gain

$$\hat{s}_k = \hat{s}_{k/k-1} + K_k (z_k - M_k H_{k-1} \hat{s}_{k-1})$$

MAJ de la prédiction  
modèle par le biais des  
mesures

$$P_k = (I - K_k) P_{k/k-1} (I - K_k)^T + K_k W_k K_k^T$$

**OUTPUT :** les estimations optimales des position et vitesse à l'instant  $t_k$ ,  $\hat{s}_k$ , et leurs incertitudes données par les éléments diagonaux de  $P_k$ .

Le filtre quantifie l'incertitude sur l'estimée de l'état, sous la forme des éléments diagonaux de la matrice de covariance d'état  $P_k$

Cette information permet au détecteur de primitives de dimensionner automatiquement la région de recherche de la primitive dans la frame suivante.

Cette région de recherche est centrée sur la meilleure position estimée et sa largeur est proportionnelle à l'incertitude.

Dans un filtre correctement conçu, la valeur de cette incertitude décroît rapidement avec le temps et la région de recherche se rétrécit d'autant avec le temps.

Dans l'exemple présenté, la taille des croix indique la zone d'incertitude dans laquelle rechercher la primitive dans la frame suivante.

**Remarque** : la matrice de covariance d'état ne dépend pas des mesures. Donc, si la dépendance temporelle de  $H_k$ ,  $M_k$ ,  $Q_k$  et  $R_k$  est connue,  $P_k$  peut être calculée off-line et ensuite être approximée par une fonction en escalier. Cette propriété peut être cruciale pour les implémentations temps réel du filtre de Kalman.

Deux problèmes se posent pour l'implémentation de cet algorithme :

### Les données manquantes

Le filtrage de Kalman repose sur la connaissance suivante :

- Le modèle du système et la matrice de covariance du bruit correspondant  $Q_k$
- Le modèle de mesures et la matrice de covariance du bruit correspondant  $R_k$  ou  $W_k$
- L'état initial du système (temps  $t_0$ ),  $\hat{s}_0$ , et la matrice de covariance d'état,  $P_0$

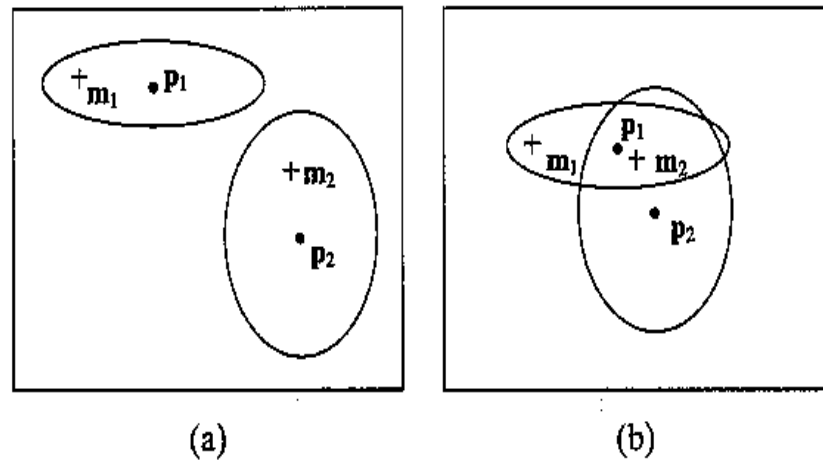
Cependant, nombre de données parmi ces dernières sont inconnues. Toutefois, en général, elles sont modélisables de façon simples linéairement. Il faut juste prendre garde à faire en sorte que  $Q$  et  $R$  soient comparables pour que la prédiction ou l'innovation ne l'emporte pas de façon injustifiée sur l'autre.

### L'association des données

En présence de plusieurs primitives images et de multiples mesures, quelle mesure observée doit être associée avec quelle primitive ?

Le cas de cibles en désordre et interférant.





(a) Disjoint search regions of two features, centered around the best position estimates  $p_1, p_2$ ; the measurements  $m_1, m_2$  are associated to the closest estimates. (b) If the search regions intersect, the minimum-distance criterion fails.

## Incertitude de Kalman.

C'est la région de l'espace d'état qui contient le véritable état avec une probabilité donnée.

Exemple : soit le vecteur d'état 2D  $e_k = [x_1, x_2]^T$ . Le filtre de Kalman calcule l'estimé d'état optimal,  $\hat{e}_k$ , comme le maximum de la densité de probabilité conditionnelle de  $e_k$  étant donnée la mesure  $z_k$ . Cette fonction de densité est supposée Gaussienne, si bien que son maximum coïncide avec sa moyenne.

### Conséquences pratiques :

- La région du plan centrée sur  $\hat{e}_k$  qui contient le véritable état avec une probabilité  $c^2$  est l'ellipse :  

$$(e - \hat{e}_k)(P_k)^{-1}(e - \hat{e}_k)^T \leq c^2$$
- Les axes de cette ellipse sont :  $\pm c \sqrt{\lambda_i} e_i, i=1,2$ , à partir des éléments propres de  $P_k$
- La variable  $(e - \hat{e}_k)(P_k)^{-1}(e - \hat{e}_k)^T$  a une distribution du Chi-2

### Utilisation pratique, pour le tracking :

- A  $t_{k-1}$ , la prédiction d'état du filtre à  $t_k$  est  $\hat{e}_{k/k-1} = H_{k-1} \hat{e}_{k-1}$  avec la matrice de covariance  $P_{k/k-1}$ . On diagonalise  $P_{k/k-1}$  pour obtenir ses vecteurs propres, et ne considère que les composantes du vecteur d'état donnant la position  $[x_1, x_2]^T$  de la primitive. On construit une ellipse d'incertitude (centrée en  $\hat{e}_{k/k-1}$  avec une probabilité désirée de  $(1-\alpha)$ ). Puis on cherche la primitive dans la frame  $k$  seulement dans l'ellipse. Une fois que votre détecteur de primitive a mesuré la position de la primitive à  $t_k$ , on calcule  $\hat{e}_k$  et  $P_k$ , ce qui vous permet de construire l'ellipse d'incertitude contenant la véritable primitive à  $t_k$  avec le niveau de probabilité choisi.

# Suivi de la trajectoire d'une particule dans le plan

[https://helios2.ml.parisdescartes.fr/~lomn/Cours/CV/SeqVideo/Articles/Region-Based\\_Tracking\\_in\\_an\\_Image\\_S](https://helios2.ml.parisdescartes.fr/~lomn/Cours/CV/SeqVideo/Articles/Region-Based_Tracking_in_an_Image_S)

- Le système: mouvement à accélération constante

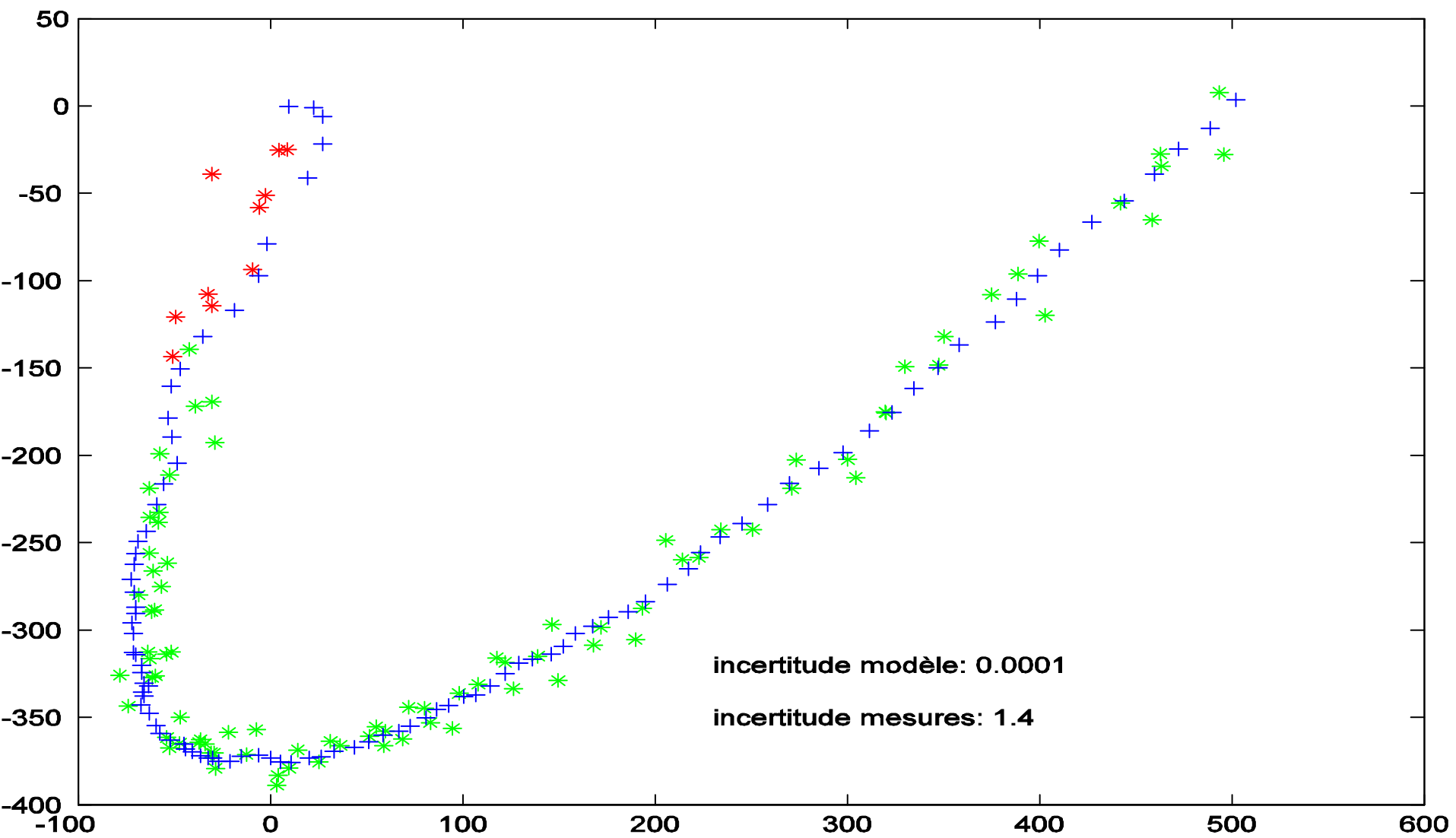
$$s_i \quad [x_i \quad \dot{x}_i \quad \ddot{x}_i \quad y_i \quad \dot{y}_i \quad \ddot{y}_i]^t \quad s_{i+1} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot s_i + n_i$$

- Equation de mesures:

$$z_i = [x_i \quad y_i]^t \quad z_i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot s_i + w_i$$

- Incertitude: Meyer a montré que

$$Q_i = \begin{bmatrix} \frac{\Delta t^5}{20} & \frac{\Delta t^4}{8} & \frac{\Delta t^3}{6} \\ \frac{\Delta t^4}{8} & \frac{\Delta t^3}{3} & \frac{\Delta t^2}{2} \\ \frac{\Delta t^3}{6} & \frac{\Delta t^2}{2} & \Delta t \end{bmatrix}$$



# Bilan sur la Technique de tracking de Kalman

## Avantages

1. Elle offre la possibilité de générer une prédiction d'une représentation générale de chaque objet
2. Elle permet de corriger cette prédiction au fur et à mesure de l'obtention des mesures.
3. Si les objets se modifient fortement dans le temps, elle permet de ne pas chercher à suivre chacun de leurs points précisément mais une description générale représentée par un vecteur d'état
4. Un vecteur d'état peut être composé de la position spatiale, du volume et de l'intensité moyenne d'une tache.

## Inconvénients

Nous verrons par la suite que ce schéma présente tout de même des inconvénients pour son adaptation à notre problématique :

1. La détection, qui ne peut pas être une étape considérée comme parfaite dans le cas du suivi de taches fluorescentes, viole systématiquement certaines hypothèses posées par les méthodes d'association courantes. Nous verrons dans la suite que ceci entraîne beaucoup d'erreurs.
2. L'utilisation de filtres bayesiens nécessite un certain nombre de connaissances *a priori*

# Suivi de primitives multi-frames : filtrage particulière ou la version évolutionnaire avec un peu de Bayes

Encore appelé algorithme CONDENSATION ou filtre bootstrap

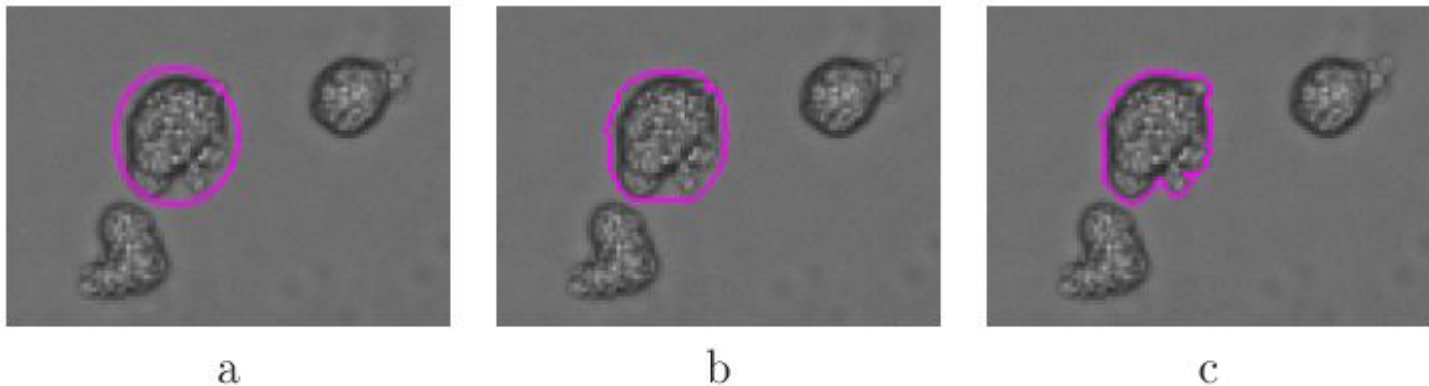
Permet de manipuler numériquement des modèles joints.

Repose sur un principe de ré-échantillonnage qui correspond à la sélection dans les algorithmes évolutionnaires (GA)

## Suivi de contours multi-frames : *Snake* ou Contours actifs

### Définition :

Il s'agit de suivre un contour d'objet légèrement mouvant sur une séquence d'images longue.



**FIG. 2.2:** Exemple de détection par contour actif : a) initialisation du *snake*, b) évolution, c) stabilisation du *snake* sur le contour de l'objet.

- D'abord une méthode de segmentation en image fixe

Zimmer *et al.*, 2002b; 2002a; Koschan *et al.*, 2003].

Le *snake* est une fonction paramétrique définie par  $v(s) = v(x(s), y(s))$  où  $x(s)$  et  $y(s)$  sont les coordonnées des points le long du contour et  $s \in [0, 1]$ . La fonctionnelle d'énergie à minimiser est alors de la forme :

$$E = \int_0^1 E_{snake}(v(s)) ds \quad (2.4)$$

$E_{snake}(v(s))$  est généralement composée de trois termes d'énergie

$$E_{snake}(v(s)) = E_{interne}(v(s)) + E_{image}(v(s)) + E_{contrainte}(v(s)) \quad (2.5)$$

où l'énergie *interne* est définie comme

$$E_{interne} = \alpha(s) \left| \frac{dv}{ds} \right|^2 + \beta(s) \left| \frac{d^2v}{ds^2} \right|^2 \quad (2.6)$$

où  $\alpha$  est le paramètre d'élasticité et  $\beta$  est le paramètre de rigidité. L'énergie dite *externe* ou *de l'image* est définie par le gradient (de Sobel par exemple) :

$$E_{image} = -\nabla F \quad (2.7)$$

Enfin, le dernier terme, désigne une *contrainte* telle que des rectifications manuelles sur le *snake* par exemple. Cette énergie peut être minimisée grâce à un schéma itératif obtenu depuis les équations d'Euler Lagrange.

## **Avantages**

Les contours actifs représentent une technique de suivi assez puissante dont voici les avantages.

1. Elle permet de disposer d'une description du contour de l'objet à chaque instant.
2. Elle offre la possibilité de suivre des objets qui se déforment légèrement dans le temps

## **Inconvénients**

Cependant, cette méthode n'est pas adaptée à tout type d'application et présente quelques inconvénients.

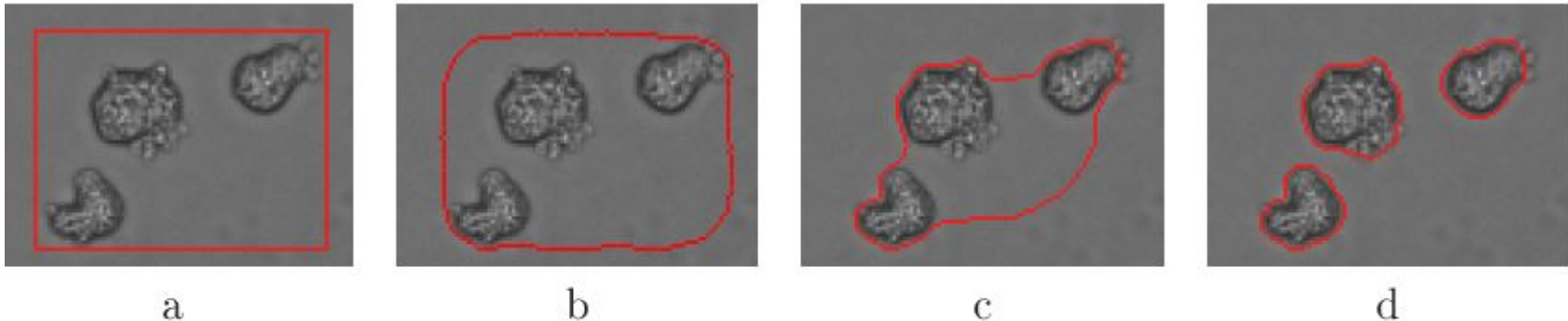
1. Elle ne supporte pas un écart trop élevé de position entre deux images.
2. Elle ne peut fonctionner que pour des objets dont les bords sont correctement définis (dans le cas de l'utilisation du gradient).
3. Elle ne gère pas le changement de topologie (entrée, sortie, séparation et regroupement d'objets). Ce point est cependant remis en question dans [McInemey and Terzopoulos 1999].
4. Elle ne gère pas le croisement d'objets et les occlusions.
5. Elle ne supporte pas un changement brutal de forme.
6. Elle est relativement sensible au bruit.



## Suivi de contours multi-frames : Level Set Methods

### Définition :

Il s'agit de suivre plusieurs contours d'objets dont le nombre varie et légèrement mouvant sur une séquence d'images longue.



**2.3:** Exemple de détection par ensemble de niveaux : a) initialisation de la fonction de *levelset* b) et c) évolution d) stabilisation autour des objets.

- Travaille dans le cadre des equations aux dérivées partielles
- D'abord une méthode de segmentation en image fixe

$$\frac{d}{dt} \begin{pmatrix} x \\ y \end{pmatrix} = F \vec{N}$$

## Avantages

Les ensembles de niveau représentent une bonne technique de suivi dont voici les avantages :

1. Elle offre un moyen de suivre des objets qui se déforment légèrement dans le temps.
2. Si on utilise un seul *levelset* pour tous les objets, elle permet de s'affranchir du problème de changement de topologie inhérent aux contours actifs. En effet, elle peut initialiser automatiquement des objets qui entrent dans le champ en cours de séquence.
3. Elle peut être étendue en 3D beaucoup plus naturellement que les contours actifs.

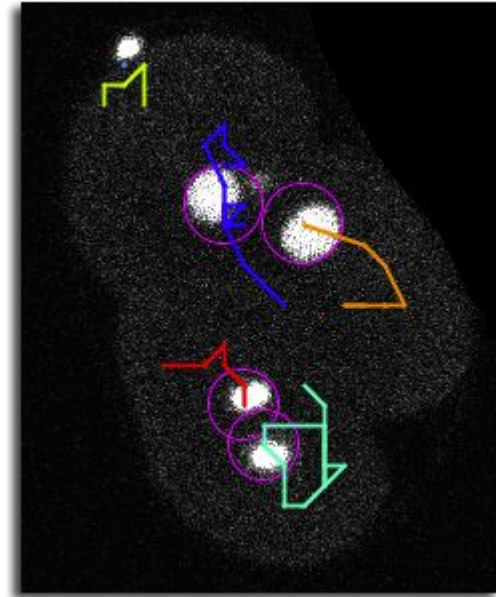
## Inconvénients

Cependant, cette méthode présente les inconvénients suivants :

1. Elle ne supporte pas un écart trop élevé de position.
2. Si on utilise un seul *levelset* pour tous les objets, elle gère mal le croisement d'objets et les occlusions.
3. Elle ne supporte pas un changement brutal de forme.
4. Elle est relativement sensible au bruit.
5. Etendue en 3D, elle nécessite un temps de calcul élevé.

# TrackMate (FiJi / Icy)

Spot tracker	Parameter	Value
<i>Linear motion tracker</i>	Initial search radius	10
	Search radius	7
	Max frame gap	3
<i>LAP Brownian motion</i>	Max linking distance	7
	Max gap-closing distance	10
	Max frame gap	3
<i>Nearest neighbor</i>	Max search distance	10



<https://imagej.net/media/plugins/trackmate/trackmate-manual.pdf> page 57

Associé à la problématique des « spot detection » / CellPose / StarDist

A titre d'exemples de données biologiques :

- 2 medical images biologiques 3D d'une séquence (comme t420.tif et t421.tif) (ISBI 2015 challenge

<http://celltrackingchallenge.net/> / FLUO\_C3DL\_MDA231 files) → recalage, SIFT, tracking etc.

- t026.tif and t027.tif dans <http://data.celltrackingchallenge.net/training-datasets/Fluo-C2DL-MSC.zip>

- Plus encore :

<https://public.celltrackingchallenge.net/documents/Naming%20and%20file%20content%20conventions.pdf>

Le temps manquant, je n'ai pas parlé de :



FIG. 1.1 – Exemple d'une transition brusque.

- Détection de changement dans des séquences vidéos (différences d'images sophistiquées) - de plans dans des films



FIG. 1.2 – Exemple d'une transition progressive de type fondu.

- Du tracking de référence de points de Harris :  
l'algorithme de Kanade Lucas et Tomassi (KLD)  
disponible dans beaucoup de bibliothèque en ligne

# Bibliographie

- <https://helios2.mi.parisdescartes.fr/~lomn/Cours/CV/SeqVideo/CoursComplements/GenovesioThesis.pdf>
  - <https://helios2.mi.parisdescartes.fr/~lomn/Cours/CV/SeqVideo/CoursComplements/KalmanParticle.pdf>
- “Use of Optimal Estimation Theory, in Particular the Kalman Filtering”, in Data Analysis and Signal Processing”, W. Cooper, Review of Scientific Instrumentation, vol. 57, pp. 2862-2869, 1986  
<https://helios2.mi.parisdescartes.fr/~lomn/Cours/CV/SeqVideo/Articles/Kalman1986.pdf>
- <https://helios2.mi.parisdescartes.fr/~lomn/Cours/CV/BABA/bookMubarakShah.pdf>
  - <https://imagej.net/media/plugins/trackmate/trackmate-manual.pdf>
- De façon générale, ici <https://helios2.mi.parisdescartes.fr/~lomn/Cours/CV/SeqVideo/CoursComplements/>