

# Formation et Analyse d'Images

James L. Crowley

ENSIMAG 3

2008/2009

Séance 11

16 janvier 2009

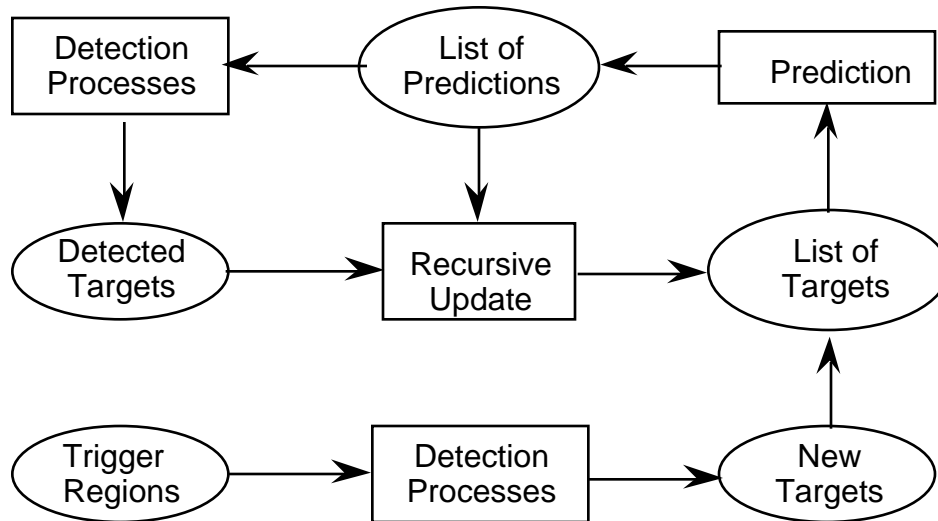
## Détection et Suivi de Blobs : le Filtre de Kalman

### Plan de la Séance :

Détection et Suivi des Blobs Gaussiens.....	2
L'architecture d'un système de suivi robuste.....	2
Les Blobs Gaussiens.....	2
Détection par couleur (rappelle).....	3
Calcul des moments :.....	3
Composantes principales.....	4
Suivi par Filtre de Kalman.....	5
Prédiction de l'évolution temporelle de l'état.....	7
Modèle du capteur.....	9
Aproximations des modèle non-linéaires.....	9
Validation.....	10
Mise à Jour.....	11
L'Innovation.....	12
Application de la Filtre de Kalman aux Suivi .....	13
Architecture.....	13
Prédiction de la taille et la position d'un cible.....	15
Estimation Bayésienne Robuste.....	17

## Détection et Suivi des Blobs Gaussiens

### L'architecture d'un système de suivi robuste



## Les Blobs Gaussiens

Les entités suivies sont une ensemble connexes de pixels sont de cibles, parfois appelées des "blobs".

On peut décrire une blob par un vecteur de caractéristiques "invariantes" à l'orientation grâce aux "moments" d'est pixels sorties d'une procédure de détection appliqué à une "ROI" (Region of Interest).

Le ROI est une rectangle englobante représentée par quatre paramètres : (u, l, b, r)

- u - "up" - le premier ligne du ROI.
- l - "left" - la première colonne du ROI.
- b - "bottom" - le dernier ligne du ROI
- r - "right" - La dernière colonne du ROI

Les procédures de détection fréquentent utilisé sont :

- 1) Ratio d'Histogramme de couleurs.
- 2) Différences d'image avec un fond adaptatif.
- 3) Différence d'image temporelle
- 4) Détection probabiliste calculée avec les caractéristiques locale (e.g. Dérivées de Gaussiens).

**Détection par couleur (rappelle)**

Pour chaque pixel  $C(i, j)$   $p(\text{objet} | C) = p(C | \text{objet}) \frac{p(\text{objet})}{p(C)}$

Soit  $M$  images de  $I \times J$  pixels. Ceci fait  $N = I \times J \times M$  Pixels.

Soit  $h(r, v)$ , l'histogramme de tous les  $N$  pixels.

Soit  $h_o(r, v)$ , l'histogramme des  $N_o$  pixels de l'objet "o".

$$p(\text{objet}) = \frac{M_o}{M}$$

$$p(C) = \frac{1}{M} h(C)$$

$$p(C | \text{objet}) = \frac{1}{M_o} h_o(C)$$

$$p(\text{objet} | C) = p(C | \text{objet}) \frac{p(\text{objet})}{p(C)} = \frac{1}{M_o} h_o(C) \frac{\frac{M_o}{M}}{\frac{1}{M} h(C)} = \frac{h_o(C)}{h(C)}$$

**Calcul des moments :**

Soit  $w(i, j)$ , des valeurs issues d'un algorithme de détection dans une ROI de taille  $N \times M$

Somme des Pixels :

$$S = \sum_{i=1}^M \sum_{j=1}^N w(i, j)$$

**Premiers moments :**

$$\mu_i = \frac{1}{S} \sum_{i=1}^M \sum_{j=1}^N w(i, j) \cdot i \quad \mu_j = \frac{1}{S} \sum_{i=1}^M \sum_{j=1}^N w(i, j) \cdot j$$

Le premier moment est le centre de gravité de la forme :

**Deuxième moment :**

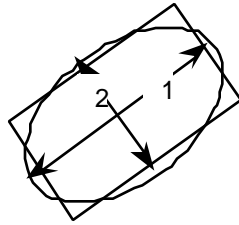
$$i^2 = \frac{1}{S} \sum_{i=1}^M \sum_{j=1}^N (w(i, j)) \cdot (i - \mu_i)^2 \quad j^2 = \frac{1}{S} \sum_{i=1}^M \sum_{j=1}^N w(i, j) \cdot (j - \mu_j)^2$$

$$ij^2 = \frac{1}{S} \sum_{i=1}^M \sum_{j=1}^N w(i, j) \cdot (i - \mu_i)(j - \mu_j)$$

Ceci permet de définir les "axes", majeur,  $\lambda_1$  et mineur,  $\lambda_2$ , de la forme par analyse des composantes principales du deuxième moment

$$C \hat{=} \begin{pmatrix} i^2 & ij^2 \\ ij^2 & j^2 \end{pmatrix}$$

## Composantes principales



Les deuxièmes moments sont "invariants" à l'orientation

Les axes sont calculés par une analyse en composantes principales de la matrice  $C$ . Il s'agit de trouver une rotation,  $\Phi$ , dans l'espace de caractéristiques  $\Phi C \Phi^T = \Lambda$  telles que  $\Lambda$  soit diagonale.

$$\Lambda = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \quad \text{tel que } 1 > 2 \quad \Phi = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

$$\Phi C \Phi^T = \Lambda = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \quad \Phi^T \Phi = \mathbf{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\Phi C \Phi^T \Phi = \Phi C_P = \Lambda \Phi = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

Les lignes du  $\Phi$  sont des vecteurs propres du  $C$ .

La longueur des axes majeurs et mineur est les valeurs propres de la matrice  $C$ .

$\theta$  est l'orientation de l'axe "majeur" et  $1 / 2$  est le rapport entre la longueur et la largeur.

$1 / 2$  est une caractéristique invariante de la taille et de l'orientation.

$$\hat{X}_t = (x, y, w, h, a)$$

où

- $x, y$  : Centre de Gravité.
- $h$  : "height" - valeur du premier vecteur propre
- $w$  : "width" - valeur du deuxième vecteur propre
- $a$  : angle - orientation du premier vecteur propre

Confiance :

$$CF_t \in [0, CF_{\max}]$$

$$CF = S / NM \cdot D$$

- $S$  = somme de valeurs de détection
- $NM$  : Surface du ROI
- $D$  : Valeur maximale d'une détection

## Suivi par Filtre de Kalman

Le filtre de Kalman est un estimateur récursif optimal pour l'état d'un système.

Son origine se trouve dans le problème de l'estimation de la trajectoire des satellites et des aéronefs. Dans un tel problème, on cherche à estimer la position  $(x, y, z)$  et la vitesse  $(v_x, v_y, v_z)$  à partir de l'observation de deux angles : Site et azimuth. ( , )

Dans la terminologie de Kalman : la position et vitesse est un vecteur d'état :  $X$  et

l'observation est  $Y$

$$X = \begin{pmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{pmatrix} \quad Y =$$

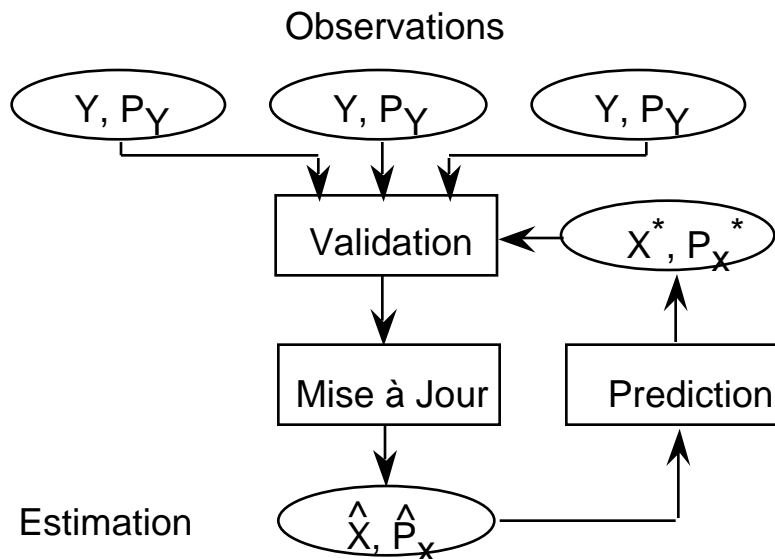
Pour chaque instant,  $t$ , le filtre donne une estimation  $\hat{X}_t$  ainsi que sa précision  $P_t$ , à partir de l'observation et l'estimation précédente,  $\hat{X}_{t-1}$ ,  $P_{t-1}$ , et l'observation  $Y$ , et sa précision,  $P_y$ . Les précisions sont des covariances.

$$\hat{X}_t, P_t := F \{ \hat{X}_{t-1}, P_{t-1}, Y, P_y \}$$

Ce permet d'estimer un vecteur de  $N$  variable aléatoire à partir de  $M < N$  observations. Les précisions sont des 2ieme moments de l'erreur.

Le filtre récursif est un processus cyclique avec 5 phases :

- 1) Prédiction de l'évolution temporelle de l'état.
- 2) Observation
- 3) Prédiction de l'observation
- 4) Validation de l'observation.
- 5) Mise à jour de l'estimation.



Techniques :

Prédictions:	Modèle Linéaire
validation :	Distance de Mahalanobis (Distance normalisée par Covariance)
Mise à Jour	Estimation linéaire

Notation:

Vecteur d'état a temp t :	$X_t$
Estimation du vecteur d'état :	$\hat{X}_t, \hat{P}_t$
Observation :	$Y, P_y$
Prédiction :	$X_t^*, P_t^*$
Modèle du capteur :	$H_X^Y$ avec précision $P_y$
Modèle du processus :	( t ) avec précision $Q_t$
Affectation :	:=
Définition :	

Le modèle de Kalman suppose que les estimations sont corrompues par un bruit B.

$$\hat{X} = X + B$$

Le bruit est une variable aléatoire de moyenne nul.  
La précision est le deuxième moment de ce bruit.

$$P = E\{ B B^T \} = E\{ (X - \hat{X})(X - \hat{X})^T \}$$

On ne peut connaître ni  $X$  ni  $B$ . Kalman nous fourni les estimations  $\hat{P}_t$

## Prédiction de l'évolution temporelle de l'état

La phase de prédiction projets  $\hat{X}_{t-1}$  et  $\hat{P}_{t-1}$  à temps  $t$  afin de produire  $X_t^*$  et  $P_t^*$

On utilise les dérivées temporelles de  $X$ .

Ces dérivées peuvent être inclus dans le vecteur  $\hat{X}_{t-1}$

Soit une propriété,  $\hat{x}$ , du vecteur  $\hat{X}$ , avec variance  $\hat{x}^2$ .

La prédiction de premier ordre  $x^*(t)$  exige la dérivée temporelle  $\hat{x}'(t-1)$ .

$$\hat{x}'_t \cong \frac{\hat{x}_t}{t}$$

Par série de Taylor, on peut écrire :

$$x^*_t := \hat{x}_{t-1} + \hat{x}'_{t-1} \cdot t + r$$

avec l'hypothèse que  $E\{r\} = 0$ .

Soit un vecteur de deux propriétés  $\hat{x}_1$  et  $\hat{x}_2$  avec une prédiction de premier ordre.

$$\hat{X}_t = \begin{pmatrix} \hat{x}_1 \\ \hat{x}_1' \\ \hat{x}_2 \\ \hat{x}_2' \end{pmatrix}$$

Le prédiction est :  $X^*_t := (t)X_{t-1} + R$

avec le matrice de transition,  $(t) = \begin{pmatrix} 1 & t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & t \\ 0 & 0 & 0 & 1 \end{pmatrix}$

La précision de  $X^*_t$  est la covariances les propriétés  $\hat{x}$

$$P_t^* \cong \begin{pmatrix} \hat{x}_1^2 & \hat{x}_1'x_1 & \hat{x}_2x_1 & \hat{x}_2'x_1 \\ \hat{x}_1x_1' & \hat{x}_1'^2 & \hat{x}_2x_1' & \hat{x}_2'x_1' \\ \hat{x}_1x_2 & \hat{x}_1'x_2 & \hat{x}_2^2 & \hat{x}_2'x_2 \\ \hat{x}_1x_2' & \hat{x}_1'x_2' & \hat{x}_2x_2' & \hat{x}_2'^2 \end{pmatrix}$$

où

$$xx' \cong E\{ (x - \hat{x})(x' - \hat{x}') \}$$

et

$$x'^2 \cong E\{ (x' - \hat{x}')^2 \}$$

La covariance  $P_t^*$  est prédite par  $P_t^* := (t)\hat{P}_{t-1} (t)^T + E\{ R_t R_t^T \}$

Le processus peut avoir les dérivés d'ordre supérieur à le modèle.

Ces dérivés donnent lieu au "résidu"  $R$ .

$R$  est supposée d'être moyenne nul.

Son deuxième moment indique la perte de précision dû au modèle linéaire.

$$X_t := X_{t-1} + \frac{X_{t-1} - X_{t-2}}{t} T + R_t.$$

avec

$$Q = E\{R_t R_t^T\}$$

Prédiction :

$$I: \quad X_t^* := (t) \hat{X}_{t-1} + R_t.$$

$$II: \quad P_t^* := (t) \hat{P}_{t-1} + Q_x$$



## Modèle du capteur

Un capteur projette l'état du monde vers un vecteur d'observation :

$$\text{III: } Y := \mathbf{H}_X^Y X_t$$

$$\text{IV: } \mathbf{P}_y = E\{W_t W_t^T\}$$

La matrice de  $\mathbf{H}_X^Y$  est un modèle linéaire du capteur.  
Les observations sont corrompues par les bruits aléatoires ainsi que une erreur d'approximation.  
Ceci est représenté par l'erreur du capteur :  $W_t$ .

On suppose que  $E\{W_t\} = 0$   
tandis que  $\mathbf{P}_y = E\{W_t W_t^T\}$

Par exemple, considère l'observation  $Y$  d'un état  $X_t$

$$Y = \mathbf{H}_X^Y X \Rightarrow \begin{matrix} y_1 \\ y_2 \end{matrix} = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{matrix} \begin{matrix} x_1 \\ x_1' \\ x_2 \\ x_2' \end{matrix}$$

deuxième exempl, la matrice de projection d'une caméra :

$$w X^i = \mathbf{M}_s^i X^s$$

## Aproximations des modèle non-linéaires

Souvent le vrai capteur n'est pas linéaire.

$$Y = F(X)$$

Dans ce cas, on peut faire une approximation linéaire avec la dérivée première, (la Jacobienne), calculé au tour de l'estimation actuelle.

$$\mathbf{H}_X^Y \quad \mathbf{J}_X^Y = \frac{F(\hat{X}_t)}{\hat{X}_t} = \begin{matrix} \frac{y_1}{x_1} & \frac{y_1}{x_2} & \frac{y_1}{x_3} & \frac{y_1}{x_4} \\ \frac{y_2}{x_1} & \frac{y_2}{x_2} & \frac{y_2}{x_3} & \frac{y_2}{x_4} \end{matrix}$$

et donc :

$$\begin{matrix} y_1 \\ y_2 \end{matrix} = \begin{matrix} \frac{y_1}{x_1} & \frac{y_1}{x_2} & \frac{y_1}{x_3} & \frac{y_1}{x_4} \\ \frac{y_2}{x_1} & \frac{y_2}{x_2} & \frac{y_2}{x_3} & \frac{y_2}{x_4} \end{matrix} \begin{matrix} x_1 \\ x_1' \\ x_2 \\ x_2' \end{matrix}$$

## Validation

Validation repose sur une mesure de similarité. Une telle mesure est la "Distance de Mahalanobis" ou "Différence normalisée par covariance"

Soit la prédiction  $X_t^*$  avec sa précision  $P_t^*$

et l'observation  $Y$  avec sa précision  $P_y$

Le distance de Mahalanobis est

$$V : d^2 = \frac{1}{2} \{ (\mathbf{H}_X^Y X_t^* - Y)^T (\mathbf{P}_t^* + \mathbf{P}_y)^{-1} (\mathbf{H}_X^Y X_t^* - Y) \}$$

Pour une scalaire :

$$d^2 = \frac{(x^* - y)^2}{2 \left( \frac{1}{x^2} + \frac{1}{y^2} \right)}$$

On considère que  $X_t^*$ ,  $P_t^*$  définie une probabilité d'une observation  $Y$ .

Si il y a  $K$  prédictions :  $X_{kt}^*$ ,  $P_{kt}^*$ , l'observation est associée a :

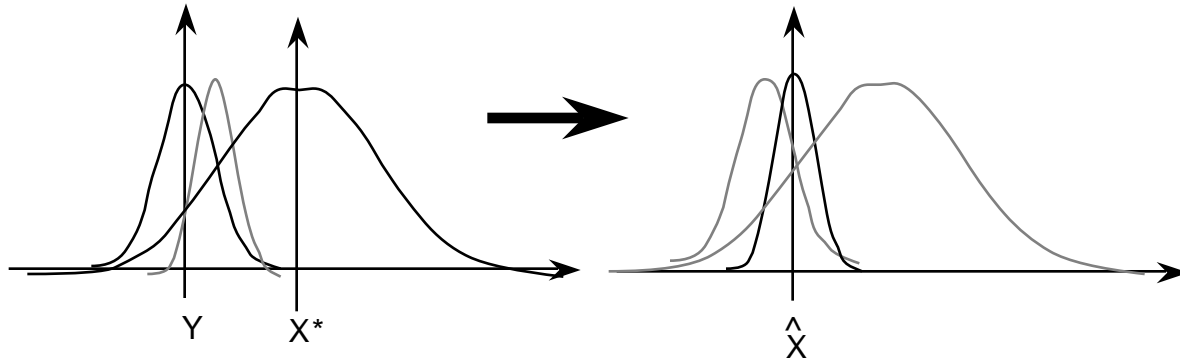
$$\text{Arg}_k \text{Min} \{ d_k^2 \} \text{ tel que } d_k \leq \text{seuil.}$$

Le seuil est un multiple de l'écart type.  
Son choix dépend du problème

## Mise à Jour

Contraint d'un modèle par une Observation

Soit un modèle  $X_t^*$ ,  $P_t^*$  et une observation  $Y$ ,  $P_y$



$Y$  fournit une contrainte sur  $X^*$ .

La nouvelle valeur est donnée par un moyen, pondérée par la variance:

Les variances combinées comme les résistances parallèles :

$$\hat{\sigma}_x^2 = \frac{\sigma_x^2 \sigma_y^2}{\sigma_y^2 + \sigma_x^2}$$

et

$$\hat{X}_{nt} = \hat{\sigma}_x^2 \left( \frac{X^*}{\sigma_x^2} + \frac{Y}{\sigma_y^2} \right)$$

Formulation Kalman (estimation récursif): Le Gain de Kalman

$$K = \frac{\sigma_x^2}{\sigma_y^2 + \sigma_x^2}$$

Puis :

$$\hat{\sigma}_x^2 = \sigma_x^2 - K \sigma_x^2$$

et

$$\hat{X} = X^* + K (Y - X^*)$$

En plusieurs dimensions, le gain de Kalman est une matrice.

Supposons que  $Y$  et  $X$  sont dans le même espace. (Même nombre de variables.

$$K := P_t^* (P_t^* + P_y)^{-1}$$

puis :

$$\hat{X}_t = X_t^* + K [Y - X_t^*].$$

et

$$\hat{P}_t = P_t^* - K P_t^*.$$

Et si  $X$  et  $Y$  ne sont pas du même repère,

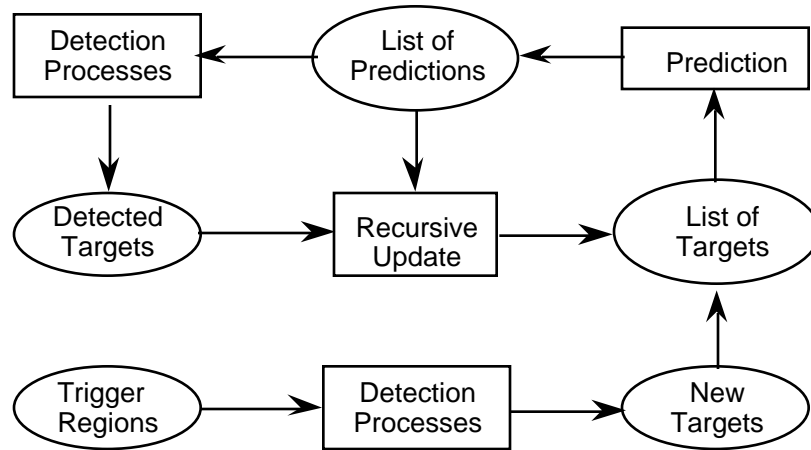
il faut la transformation  $H_X^Y$  et son inverse  $H_Y^X = H_X^Y T$

Équations VI, VII et VIII du Filtre de Kalman :



# Application de la Filtre de Kalman aux Suivi

## Architecture



The target state vector is

$$\hat{\mathbf{X}}_t = \begin{bmatrix} x \\ x' \\ y \\ y' \\ s_x \\ s_y \\ s_{xy} \end{bmatrix}$$

where

$x, y$  are the position of the target in pixels

$x', y'$  are the first temporal derivatives  $\frac{x}{t}$  and  $\frac{y}{t}$  of the position (in pixels/sec)

$s_x, s_y, s_s$  are the width, height and slant of the target measured by the square root of the second moments (Covariance) matrix.  $s_x = \sqrt{\hat{xx}^2}$   $s_y = \sqrt{\hat{yy}^2}$  ,  $s_s = \sqrt{\hat{xy}^2}$

$$CF_t \quad [0, CF_{max}]$$

Incertitude de la position

$$\mathbf{P} = \begin{bmatrix} \hat{xx}^2 & \hat{xy}^2 \\ \hat{xy}^2 & \hat{yy}^2 \end{bmatrix}$$

Incertitude de la vitesse

$$\mathbf{V} = \begin{bmatrix} \hat{x'x'}^2 & \hat{x'y'}^2 \\ \hat{x'y'}^2 & \hat{y'y'}^2 \end{bmatrix}$$

size uncertainty is

$$\mathbf{S} = \begin{pmatrix} \hat{\sigma}_{xx} & \hat{\sigma}_{xy} & \hat{\sigma}_{xs} \\ \hat{\sigma}_{yx} & \hat{\sigma}_{yy} & \hat{\sigma}_{ys} \\ \hat{\sigma}_{sx} & \hat{\sigma}_{sy} & \hat{\sigma}_{ss} \end{pmatrix}$$

total covariance matrix is :

$$\mathbf{P}_t = \mathbf{P} + \mathbf{V} \mathbf{S}$$

For each target, at each time,  $t$ , the tracker maintains an estimated state  $\hat{\mathbf{X}}_t$  as well as its precision  $\mathbf{P}_t$  and confidence factor,  $CF_t$ . Based on a previous state and precision,  $\hat{\mathbf{X}}_{t-1}$ ,  $\mathbf{P}_{t-1}$ ,  $CF_{t-1}$  and the observation  $Y_d$  from detection functions  $d$ , accompanied by the observed precision  $\mathbf{P}_d$ , and detection confidence  $CF_d$ .

$$\hat{\mathbf{X}}_t, \mathbf{P}_t, CF_t := F \{ \hat{\mathbf{X}}_{t-1}, \mathbf{P}_{t-1}, CF_{t-1}, Y_d, \mathbf{P}_d, CF_d \}$$

## Prédiction de la taille et la position d'un cible

Given a target at time  $t-1$ , the prediction equations predict its new position, and validation gate at time  $t$ . The general form of the prediction equations are :

$$\mathbf{X}_t^* := \Phi(t) \hat{\mathbf{X}}_{t-1} + \mathbf{R}_t \tag{1}$$

$$\mathbf{P}_t^* := \Phi(t) \hat{\mathbf{P}}_{t-1} \Phi(t)^T + \mathbf{Q}_x \tag{2}$$

Thus second moments,  $\mathbf{Q}$ , estimates the loss of precision due to higher order terms.

$$\mathbf{Q} = E \{ \mathbf{R} \mathbf{R}^T \}$$

When included in equation (2) this translates to an additive growth in the validation gate.

$$x_t = x_{t-1} + k \frac{y}{t} \quad y_t = y_{t-1} + k \frac{y}{t}$$

The constant  $k$  can be set by observing the change in width for a rectangular sheet of paper (or other calibration pattern).

For a zeroth order Kalman filter, the prediction matrix,  $\Phi(t)$ , takes on a trivial form:

$$\Phi = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

In this case our prediction equations are reduced to

$$\mathbf{X}_t^* := \hat{\mathbf{X}}_{t-1} \tag{3}$$

$$\mathbf{P}_t^* := \hat{\mathbf{P}}_{t-1} + \mathbf{Q}_x \tag{4}$$

For a first order Kalman filter the prediction matrix is:

$$\Phi(t) = \begin{pmatrix} 1 & t & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & k t & 1 & 0 & 0 \\ 0 & 0 & 0 & k t & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

A ROI based on twice the size plus twice the precision represents a reasonable size search region. Such a ROI is defined as

$$\begin{aligned} \text{left} &= x - 2 s_x - 2 \sigma_x \\ \text{right} &= x + 2 s_x + 2 \sigma_x \\ \text{top} &= y - s_y - 2 \sigma_y \\ \text{bottom} &= x + 2 s_y + 2 \sigma_y \end{aligned}$$

We assume image coordinates with positive y coordinates pointing down.

Précision de la position

$$\mathbf{P} = \mathbf{C} = \begin{pmatrix} \hat{\sigma}_{xx}^2 & \hat{\sigma}_{xy}^2 \\ \hat{\sigma}_{xy}^2 & \hat{\sigma}_{yy}^2 \end{pmatrix}$$

La précision englobe la taille ET les erreurs d'estimation.

Pour chaque "blob" a chaque instante, on maintient :

$\hat{\mathbf{X}}_t, \mathbf{P}_t, \mathbf{CF}_t$ .

$$\hat{\mathbf{X}}_t, \mathbf{P}_t, \mathbf{CF}_t := F \{ \hat{\mathbf{X}}_{t-1}, \mathbf{P}_{t-1}, \mathbf{CF}_{t-1}, Y_d, \mathbf{P}_d, \mathbf{CF}_d \}$$

Prédiction : generation du ROI.

On ajoute une estimation de l'erreur due aux evolution du cible :  $e$

$$\begin{aligned} \text{left} &= x - 2 \sigma_x + e \\ \text{right} &= x + 2 \sigma_x + e \\ \text{top} &= y - 2 \sigma_y + e \\ \text{bottom} &= y + 2 \sigma_y + e \end{aligned}$$



## Estimation Baysienne Robuste

L'estimation robuste utilise une Gaussien comme "fenêtre" afin de favoriser l'influence de pixels proche de la prédiction. L'idée est de multiplier les pixels de détection par une Gaussien.

Parce que le produit de deux Gaussien est aussi une Gaussien, les moments sont artificiellement réduits par ce produit. Pour éviter cela on accroît la taille de la fenêtre par 2.

$$V = 2 P_t^*$$

On utilise  $V$  pour calculer le ROI ( $u, l, b, r$ ).

Ensuite on calcule l'algorithme de détection dans la fenêtre.

(différence de fond, ratio d'histogramme de couleur, différence d'images, etc.....)

On évalue une porte de validation :

$$G(x, y; X_t, V) = e^{-\frac{1}{2} \begin{bmatrix} x & y \\ \hat{\sigma}_{xx} & \hat{\sigma}_{xy} \\ \hat{\sigma}_{xy} & \hat{\sigma}_{yy} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}}$$

$$CF_d = \frac{1}{N} \sum_{y=\text{top}}^{\text{bottom}} \sum_{x=\text{left}}^{\text{right}} d(x, y) G(x, y; X_t, V)$$

La nouvelle position estimée est  $\hat{Y}_d$

$$\hat{Y}_d = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}$$

ou

$$\mu_x = \frac{1}{N} \sum_{y=\text{top}}^{\text{bottom}} \sum_{x=\text{left}}^{\text{right}} d(x, y) \cdot x \cdot G(x, y; X_t, V)$$

$$\mu_y = \frac{1}{N} \sum_{y=\text{top}}^{\text{bottom}} \sum_{x=\text{left}}^{\text{right}} d(x, y) \cdot y \cdot G(x, y; X_t, V)$$

La taille du cible est,  $P_d$ , est

$$P_d = \begin{bmatrix} \hat{\sigma}_{xx}^2 & \hat{\sigma}_{xy}^2 \\ \hat{\sigma}_{xy}^2 & \hat{\sigma}_{yy}^2 \end{bmatrix}$$

ou

$$xx^2 = \frac{1}{N} \int_{y=\text{top}}^{\text{bottom}} \int_{x=\text{left}}^{\text{right}} d(x, y) \cdot (x - \mu_x)^2 \cdot G(x, y; X_t, V)$$

$$yy^2 = \frac{1}{N} \int_{y=\text{top}}^{\text{bottom}} \int_{x=\text{left}}^{\text{right}} d(x, y) \cdot (y - \mu_y)^2 \cdot G(x, y; X_t, V)$$

$$xy^2 = \frac{1}{N} \int_{y=\text{top}}^{\text{bottom}} \int_{x=\text{left}}^{\text{right}} d(x, y) \cdot (x - \mu_x) \cdot (y - \mu_y) \cdot G(x, y; X_t, V)$$

Les autres etaps sont :

Initialisation: zones de détection

Prédiction-verification

élimination d'un cible

Split-Merge