

# Formation et Analyse d'Images

James L. Crowley

ENSIMAG 3

Premier Bimestre 2007/2008

Séance 10

14 décembre 2007

## Suivi et estimation de mouvements : le Filtre de Kalman

### Plan de la Séance :

Le Filtre de Kalman.....	2
Prédiction de l'évolution temporelle de l'état.....	3
Modèle du capteur.....	6
Aproximations des modèle non-linéaires.....	6
Validation.....	7
Mise à Jour.....	8
L'Innovation.....	9
Suivi des Blobs avec un filtre de Kalman.....	10
Architecture.....	10
Caractérisation par moments.....	10
Composantes principales.....	11
Predicting target location and size.....	12
Suivi de Segments de Contraste .....	14
Représentation de Segments de Contraste.....	14
Test de proximités de pixels.....	15
Représentation MOL : "Midpoint-Orientation-Length" .....	16
La précision d'un segment.....	17
Proximité d'une droite et un point.....	18
Proximité de deux segments.....	19
Jonctions.....	20

## Le Filtre de Kalman

Le filtre de Kalman est un estimateur récursif optimal.

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  $Y$

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

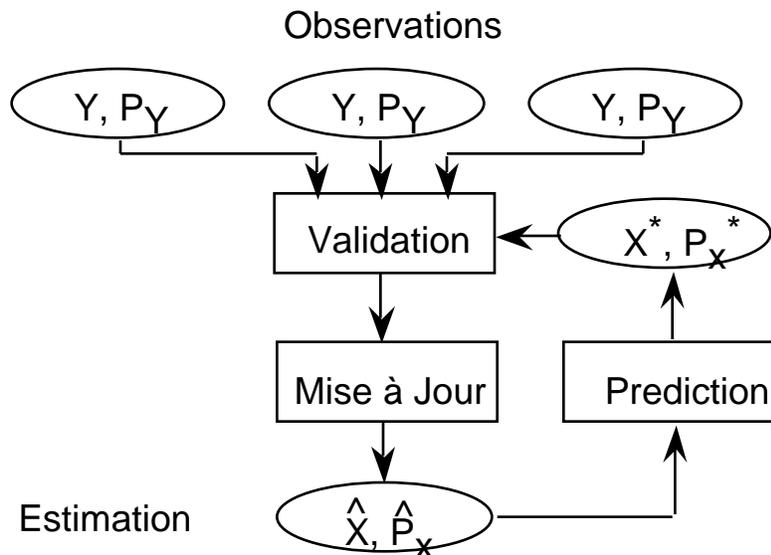
Pour chaque instant,  $t$ , le filtre donne une estimation  $\hat{X}_t$  ainsi que sa précision  $P_t$ , à partir de l'observation et l'estimation précédente,  $\hat{X}_{t-1}$ ,  $P_{t-1}$ , et l'observation  $Y$ , et sa précision,  $P_y$ . Les précisions sont des covariances.

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

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 :

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

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.

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

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

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

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

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

où

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

et

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

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

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.

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

avec

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

Prédiction :

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

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

### Modèle du capteur

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

$$\text{III: } Y := H_X^Y X_t$$

$$\text{IV: } P_y = E\{W_t W_t^T\}$$

La matrice de  $H_X^Y$  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  $P_y = E\{W_t W_t^T\}$

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

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

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

$$w X^i = M_s^i X^s$$

### 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.

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

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 :

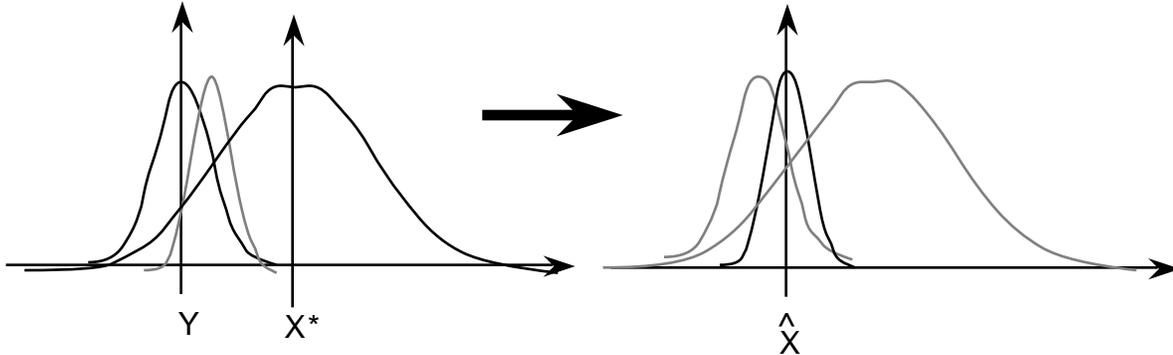
$$\text{Arg}_k \text{Min} \{ d_k^2 \} \text{ tel que } d_k \leq \text{seuil.}$$

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 :

$$\text{VI: } \mathbf{K} := \mathbf{P}_t^* \mathbf{H}_X^Y \text{T} [\mathbf{H}_X^Y \mathbf{P}_t^* \mathbf{H}_X^Y \text{T} + \mathbf{P}_y]^{-1}$$

$$\text{VII: } \hat{\mathbf{X}}_t := \mathbf{X}_t^* + \mathbf{K} [Y - \mathbf{H}_X^Y \mathbf{X}_t^*]$$

$$\text{VIII: } \hat{\mathbf{P}}_t := \mathbf{P}_t^* - \mathbf{K} \mathbf{H}_X^Y \mathbf{P}_t^*$$

### L'Innovation

Le différence  $(Y - \mathbf{H}_X^Y \mathbf{X}_t^*)$  est l'innovation de l'observation.

Le gain  $\mathbf{K}$  est une matrice qui donne le poids pour une mise à jour de  $x_i$  à partir de chaque composant de l'innovation.

$$\hat{\mathbf{X}}_t := \mathbf{P}_t^* - \mathbf{K} \mathbf{H}_X^Y \mathbf{P}_t^*$$

$$\begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{matrix} = \begin{matrix} k_{x1y1} & k_{x1y2} \\ k_{x2y1} & k_{x2y2} \\ k_{x3y1} & k_{x3y2} \\ k_{x4y1} & k_{x4y2} \end{matrix} \begin{matrix} y_1 - \mathbf{H}_X^1 \mathbf{X}_t^* \\ y_2 - \mathbf{H}_X^2 \mathbf{X}_t^* \end{matrix} = \begin{matrix} \frac{x_1}{y_1} & \frac{x_1}{y_2} \\ \frac{x_2}{y_1} & \frac{x_2}{y_2} \\ \frac{x_3}{y_1} & \frac{x_3}{y_2} \\ \frac{x_4}{y_1} & \frac{x_4}{y_2} \end{matrix} \begin{matrix} y_1 - \mathbf{H}_X^1 \mathbf{X}_t^* \\ y_2 - \mathbf{H}_X^2 \mathbf{X}_t^* \end{matrix}$$

$$\mathbf{K} := \mathbf{P}_t^* \mathbf{H}_X^Y \text{T} [\mathbf{H}_X^Y \mathbf{P}_t^* \mathbf{H}_X^Y \text{T} + \mathbf{P}_y]^{-1}$$

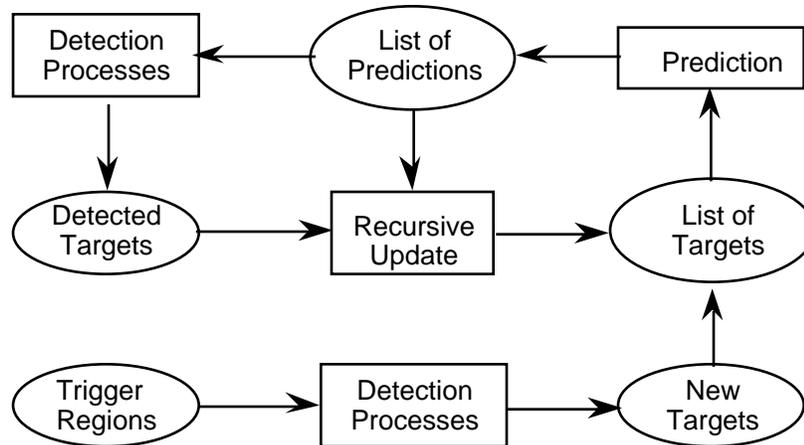
$$\begin{matrix} k_{x1y1} & k_{x1y2} \\ k_{x2y1} & k_{x2y2} \\ k_{x3y1} & k_{x3y2} \\ k_{x4y1} & k_{x4y2} \end{matrix} = \begin{matrix} 11^2 & 12^2 & 13^2 & 14^2 \\ 21^2 & 22^2 & 23^2 & 24^2 \\ 31^2 & 32^2 & 33^2 & 34^2 \\ 41^2 & 42^2 & 43^2 & 44^2 \end{matrix} \begin{matrix} h_{11} & h_{21} \\ h_{12} & h_{22} \\ h_{13} & h_{23} \\ h_{14} & h_{24} \end{matrix} \begin{matrix} \bullet & \bullet & -1 \\ \bullet & \bullet & \end{matrix}$$

ou

$$\begin{matrix} \bullet & \bullet \\ \bullet & \bullet \end{matrix} = \left( \begin{matrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \end{matrix} \begin{matrix} 11^2 & 12^2 & 13^2 & 14^2 \\ 21^2 & 22^2 & 23^2 & 24^2 \\ 31^2 & 32^2 & 33^2 & 34^2 \\ 41^2 & 42^2 & 43^2 & 44^2 \end{matrix} \begin{matrix} h_{11} & h_{21} \\ h_{12} & h_{22} \\ h_{13} & h_{23} \\ h_{14} & h_{24} \end{matrix} + \begin{matrix} 11^2 & 12^2 \\ 21^2 & 22^2 \end{matrix} \right)$$

## Suivi des Blobs avec un filtre de Kalman

### Architecture



### Caractérisation par moments

On peut définir les caractéristiques "invariantes" de l'orientation avec des "moments" de la forme.

Les moments sont invariants aux transformations affines.

Pour une image  $w(i, j)$  de taille  $N \times M$

Somme des Pixels : 
$$S = \sum_{i=0}^M \sum_{j=0}^N w(i, j)$$

Premiers moments :

$$\mu_i = \frac{1}{S} \sum_{i=0}^M \sum_{j=0}^N w(i, j) \cdot i \qquad \mu_j = \frac{1}{S} \sum_{i=0}^M \sum_{j=0}^N w(i, j) \cdot j$$

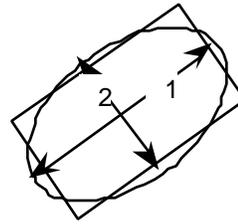
Le premier moment est le centre de gravité de la forme :

Deuxième moment :

$$i^2 = \frac{1}{S} \sum_{i=0}^M \sum_{j=0}^N (w(i, j)) \cdot (i - \mu_i)^2 \qquad j^2 = \frac{1}{S} \sum_{i=0}^M \sum_{j=0}^N w(i, j) \cdot (j - \mu_j)^2$$

$$ji^2 = \frac{1}{S} \sum_{i=0}^M \sum_{j=0}^N w(i, j) \cdot (i - \mu_i)(j - \mu_j)$$

Ceci permet de définir les "axes", majeur,  $\lambda_1$  et mineur,  $\lambda_2$ , de la forme



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

Les deuxièmes moments sont "invariants" à l'orientation

## Composantes principales

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  $\Phi C_P \Phi^T = \Lambda$  telles que  $\Lambda$  soit diagonale.

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

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

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

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

La longueur des axes majeur 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.

The target state vector is

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

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

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

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  $CF_d$ .

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

## Predicting target location and size

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

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

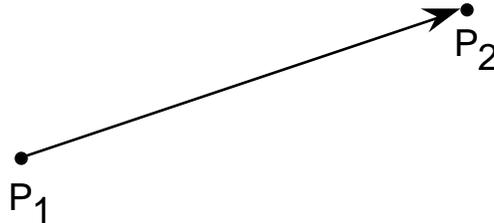
We assume image coordinates with positive y coordinates pointing down.

## Suivi de Segments de Contraste

### Représentation de Segments de Contraste

Technique classique de détection : découpage récursif d'un segment de droite :

Segment :



Un segment est défini par deux points extrêmes :  $P_1$  et  $P_2$ .

```
en C :          point struct { int x, y;      /* Coordonnées */ /
                    covariance_2 *C; /* precisions */
                };
```

Un point peut avoir une incertitude en position (une précision).  
Ceci est représenté par une covariance :  $C_p$

$$C_p \hat{=} \begin{array}{cc} x^2 & xy \\ yx & y^2 \end{array}$$

où

$$xy \hat{=} E\{ (x - \hat{x}) (y - \hat{y}) \}$$

et

$$x^2 \hat{=} E\{ (x - \hat{x})^2 \}, \quad y^2 \hat{=} E\{ (y - \hat{y})^2 \}$$

Les incertitudes sont estimées à base d'observations (un calibrage).  
Elles sont souvent des paramètres du système.

## Test de proximités de pixels

Deux points,  $P_1$  et  $P_2$  sont en "correspondance" si leur distance est plus petite que la somme de leurs précisions.

On utilise une distance de Mahalanobis afin de tester la correspondance :

$$d^2 = -\frac{1}{2} (P_1 - P_2)^T (C_{p1} + C_{p2})^{-1} (P_1 - P_2)$$

$$\text{Similar}(P_1, P_2) = \begin{cases} \text{true} & -\frac{1}{2} (P_1 - P_2)^T (C_{p1} + C_{p2})^{-1} (P_1 - P_2) \leq s^2 \\ \text{false} & \text{Sinon.} \end{cases}$$

ou "s" est l'écart type pour lequel on accepte une correspondance. (typiquement  $s = 2$ ).

### Equation homogène d'une droite

Les segments appartiennent à une droite de support :

$$\begin{aligned} \text{équation homogène :} & \quad a x + b y + c = 0 \\ \text{ou bien :} & \quad \sin(\alpha) x - \cos(\alpha) y + c = 0 \end{aligned}$$

Soit  $P_1$  avec coordonnées  $(x_1, y_1)$  et  $P_2$  a pour coordonnées  $(x_2, y_2)$ .

On peut calculer :

$$d_x = (x_2 - x_1) \quad d_y = (y_2 - y_1) \quad L = \|(d_x, d_y)\| = \sqrt{d_x^2 + d_y^2}$$

$$\cos(\alpha) = \frac{d_x}{L} \quad \sin(\alpha) = \frac{d_y}{L} \quad c = -(a x_1 + b y_1)$$

la distance de tout point  $(x, y)$  à la droite est :  $d = |ax + by + c|$

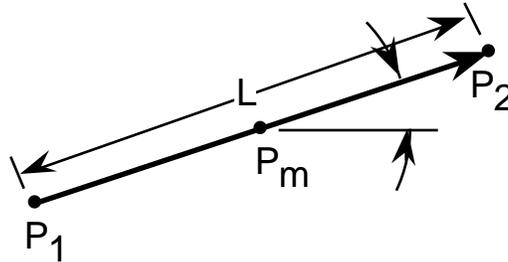
La droite peut être représentée par un vecteur  $D = (a, b, c)^T$ .

pour un point  $P = (x, y, 1)^T$  :

$$\langle D, P \rangle = d \quad \text{et } d = 0 \text{ pour tout } P \in D$$

## Représentation MOL : "Midpoint-Orientation-Length".

Un segment de droite peut être décrit par une représentation "paramétrique" : composer un point central, direction, et longueur (Midpoint, Orientation, Length)



Segment :  $P_m$ : Point du Centre:  $(x, y)$   
 : Orientation  
 L: Longueur

ou bien  $S \hat{=} (x, y, \theta, L)$

on peut ajouter les paramètres de la droite support :  $(a, b, c)$ .

Il est également utile d'avoir  $d$ , distance perpendiculaire à l'origine.

$$d = x \cos(\theta) + y \sin(\theta)$$

On note que  $(c, d)$  est une autre représentation de  $(x, y)$   
 et que  $(a, b)$  est une autre représentation de  $(\theta)$ .

en C :

```
segment struct {point *p1, *p2; /* points extremes */
  int x, y; /* point centrale */
  double theta; /* Angle */
  double l; /* length */
  double a, b, c; /* droit du support */
  double d; /* distance de l'origine */
  covariance_4D *C; /* Covariance */
};
```

## La précision d'un segment.

Afin d'apparier les segments, il faut des précisions.

Une précision est représentée par une variance (ou un écart type) :

$C_{xy}$  : Covariance en position du point central  
 $\sigma^2$  : Variance en orientation  
 $l^2$  : Variance en longueur

$$C_{xy} \cong \begin{pmatrix} \sigma^2 & xy \\ yx & l^2 \end{pmatrix}$$

Les incertitudes sont estimées à base des observations (un calibrage).  
Elles sont souvent des paramètres du système.

On peut calculer une estimation de la précision pour chaque paramètre par les moyens statistiques. On observe des segments et on mesure les statistiques.

Pour les segments de contraste : on constate que le bruit d'un segment parvient surtout de la "cassure".

Si on "diagonalise" la matrice  $C_{xy}$ , on trouve un vecteur propre aligné avec le segment, et un vecteur "perpendiculaire" du segment.

La covariance dans la direction du segment est souvent très large  
La covariance perpendiculaire est petite.

Donc, on peut simplifier, en remplaçant  $C_{xy}$  par  $c^2$  et  $d^2$

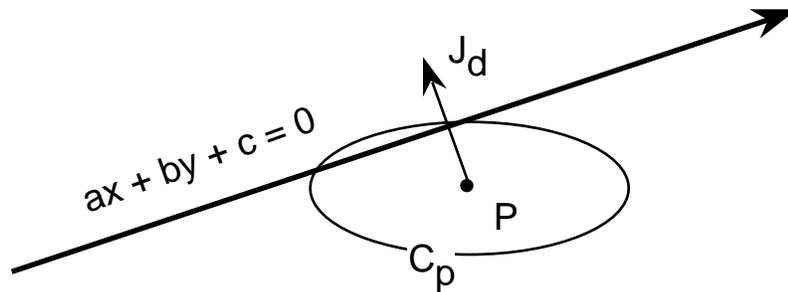
Ceci donne la représentation d'un segment :  $S \cong \{c, d, \theta, L\}$   
avec  $C_s$  :

$c$  : Incertitude en position perpendiculaire  
 $d$  : Incertitude en position tangentielle  
 $\theta$  : Incertitude en orientation  
 $l$  : Incertitude en longueur

$$\hat{C}_s \cong \begin{pmatrix} c^2 & 0 & 0 & 0 \\ 0 & d^2 & 0 & 0 \\ 0 & 0 & h^2 & 0 \\ 0 & 0 & 0 & l^2 \end{pmatrix}$$

### Proximité d'une droite et un point.

pour la vraisemblance entre une droite et un point il faut déterminer la composante de la précision du point perpendiculaire à la droite.



La composante de l'incertitude perpendiculaire à la droite est calculée par la Jacobienne de la distance  $d = F(P) = a x + by + c$

$$J_d = \frac{F(P)}{P} = \frac{(ax+by+c)}{(x, y)} = \frac{(\sin(\theta)x + \cos(\theta)y + c)}{(x, y)}$$

$$= \left( \frac{(\sin(\theta)x + \cos(\theta)y + c)}{x}, \frac{(\sin(\theta)x + \cos(\theta)y + c)}{y} \right)$$

$$J_d = (a, b) = (\sin(\theta), \cos(\theta)) = N_d \text{ (La normale de la droite)}$$

Noter qu'il s'agit d'un vecteur ligne !

$$pN^2 = J_d C_{xy} J_d^T = (a, b) \begin{pmatrix} x^2 & xy & a \\ yx & y^2 & b \end{pmatrix}$$

$$= (a, b) \begin{pmatrix} b x^2 + a xy \\ b yx + a y^2 \end{pmatrix} =$$

$$= a(a x^2 + b xy) + b(a yx + b y^2)$$

$$= a^2 x^2 + 2ba yx + b^2 y^2$$

$$= \sin(\theta)^2 x^2 + 2 \sin(\theta) \cos(\theta) xy + \cos(\theta)^2 y^2$$

$$2 = pN^2 + c^2$$

$$\text{Similar}( \text{droite } D, \text{ point } P, pN^2 + c^2 ) = \begin{cases} \text{true} & \frac{1}{2} \frac{\langle DP \rangle^2}{pN^2 + c^2} \leq s^2 \\ \text{false} & \text{sinon.} \end{cases}$$

où  $s$  est la nombre d'ecart type qu'on accept. On peut faire  $s = 2$ .

**Proximité de deux segments.**

Méthode 1 : calculer la distance de Mahalanobis entre les paramètres

Pour deux segments :  $S_1 = (x, y, \dots, L)$ , avec  $C_{s1}$

$S_2 = (x, y, \dots, L)$ , avec  $C_{s2}$

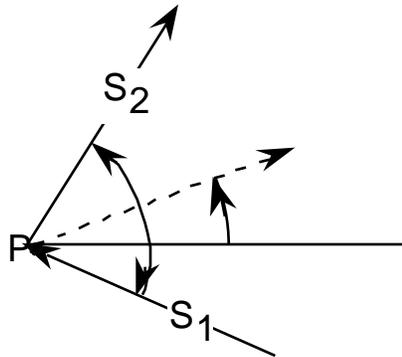
$$\text{où } \hat{C}_S \equiv \begin{pmatrix} x^2 & xy & 0 & 0 \\ yx & y^2 & 0 & 0 \\ 0 & 0 & h^2 & 0 \\ 0 & 0 & 0 & \dots^2 \end{pmatrix}$$

$$\text{Similar}(S_1, S_2, C_{s1}+C_{s2}) = \begin{cases} \text{true} & \frac{1}{2} (S_1-S_2)^T (C_{s1}+C_{s2})^{-1} (S_1-S_2) \leq s^2 \\ \text{false} & \text{Sinon.} \end{cases}$$

## Jonctions

Deux segments forment une jonction s'ils partagent un point.

if similar( $S_1.P_2, S_2.P_1, S_1.C+S_1.C$ ) then  $J = \text{MakeJunction}(S_1, S_2)$ ;



Les paramètres d'une jonction sont :

P : Point Commun  
 : angle interne  $= S_2. - S_1.$   
 : angle externe  $= \frac{1}{2} (S_2. + S_1. )$

```
segment struct {jonction *s1, *s2; /* segments */
    int x, y; /* point centrale */
    double phi; /* angle interne*/
    double theta; /* angle externe*/
    covariance_4 *C; /* Covariance */
};
```

Les jonctions se ressemblent si leurs paramètres sont semblables.

Similar(  $J_1, J_2, J_1.C+J_2.C$  ) = true  $\frac{1}{2} (J_1-J_2)^T C^{-1} (J_1-J_2) \leq s^2$   
 false Sinon.