



République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université des Sciences et de la Technologie Houari Boumediene  
Faculté d'Electronique et d'Informatique  
Département d'Informatique



# La vision par ordinateur

## *Chapitre 5 : Le mouvement en vision par ordinateur*

Master 2 : Systèmes Informatique Intelligents  
Lyes\_sii@yahoo.fr lyes\_abada@yahoo.fr

# Séquence d'images

---

Traiter le mouvement dans des images implique traiter **une séquence d'images** prises à des intervalles de temps (**de 24 à 60** images par seconde (FPS : frame per second)).

Le mouvement est très utilisé dans la vision 3D:

- La **dimension temporelle** ajoute de l'information pour analyser la scène 3D
- Permet **d'extraire des propriétés** de la scène sans avoir à faire une **reconstruction complète** et sans calibration

**Remarque:** La vitesse du mouvement est inversement proportionnelle à la distance (les objets plus lointains semblent bouger lentement)

# Les causes et les types du mouvement

---

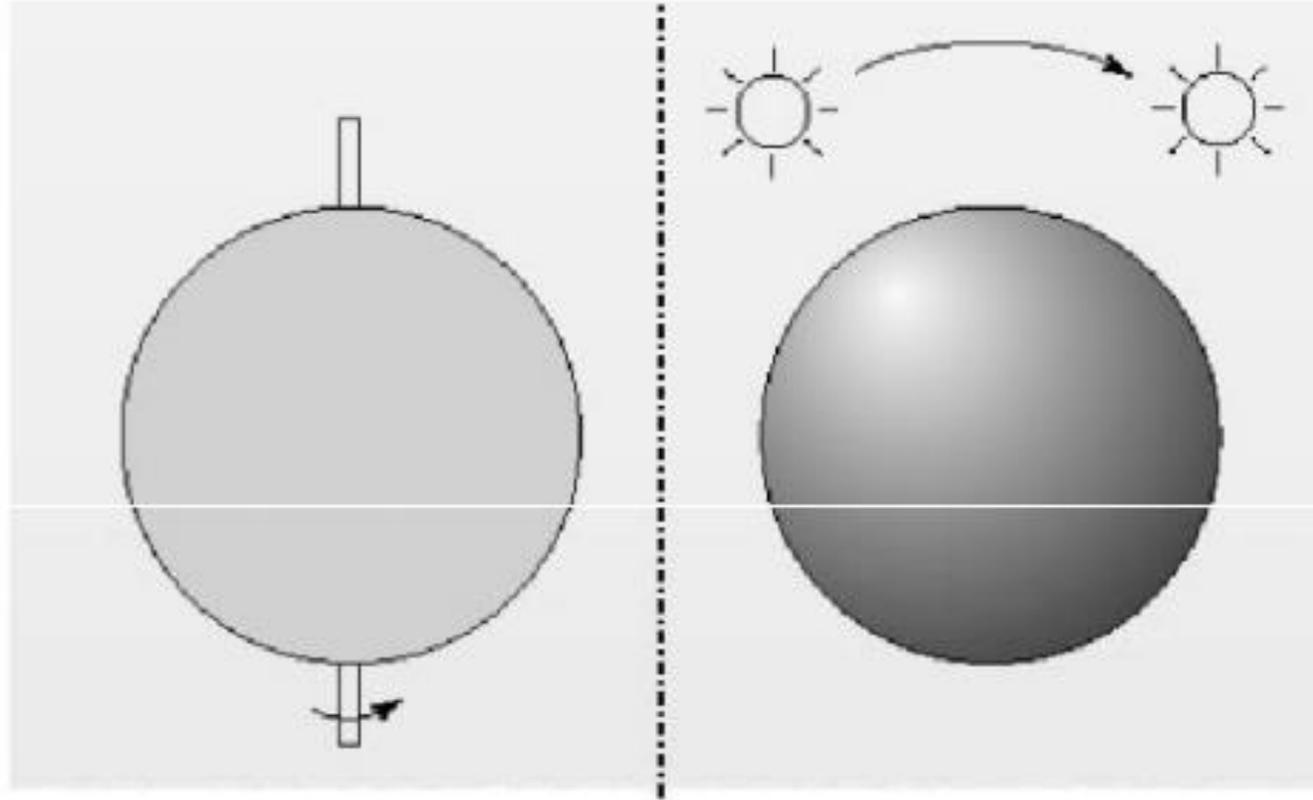
## a) Les causes du mouvement

Les causes sont:

- Déplacement d'un objet dans la scène
- Déplacement de la caméra

Dans l'image, on mesure le mouvement apparent, qui peut être différent du mouvement réel dans la scène.

# Les causes et les types du mouvement



***La sphère tourne sur elle-même***

- **Mouvement réel** non-nul  
(rotation)
- **Mouvement apparent** nul

***La source lumineuse se déplace***

- **Mouvement réel** nul
- **Mouvement apparent** non-nul

## b) Les types du mouvement

- **Scène rigide:** où la caméra est la seule qui se déplace. Par exemple une caméra sur un robot mobile.
- **Objets rigides avec des mouvements différents,** comme par exemple le trafic routier.
- **Un ou plusieurs objets non rigides:** par exemple le mouvement humain, les déformations du visage, le mouvement des nuages, le mouvement des cellules vivantes sous microscope

**Remarque:** Une fois les correspondances faites sur plusieurs points d'un objet, on peut déterminer son mouvement 3D

# Détection du mouvement

## a) Détection du mouvement

La détection du mouvement peut se faire avec la différence d'images (sans avoir le vecteur du mouvement).



- Avec un seuillage de la différence, on obtient plus clairement les zones de mouvement (par exemple  $> 50$ )
- La différence d'images n'est pas une méthode complète mais elle est très rapide pour la détection simple du mouvement.

## b) Estimation du mouvement

Estimer le mouvement, consiste à reconstruire **le champ de vitesse  $v(x,y)$**  d'une séquence d'images  $I(x,y,t)$ .

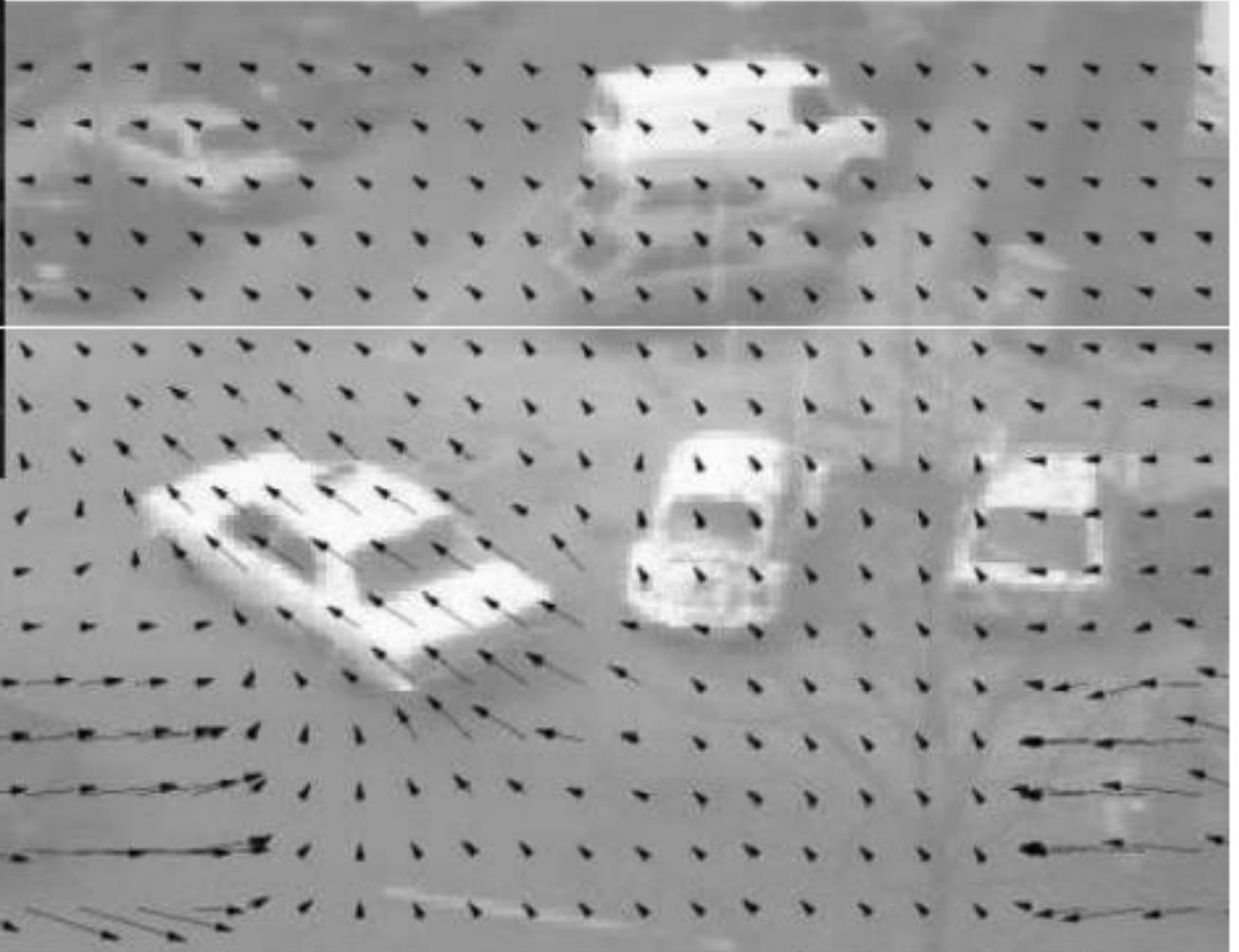
Le champs de vitesse est appelé le flot ou le flux optique (optical flow).

Il est donc nécessaire de calculer les correspondances:

- Entre deux images prises au même moment dans le cas de la **stéréo** (une paire d'images)
- Entre deux images prises à des moments différents (**une séquence d'images**)

# Estimation du mouvement

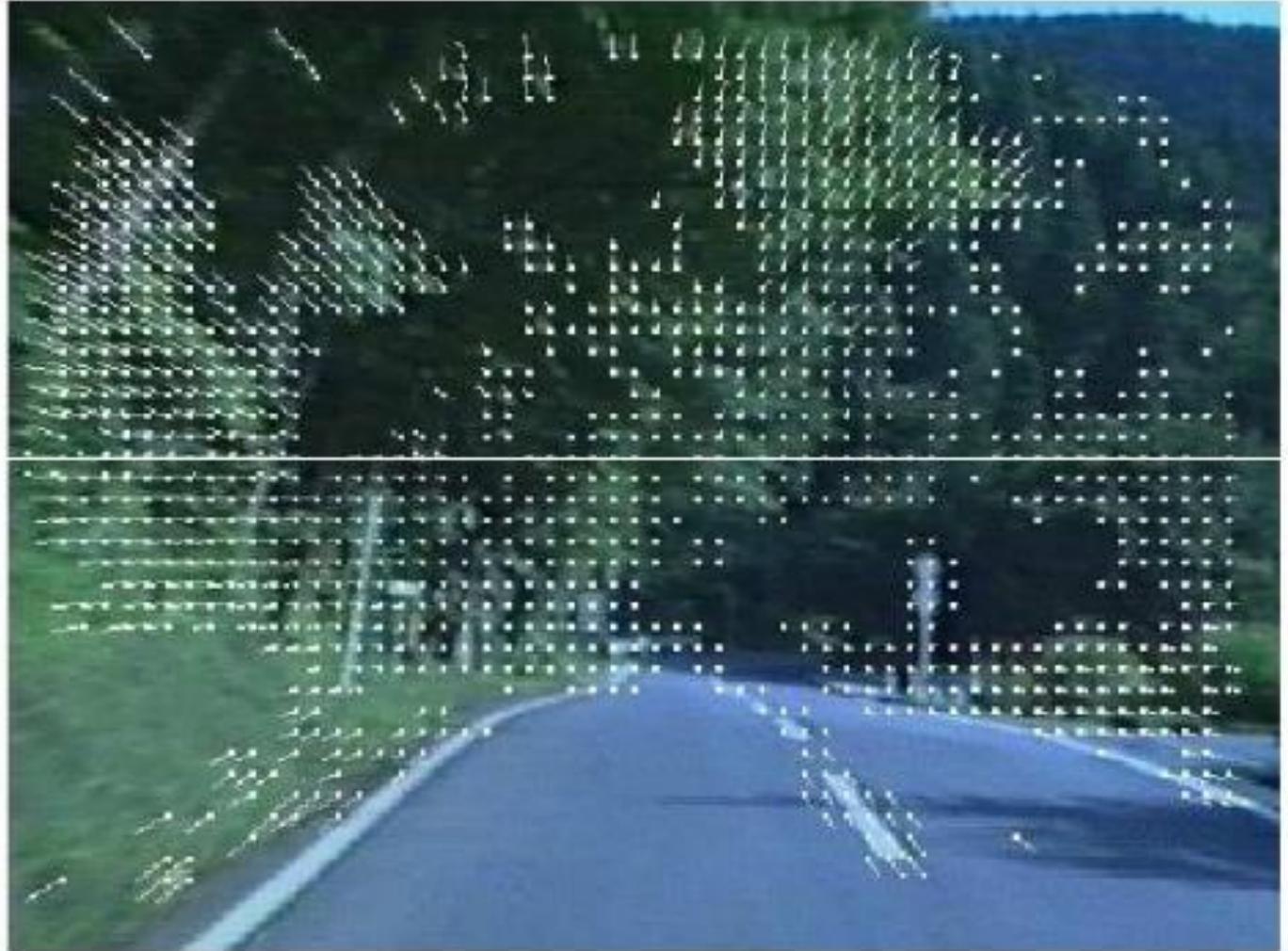
b) Estimation du mouvement (exemple 1)



Le déplacement :  $I_1(x,y) = I_2(x + d_x, y + d_y)$

La vitesse :  $v(x,y) = (d_x, d_y)^T$

b) Estimation du mouvement  
(exemple 2)



*Caméra fixée devant une voiture en  
mouvement sur la route*

# Suivi du mouvement (tracking)

---

## c) Suivi du mouvement (tracking)

Une fois que le mouvement est détecté (différence d'images), puis estimé (flot optique), on peut faire le suivi (the tracking)

**But :** suivre un objet sur plusieurs images en temps réel (une image après l'autre).

### → Problèmes :

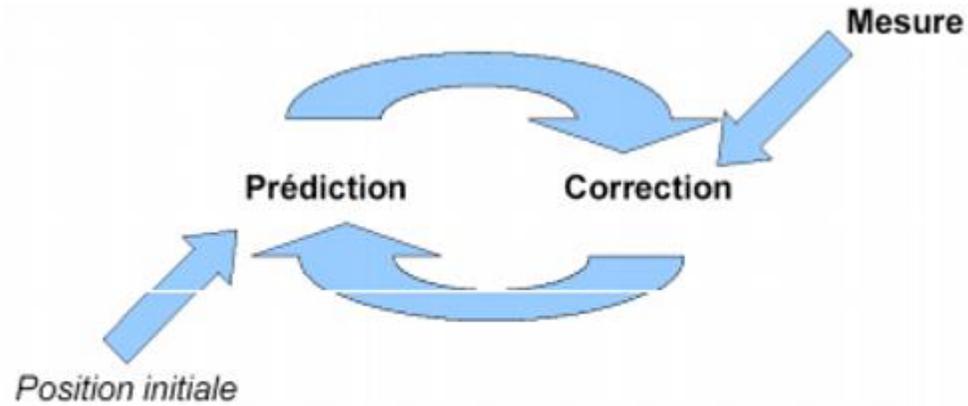
- Suivre le même objet sans le perdre
- Trouver sa position dans chaque nouvelle image
- Séparer les différents mouvements de l'image (Si plusieurs objets se déplacent en même temps, on veut suivre chaque objet séparément et sans les confondre)

**Solution:** plusieurs méthodes, la plus connue est le filtre de Kalman.

→ il s'agit de prédire, mesurer puis corriger la position de l'objet.

# Suivi du mouvement (tracking)

## c) Suivi du mouvement (tracking)



➔ On veut calculer la trajectoire de l'objet en diminuant les erreurs.

## c) Suivi du mouvement (tracking)

### Représenter un objet dans l'image:

On représente un objet à suivre par un point

Ce point est en fait un vecteur de caractéristiques

- La position  $x, y$
- La vitesse  $V_x, V_y$
- L'accélération  $A_x, A_y$
- Taille en pixels de l'objet dans l'image.

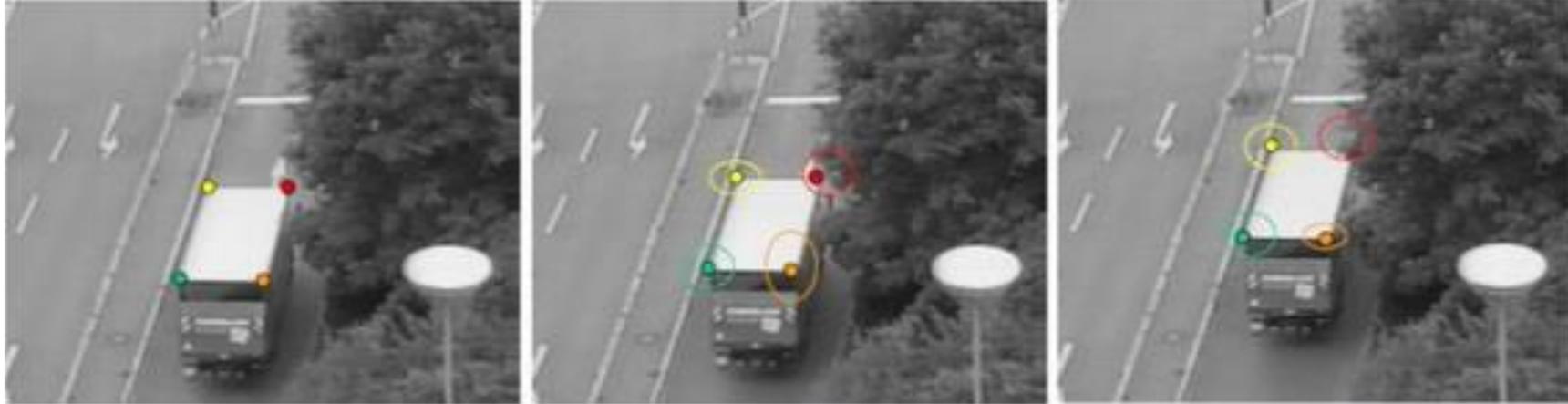
Exemple de vecteur avec position + vitesse :  $(x, y, V_x, V_y)^T$

## c) Suivi du mouvement (tracking)

On utilise plusieurs vecteurs différents :

- Une **prédiction** sur le point dans l'image  $(x, y, v_x, v_y)$ 
  - Estimation de la nouvelle position du point dans l'image
- Une **mesure** dans l'image (*position x et y seulement*)
  - Après segmentation (ou autre), on mesure le point
- Une **correction** sur le point  $(x, y, v_x, v_y)$ 
  - Estimation corrigée avec la mesure qui a été faite

## c) Suivi du mouvement (tracking)



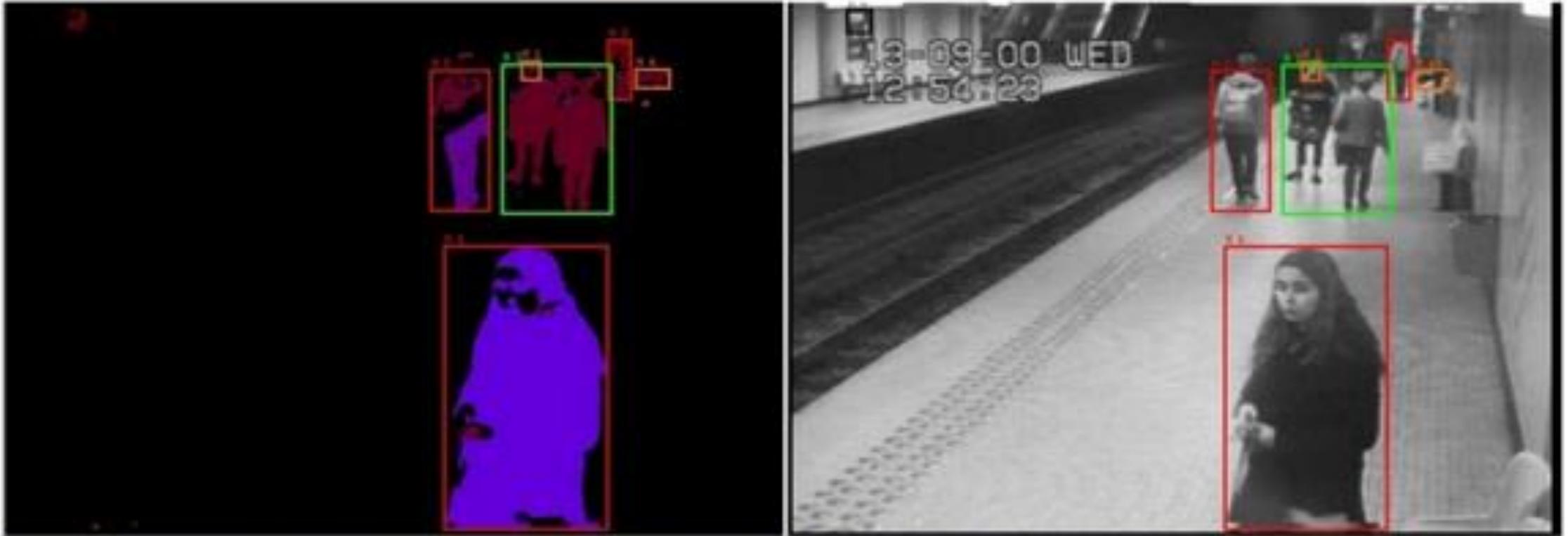
$t = 0$

$t = 20$

$t = 80$

- **On veut :**
  - Trouver des points de référence
  - **Prédire** où ils se trouveront dans l'image suivante à l'aide d'un modèle du mouvement
  - **Rechercher et mesurer** ceux-ci dans la zone de recherche prédite
  - **Corriger** l'estimation qu'on a faite
  - Répéter

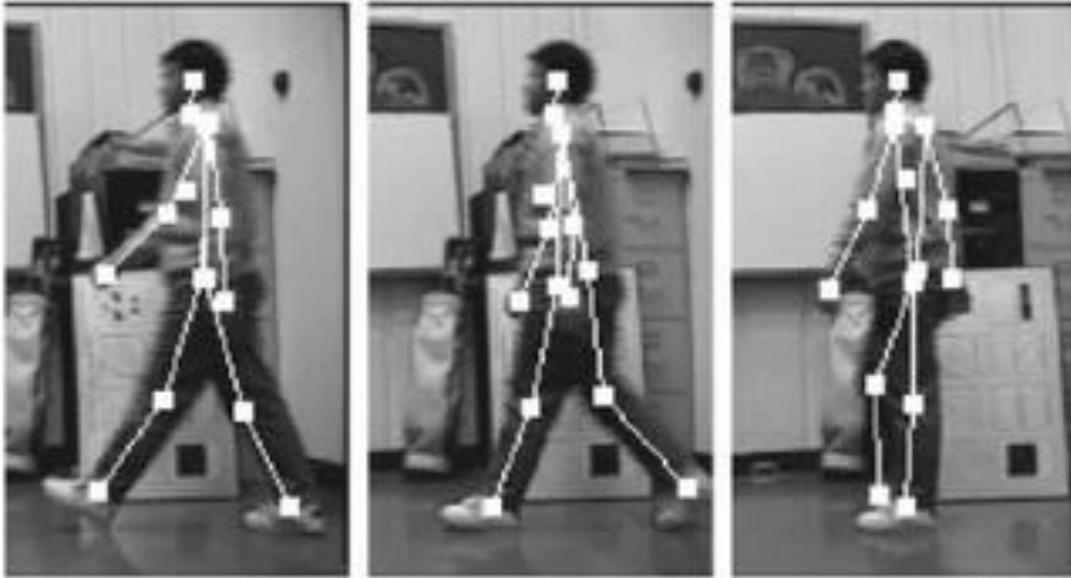
## c) Suivi du mouvement (tracking)



*Source : François Brémont, équipe-projet PULSAR, INRIA Sophia-Antipolis (France).*

Détecter les différents objets, chacun dans une boîte englobante.

# Suivi du mouvement (tracking)



Human activity recognition (Ben-Arie 2002)



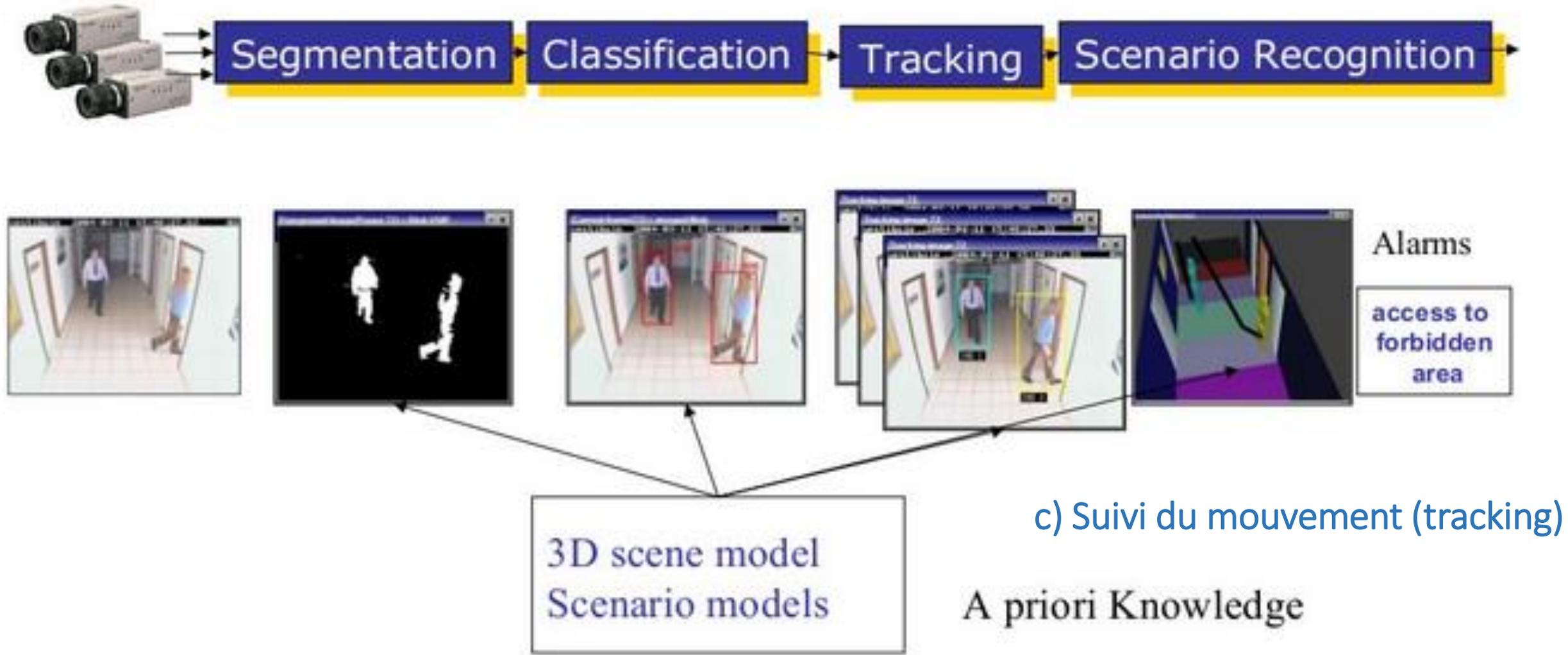
Visual surveillance (Thonnat 2002)



Tracking of a mobile arm (Rasmussen et al. 2001)

c) Suivi du mouvement (tracking)

# Suivi du mouvement (tracking)



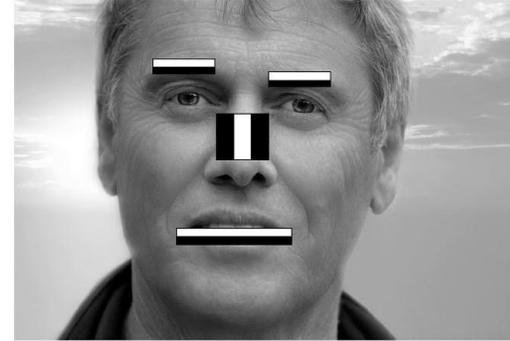
# Haar Cascade Classifier

## 1) Détection d'objet avec Haar Cascade Classifier : (Méthode de Viola et Jones)

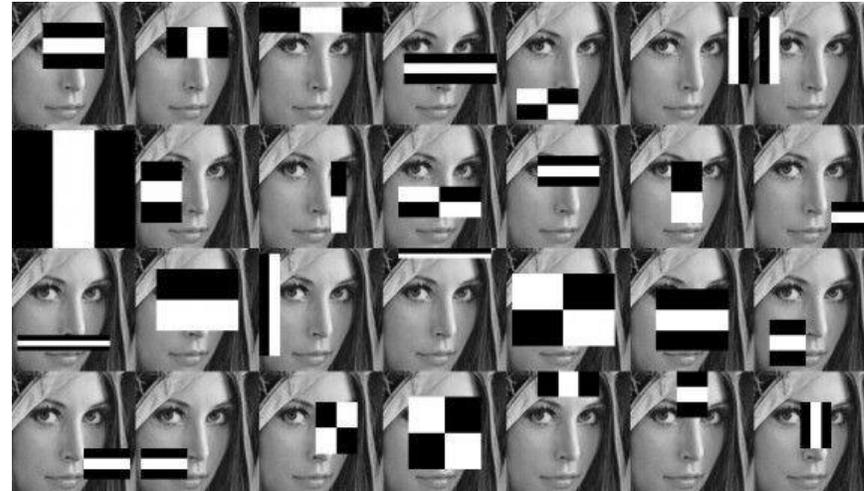
### a) Extraction des caractéristiques d'une image

classificateurs qui ont été utilisés dans le premier détecteur de visage en temps réel.

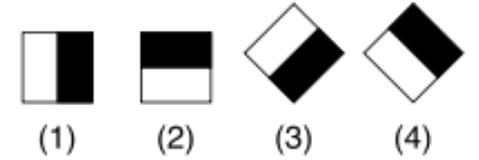
Un classificateur de Haar, ou un classificateur en cascade de Haar, est un algorithme de détection d'objets basé sur l'apprentissage automatique qui identifie des objets dans une image et une vidéo.



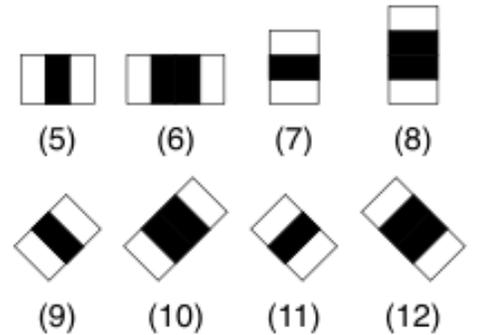
WE CAN REPRESENT THE MOST RELEVANT FEATURES WITH HAAR-FEATURES !!!



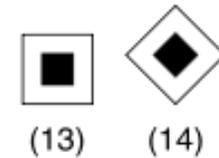
#### Caractéristiques de bord



#### Caractéristiques de ligne



#### Caractéristiques centre-pourtour



## b) image intégrale

L'image intégrale est une méthode pour calculer la somme d'un rectangle de pixels très rapidement, en temps constant  $O(1)$ , au lieu de le faire naïvement avec  $O(n^2)$ .

- La première étape consiste à obtenir l'image intégrale, où chaque pixel est la somme de ses pixels précédents (pixels du haut et de gauche).
- Ensuite, on calcule la somme d'un rectangle donné comme  $3+2+5+4=14$ , on peut utiliser l'image intégrale et n'utiliser que les coins de la manière suivante :

bleu-marron-vert+noir qui est égal à  $46-22-20+10=14$

1	2	2	4	1
3	4	1	5	2
2	3	3	2	4
4	1	5	4	6
6	3	2	1	3

input image

0	0	0	0	0	0
0	1	3	5	9	10
0	4	10	13	22	25
0	6	15	21	32	39
0	10	20	31	46	59
0	16	29	42	58	74

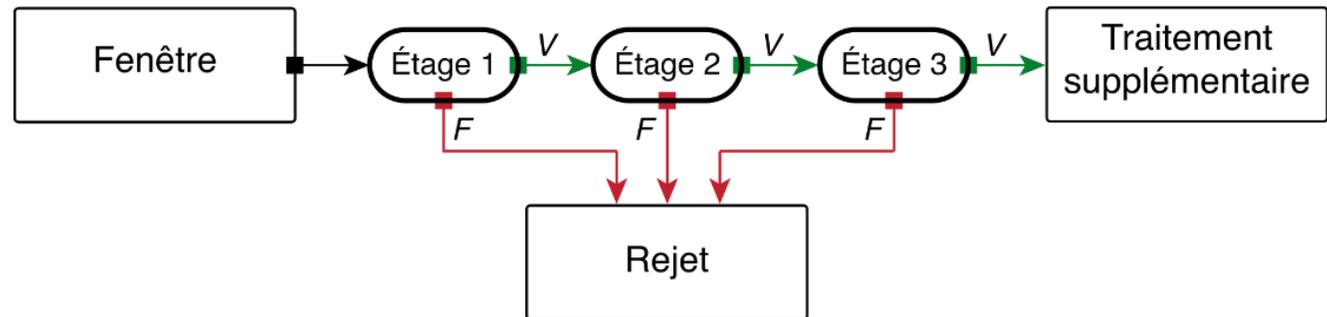
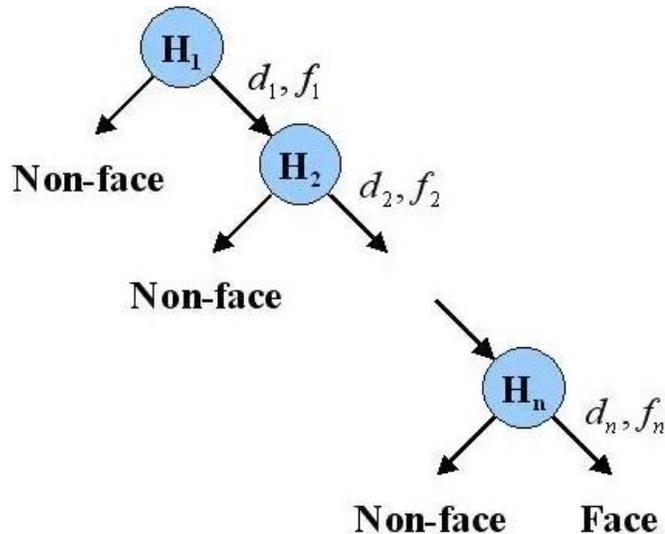
integral image

# Haar Cascade Classifier

## c) Adaboost

Nous pouvons obtenir des centaines, voire des milliers d'entités de Haar, il serait donc très inefficace de toutes les calculer pour chaque région de l'image.

La cascade indique si une entité Haar ne correspond pas, supprimez cette région et passez à la région suivante.



# Haar Cascade Classifier

a) Exemple :



## II) Le filtre de Kalman

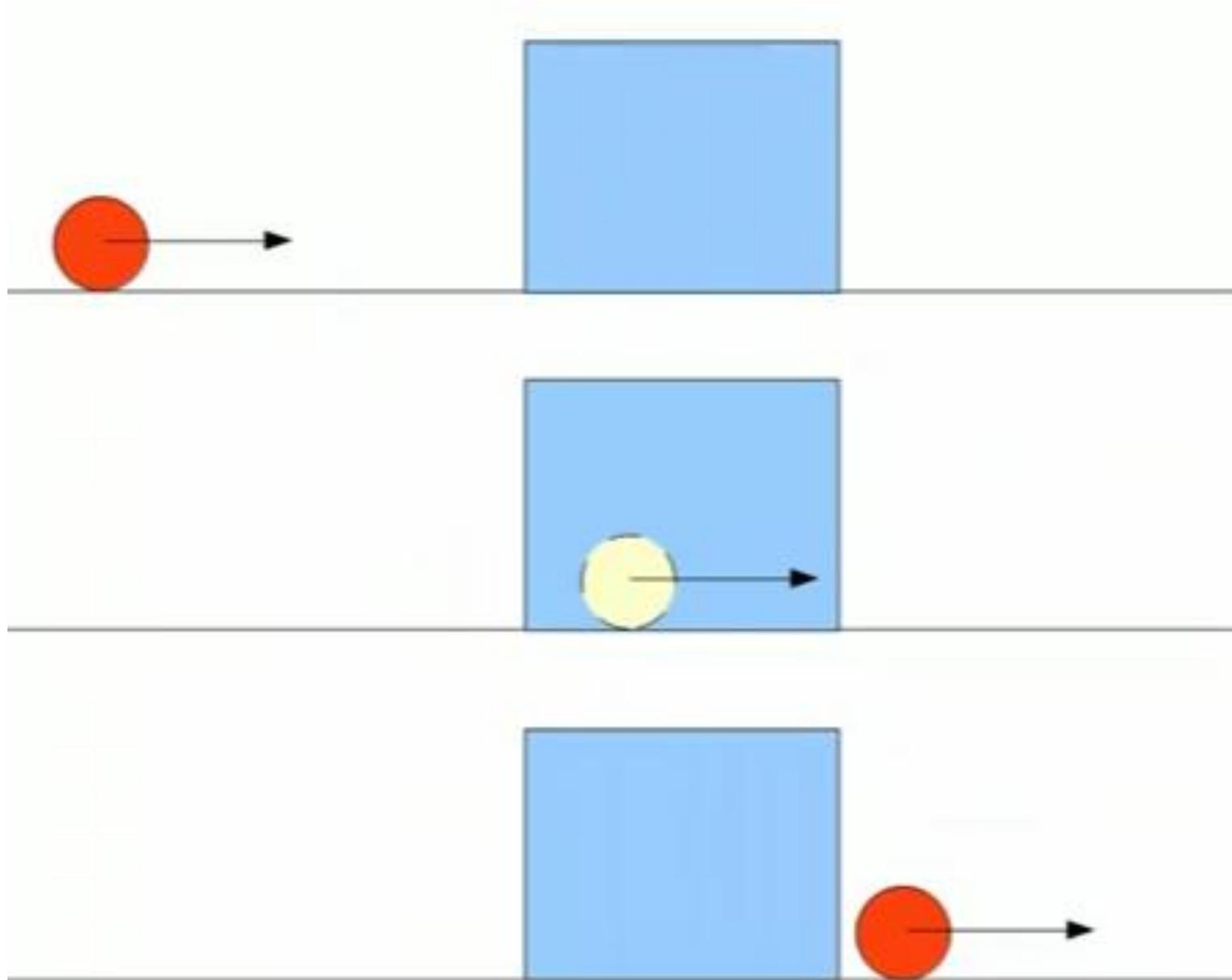
Le filtre de Kalman est une méthode visant à estimer des paramètres d'un système évoluant dans le temps à partir de mesures bruités.

On retrouve ce filtre dans bon nombre de domaines relatifs au traitement du signal, radar, traitement d'images etc

La force de ce filtre est sa capacité de prédiction des paramètres et de rectification des erreurs,

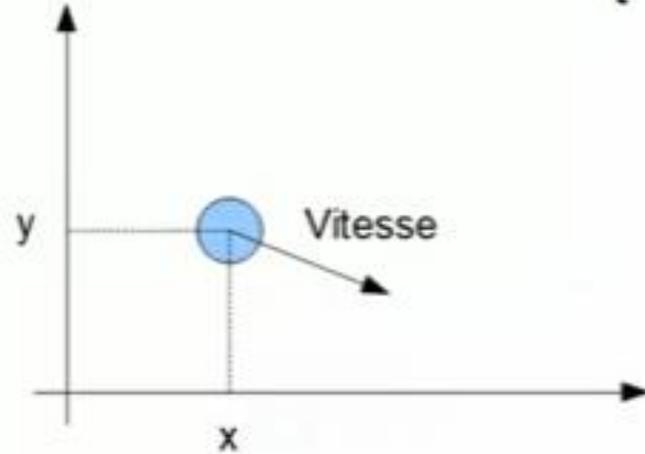
# Filtre de Kalman

Le filtre de Kalman



# Filtre de Kalman

## Le filtre de Kalman



$$E_t = A \cdot E_{t-1} + q_{t-1}$$

E: vecteur d'état

A: matrice de transition

q: bruit gaussien (matrice de covariance Q)

$$X_t = X_{t-1} + v_{xt-1} \cdot dt$$

$$Y_t = Y_{t-1} + v_{yt-1} \cdot dt$$

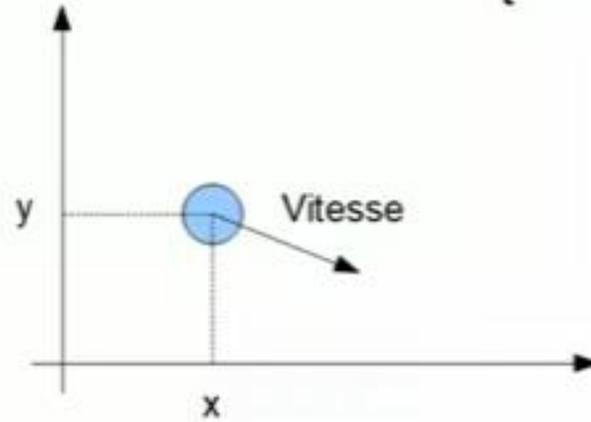
$$E_t = \begin{pmatrix} X_t \\ y_t \\ v_{xt} \\ v_{yt} \end{pmatrix}$$

$$A = \begin{vmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

# Filtre de Kalman

## Le filtre de Kalman

$$Z_t = H \cdot E_t + r_t$$



Z: vecteur contenant les mesures  
H: matrice d'observation  
r: bruit gaussien (matrice de covariance R)

$$E_t = \begin{pmatrix} x_t \\ y_t \\ v_{xt} \\ v_{yt} \end{pmatrix}$$

$$H = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{vmatrix}$$

## Le filtre de Kalman

### Prédiction / estimation

Prédiction de l'état

$$\hat{E}_{t+1} = A \cdot \hat{E}_t$$

Estimation de la covariance de l'erreur

$$P_{t+1}^- = A_t \cdot P_t \cdot A_t^T + Q_t$$

### Mises à jour

(si mesure)

Gain de Kalman

$$K_{t+1} = P_t \cdot H_t^T \cdot (H_t \cdot P_t \cdot H_t^T + R_t)^{-1}$$

Correction / innovation

$$\hat{E}_{t+1} = \hat{E}_t + K_t \cdot (Z_t - H_t \hat{E}_t)$$

$$P_{t+1} = (I - K_t \cdot H_t) \cdot P_t^-$$

Le **gain de Kalman**  $K$  spécifie la force avec laquelle

- L'état courant va tendre vers les mesures avec le temps
- La précision de l'estimation de l'état va être réduite avec le temps