Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

ENSTA 3e année - ROB313
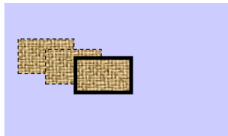# Vision pour la Robotique
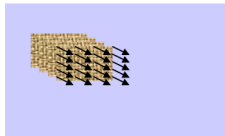# Object Tracking in videos

Antoine Manzanera
ENSTA-Paris

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Context and Objectives
Problematic
Fundamental Components

Object Tracking and Video Analysis

Three kinds of image processing primitives in Video analysis:
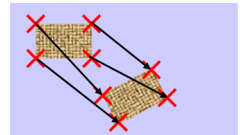

Detection
Separate mobile pixels from the static background

Estimation
Calculate the apparent velocity of each pixel

Tracking
Match spatial structures from frame to frame

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Context and Objectives
Problematic
Fundamental Components

Content and Goals of the lecture

- Present the characteristics, challenges and difficulties of object tracking in image sequences.
- Present the components of a tracking algorithm, and the main categories of object tracking methods.
- Explain the principles of *observation* and *prediction* that found the different approaches.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Context and Objectives
Problematic
Fundamental Components

Lecture outline

1. Introduction
   - Context and Objectives
   - Problematic
   - Fundamental Components

2. Observations and detection
   - Matching measures
   - Similarity between Distributions
   - Hough Transforms

3. Tracking a distribution
   - Mean-Shift Algorithm

4. Predictive Filtering
   - Kalman Filter
   - Particle Filtering

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Context and Objectives
Problematic
Fundamental Components
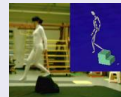
Application fields

Smart videosurveillance

- Geofencing / Abnormal activity
- Aggression / distress detection / crowd surveillance
- Dynamic (e.g. gait) biometry

Human-Machine Interfaces

- Visual command
- Avatar control
- Language sign

Bio-medical applications

- Gait analysis
- Elderly monitoring
- Sport analysis

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Context and Objectives
Problematic
Fundamental Components

Object Tracking

Context

- Mobile camera
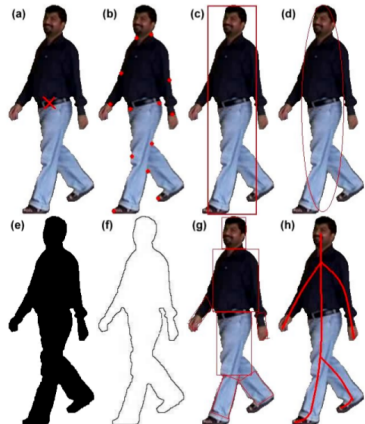- Mobile / deformable objects

Localisation vs Segmentation

The goal of object tracking is to localise an object initially defined (how?) by calculating in each video frame the spatial support (which one?) circumscribing the object.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Context and Objectives
Problematic
Fundamental Components

Spatial support of the object

The representation support of the object is determining in terms of: Flexibility / Precision / Complexity / Invariance...
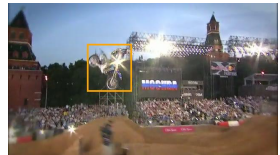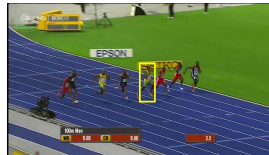On the right, some typical examples: (a) One point, (b) Several points, (c) Bounding box, (d) Ellipse, (e) Silhouette, (f) Contour, (g) Parts, (h) Skeleton.

From [Jalal12]

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Context and Objectives
Problematic
Fundamental Components

Tracking: difficulties and challenges

Examples taken from the dataset of the **[VOT14]** competition.

- Rotations, scale changes
- 3d and deformable Objects
- Similar Objects
- Complex and varying Background

- Occlusions
- Illumination changes
- Poor contrast
- Motion blur
- Sudden movements

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Context and Objectives
Problematic
Fundamental Components

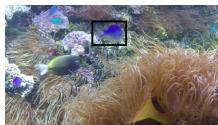# Tracking: difficulties and challenges

Examples taken from the dataset of the **[VOT14]** competition.

- Rotations, scale changes
- 3d and deformable Objects
- Similar Objects
- Complex and varying Background

- Occlusions
- Illumination changes
- Poor contrast
- Motion blur
- Sudden movements

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Context and Objectives
Problematic
Fundamental Components

Fundamental Components of Object Tracking

An object tracking algorithm is made of two fundamental elements:

- **Prediction**: provides, for each video frame, an initial location *hypothesis* of the tracked object. This hypothesis is usually based on a *dynamical model* (e.g.: kinematics) of the object and the background (camera).

- **Detection**: refines the localisation of the object using *observations* extracted from the image. This refinement is usually based on an *appearance model*.
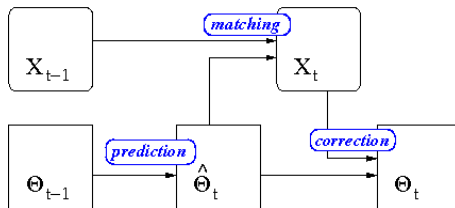
Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Context and Objectives
Problematic
Fundamental Components

General Formalism

Tracking can usually be expressed through the general framework:
$$\arg\max_{\theta \in S} P(\Theta_t = \theta / \Theta_{t-1} = \theta', X_t = x)$$
where S represent the *state* space; θ is a multi-dimensional state configuration, which can include (explicitely or implicitely) position, velocity, object category, etc, of the tracked objects; X_t represents the observation from the image space.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Context and Objectives
Problematic
Fundamental Components
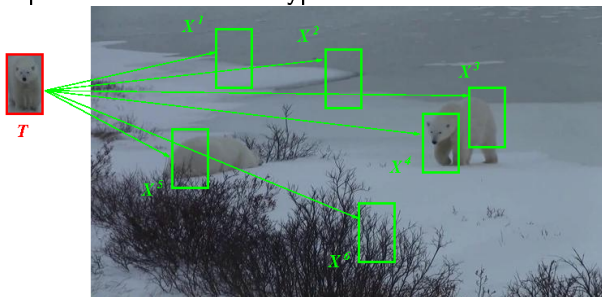
Prediction *vs* Detection

The actual balance between prediction and detection is very
depending on the type of algorithms, and can even be dramatically
broken in favour of either of the two:

- **Tracking by detection:** The tracking is entirely made by the
  detector: no location hypothesis required.
- **Track before detect:** The tracking is essentially provided by
  the prediction and can work even in case of extremely
  degraded visibility conditions (occlusion, noise, small size...).

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Global (di)similarity measures

Global (di)similarity measures are applied between two vectors T and X of the same dimension n, one of which being the model (template) of the object, (usually) represented by a rectangular patch, and the other a patch extracted from the current image, that corresponds to a location hypothesis.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Dissimilarity measures

Minkowski Distances

$$\mathcal{D}_k(T, X) = \sum_{i < n} |T_i - X_i|^k$$

Distances SAD $\mathcal{D}_1$ or SSD $\mathcal{D}_2$ are very popular for matching patches. These are dissimilarity measures whose value hardly depends on the size n and on the content of images.

The detection algorithm then consists in searching within a location hypotheses space $\{X^h\}_{h \in \mathcal{H}}$ to find the one that minimises the dissimilarity:
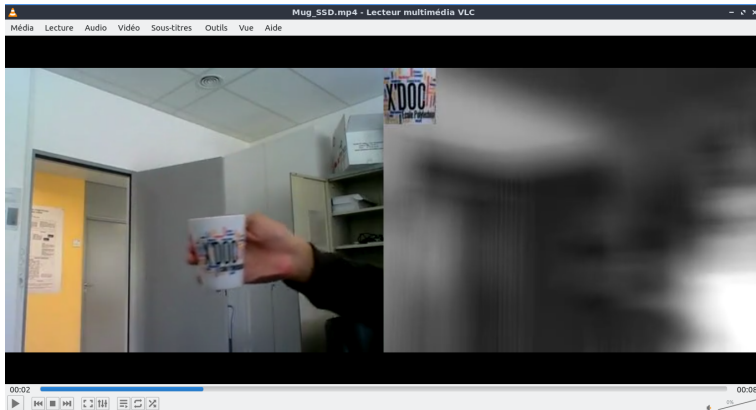
$$\lambda = \arg \min_{h \in \mathcal{H}} \mathcal{D}_k(T, X^h)$$

$+$ Model temporally adaptive

- Sensitive to deformations

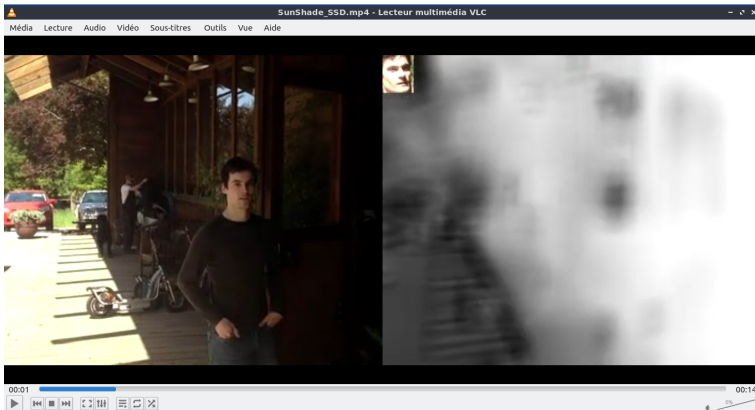- Sensitive to illumination variations

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Tracking by SSD Minimisation

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Tracking by SSD Minimisation

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Similarity measures

Correlation (Pearson) coefficient

The normalised dot product between two centred patches varies between -1 (total dissimilarity) and $+1$ (perfect match). Let $\tilde{X}$ be the average value of X in its support.

$$\chi(T, X) = \frac{\sum_{i < n}(T_i - \tilde{T})(X_i - \tilde{X})}{\sqrt{\sum_{i<n}(T_i - \tilde{T})^2}\sqrt{\sum_{i<n}(X_i - \tilde{X})^2}}$$

The detection algorithm then consists in searching within a location hypotheses space $\{X^h\}_{h \in \mathcal{H}}$ to find the one that maximises the correlation:
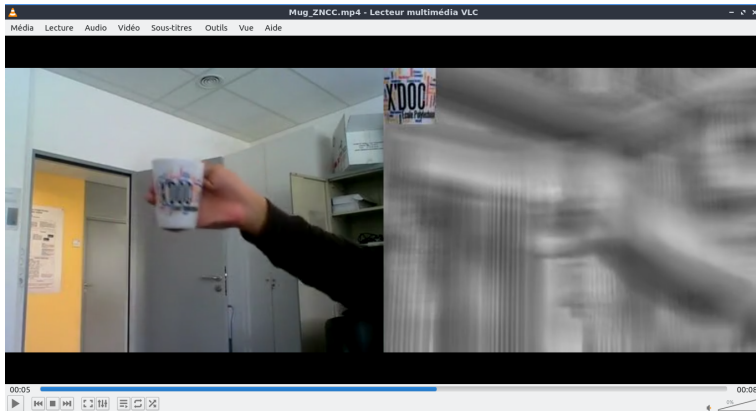
$$\lambda = \arg\max_{h \in \mathcal{H}} \chi(T, X^h)$$

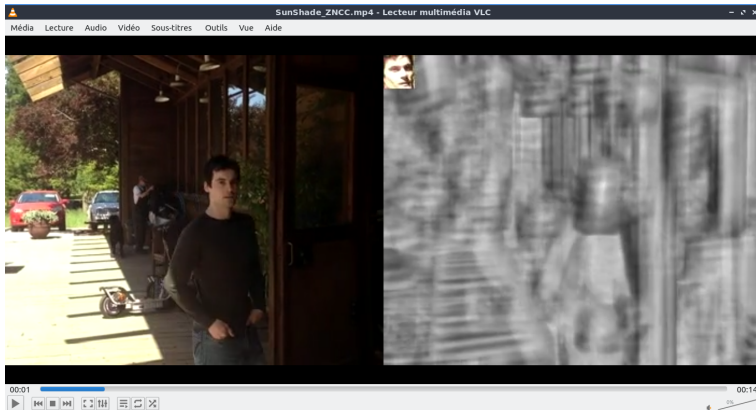$+$ Robust to illumination variations   - Sensitive to deformations

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Tracking by ZNCC Maximisation

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Tracking by ZNCC Maximisation

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
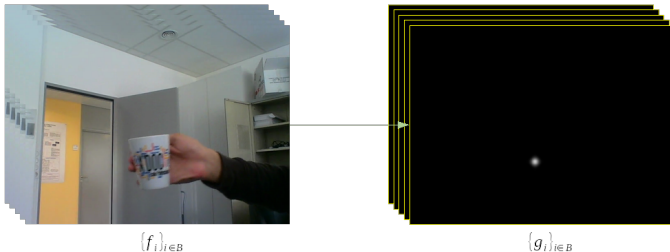Similarity between Distributions
Hough Transforms

Correlation Filters

- Centering and normalising the patch and image beforehand allows to use simple correlation filters (i.e. translation invariant dot-products).

- Correlation filters can be calculated very fastly by point-wise multiplication in the Fourier domain (with $\mathcal{O}(n \log n)$ complexity using the FFT).

- Correlation filters can be applied to any component or feature space calculated from the image.

- Custom / adapted Correlation Filters can be learned from training examples to produce ideal peak in the object position.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Learning Correlation Filters

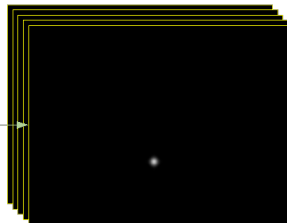$\{f_i\}_{i \in B}$ $\{g_i\}_{i \in B}$

Correlation filters can be learned from a collection of training images $\{f_i\}$ with their expected outputs $\{g_i\}$ (e.g. a 2d Gaussian with $\sigma = 2.0$ centred at the desired position). The problem is then to find the correlation kernel h such that $f_i \star h = g_i$, which in the Fourier domain becomes $F_i \times H^* = G_i$.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Learning Correlation Filters

$\{f_i\}_{i \in B}$ $\{g_i\}_{i \in B}$

[**Bolme10**] formulates the problem as the optimisation:

$$\arg \min_H \sum_{i \in B} |F_i \times H^* - G_i|^2$$

which has the closed solution:

$$H^* = \frac{\sum_{i \in B} G_i \times F_i^*}{\sum_{i \in B} F_i \times F_i^*}$$

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
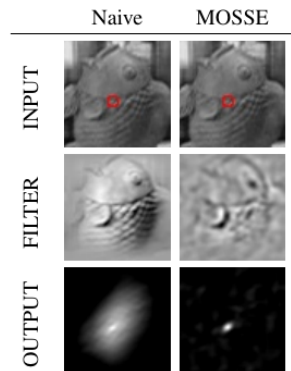Hough Transforms

Online Learning Correlation Filters

$$H^* = \frac{\sum_{i \in B} G_i \times F_i^*}{\sum_{i \in B} F_i \times F_i^*}$$

- The correlation filter h is learned *online*, from the first images of the sequence, or from some custom deformations of the input image (patch) f_i and output peak g_i.

- Note that using one single image and a trivial (Kronecker) peak, this is equivalent to "naive" normalised correlation.

[Bolme10]

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Online Learning Correlation Filters

To continuously adapt to deformable target, the correlation filter is updated at each frame t, according to some learning rate α:

$$H^* = \frac{A_t}{B_t}$$

with:
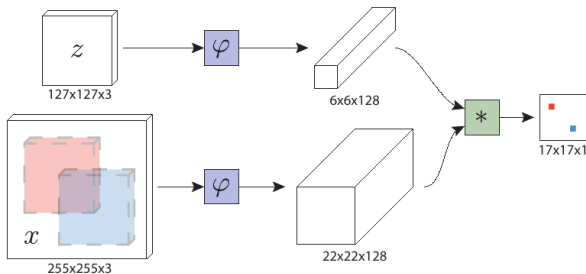
$$A_t = \alpha G_t \times F_t^* + (1 - \alpha)A_{t-1}$$

$$B_t = \alpha F_t \times F_t^* + (1 - \alpha)B_{t-1}$$

[Bolme10]

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

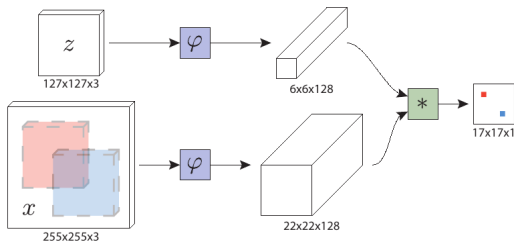# End-to-end Learning to Track using Siamese Networks

[Bertinetto-Valmadre16]

- The correlation can be applied to a latent space that has been learned *offline*.
- A fully convolutional network φ can be applied to images of any dimension.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

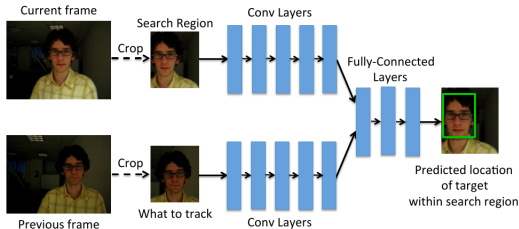# End-to-end Learning to Track using Siamese Networks

[Bertinetto-Valmadre16]

- The output is the correlation map: $y \simeq \varphi(x) \star \phi(z)$
- Training examples $\{x_i, z_i, v_i\}$ come from videos or object detection datasets, where v is the ground truth map valued in $\{-1, +1\}$.
- The loss is: $\mathcal{L}(y, v) = \log(1 + exp(-yv))$.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
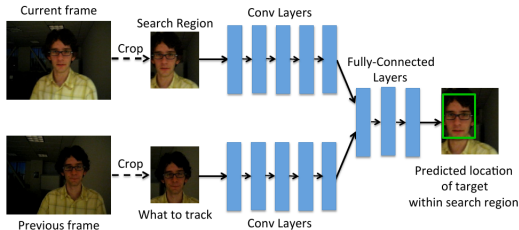Hough Transforms

More Siamese Networks

[Held16]

- Simpler Siamese networks without correlation have also been used.
- Training is made offline, on the $\mathcal{L}_1$ loss between ground truth and predicted bounding boxes.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
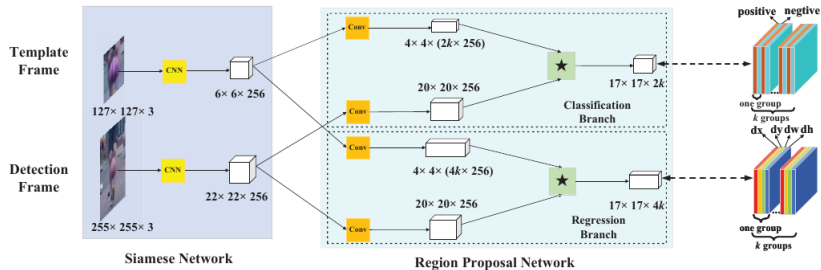Hough Transforms

More Siamese Networks

[Held16]

- Prediction is very fast (100 fps on GPU), but its accuracy and robustness is more dependent on the objects in the training set.
- Since it is not Fully Convolutional, it is no longer translation invariant, and the sizes of the track target and search region must be fixed.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
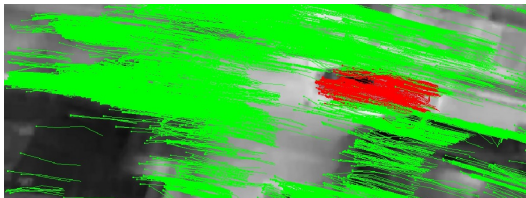Hough Transforms

More Siamese Networks

[Li18]

Siamese networks can also behave like customed object detectors,
e.g. using Region Proposal Network (backbone of Faster-RCNN)
combining Classification (here binary) and Regression of Candidate
boxes.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Using Apparent Displacement Fields

An algorithm of optical flow or point tracking can estimate the apparent displacements of all (or many) points $\{\mathbf{t}_j\}_j \in T$ of the model in the current image, by locally matching each point $\mathbf{t}_j$ to its counterpart $\mathbf{y}_j$ in the next image. The detection can then be performed by a statistics on the matched points $Y = \{\mathbf{y}_j\}$, for example the (2d) median:

$$\lambda = \arg \min_{\mathbf{m} \in Y} \sum_{\mathbf{y} \in Y} ||\mathbf{m} - \mathbf{y}||$$

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Similarity between distributions

Bhattacharyya Index

This index, whose value is between 0 and 1 (perfect match) measures the similarity between two distributions. Let H_T and H_X be the normalised histograms respectively associated to the model and to a location hypothesis:
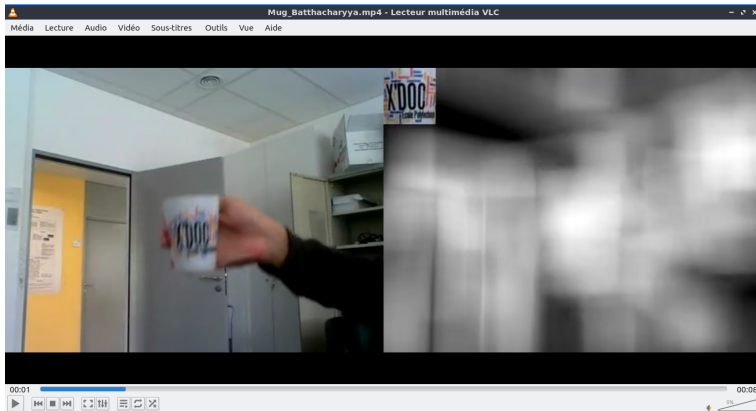
$$\mathcal{B}(H_T, H_X) = \sum_v \sqrt{H_T(v) H_X(v)}$$

+ Robust to any deformation

- Sensitive to illumination variations
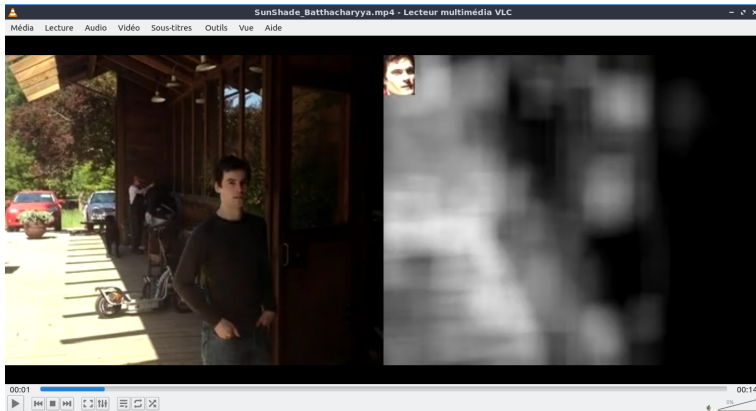- Poorly discriminant geometrically

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Tracking by distribution

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Tracking by distribution

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Generalised Hough Transforms

Generalised Hough Transforms (AKA "Implicit Shape Models") are object representations based on *co-occurrence of local elements*.
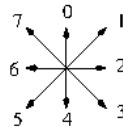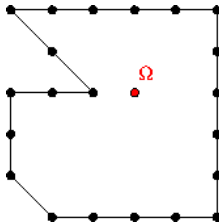
Modelling

- The model (template) T is sampled to a set of points $\{\mathbf{t}_i\}_i$.
- For each point $\mathbf{t}_i$, is computed:
  - an appearance index $\lambda(\mathbf{t}_i)$ (e.g.: colour, orientation, curvature...).
  - a vector $\mathbf{v}(\mathbf{t}_i)$ corresponding to the relative position of $\mathbf{t}_i$ with respect to the centre of template T.
  - (possibly) a confidence index $\omega(\mathbf{t}_i)$.
- The previous set of triplets is structured (array, search tree, random tree...) using the appearance index λ.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
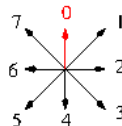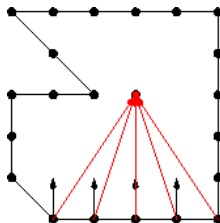Similarity between Distributions
Hough Transforms

Construction of the R-Table

The R-Table is a shape model, constructed from a prototype. Let Ω be an arbitrary centre of the prototype. Every point M of the prototype is indexed by a geometrical feature i, corresponding to the row indices of the R-table. The R-table is constructed by adding the displacement vector $\overrightarrow{M\Omega}$ in the line of index i.
For example consider the following contour points as a prototype, indexed by the normal direction to the contour, quantised to 8 values:
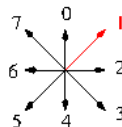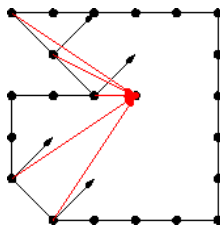
Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Construction of the R-Table

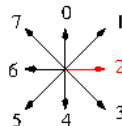| Index	List of vectors					
0	$\begin{pmatrix} -2 \\ -3 \end{pmatrix}$	$\begin{pmatrix} -1 \\ -3 \end{pmatrix}$	$\begin{pmatrix} 0 \\ -3 \end{pmatrix}$	$\begin{pmatrix} 1 \\ -3 \end{pmatrix}$	$\begin{pmatrix} 2 \\ -3 \end{pmatrix}$	end

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Construction of the R-Table

Index	List of vectors					
0	$\binom{-2}{-3}$	$\binom{-1}{-3}$	$\binom{0}{-3}$	$\binom{1}{-3}$	$\binom{2}{-3}$	end
1	$\binom{2}{-3}$	$\binom{3}{-2}$	$\binom{1}{0}$	$\binom{2}{1}$	$\binom{3}{2}$	end

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Construction of the R-Table

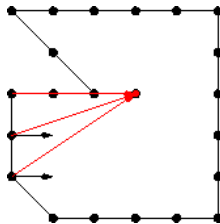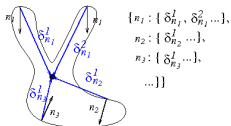| Index	List of vectors					
0	$\begin{pmatrix} -2 \\ -3 \end{pmatrix}$	$\begin{pmatrix} -1 \\ -3 \end{pmatrix}$	$\begin{pmatrix} 0 \\ -3 \end{pmatrix}$	$\begin{pmatrix} 1 \\ -3 \end{pmatrix}$	$\begin{pmatrix} 2 \\ -3 \end{pmatrix}$	end
1	$\begin{pmatrix} 2 \\ -3 \end{pmatrix}$	$\begin{pmatrix} 3 \\ -2 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 2 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 3 \\ 2 \end{pmatrix}$	end
2	$\begin{pmatrix} 3 \\ -2 \end{pmatrix}$	$\begin{pmatrix} 3 \\ -1 \end{pmatrix}$	$\begin{pmatrix} 3 \\ 0 \end{pmatrix}$	end		

and so on...

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms
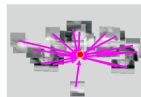
Generalised Hough Transforms: variants

The R-Table can be constructed using different supports, and different appearance indexes:

Support: Contour	Interest points	All pixels
Index: Normal	Patch codebook	Gradient orientation

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Generalised Hough Transforms: Detection

In a Generalised Hough Transforms, each point, according to its appearance, indicates, by a list of *votes*, a set of location hypotheses of the object centre. The centres that got maximal number of votes are considered as the most probable positions.

Detection

- For each point $\mathbf{x}$ of the image, calculate its appearance index $\lambda(\mathbf{x})$.
- For all points $\mathbf{t}_j$ of the model such that $\lambda(\mathbf{t}_j) = \lambda(\mathbf{x})$, increment the vote map $H(\mathbf{x} + \mathbf{v}(\mathbf{t}_j)) += \omega(\mathbf{t}_j)$.
- The most probable object position is $\arg \max_{\mathbf{x}} H(\mathbf{x})$.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Matching measures
Similarity between Distributions
Hough Transforms

Generalised Hough Transform: Object Detection

Initial: $H(\mathbf{x}) = 0$ everywhere.
For all image point $\mathbf{x}$,
let $\lambda(\mathbf{x})$ the quantised derivative.
   For all occurence j of the R-Table
   associated to $\lambda(\mathbf{x})$, do:

$$H(\mathbf{x} + \delta^j_{\lambda(\mathbf{x})}) \mathrel{+}= \omega^j_{\lambda(\mathbf{x})}$$

The best object candidates are
then located at the maxima of H
(Right: Hough transform and the
10 best candidate cars).

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Mean-Shift Algorithm

Introduction to the Mean-Shift

Mean-Shift 1/2

The Mean-Shift is an *iterative* algorithm for estimating the *mode* of a distribution. For each iteration:

- $S = \{\mathbf{x}_i\}_i \subset \mathcal{E}$ a set of points in a Euclidian space.
- $K : \mathcal{E} \longrightarrow \mathbb{R}_+$ a kernel function, that determines the contribution of each point.
- $m(\mathbf{x}) = \dfrac{\sum_S K(\mathbf{x} - \mathbf{x}_i)\mathbf{x}_i}{\sum_S K(\mathbf{x} - \mathbf{x}_i)}$ the current average around $\mathbf{x}$.
- $v(\mathbf{x}) = m(\mathbf{x}) - \mathbf{x}$ the mean-shift vector.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering
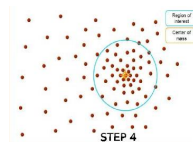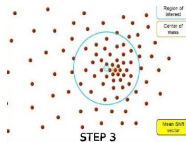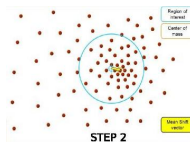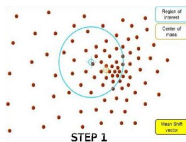
Mean-Shift Algorithm

Introduction to the Mean-Shift

Mean-Shift 2/2

The Mean-Shift algorithm iteratively displaces the point $\mathbf{x}$ toward the local average. For each iteration (contd) :

- $\mathbf{x} \leftarrow m(\mathbf{x})$
- The algorithm converges when $\mathbf{x} = m(\mathbf{x})$
- The Mean-Shift trajectory is $\{\mathbf{x}, m(\mathbf{x}), m(m(\mathbf{x})), \ldots\}$

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Mean-Shift Algorithm

Kernel Function

The kernel K, that determines the contribution weight of the different points in the average, is usually isotrope in the Euclidian space, i.e.:

$$K(\mathbf{y}) = k(||\mathbf{y}||).$$

k is usually non increasing and derivable; the Gaussian kernel is often used:

$$k(y) = e^{-y^2}, \text{ i.e. } K(\mathbf{y}) = e^{-||\mathbf{y}||^2}. \tag{1}$$

The kernel function K is related to a regularisation criterion used in the estimation of a distribution (Parzen's kernel technique).

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Mean-Shift Algorithm

Estimating the local distribution

Let f be a (quantized) value of the image in one point (Gray level, colour, or any scalar feature calculable in each pixel). Estimating the distribution associated to f in the neighbourhood of the current point $\mathbf{x}$ defined by the support S can be made as follows:

$$h_{\mathbf{x}}(u) = \frac{\sum_S k(||\mathbf{x} - \mathbf{x}_i||)\delta_u^{f(\mathbf{x}_i)}}{\sum_S k(||\mathbf{x} - \mathbf{x}_i||)} \tag{2}$$

with $\delta_u^v = 1$ if $u = v$ and $\delta_u^v = 0$ if $u \neq v$.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Mean-Shift Algorithm

Similarity and Batthacharyya's index

If $h_R(u)$ corresponds to a reference distribution (i.e. the model), its similarity with the current local distribution can be estimated using Batthacharyya' index:

$$\mathcal{B}_{\mathbf{x}} = \sum_u \sqrt{h_{\mathbf{x}}(u) h_R(u)}.$$

The principle of Mean-Shift consists in using *smooth* distributions, in order to make the similarity function regular enough so that its behaviour can be predicted from its spatial derivatives (Taylor's Formula).

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Mean-Shift Algorithm

Similarity and Batthacharyya's index

A first order approximation of Batthacharyya's index around the first estimation point $\mathbf{x_0}$ provides:

$$
\begin{aligned}
\mathcal{B}_\mathbf{x} &\simeq \tfrac{1}{2}\sum_u \sqrt{h_{\mathbf{x_0}}(u)h_R(u)} + \tfrac{1}{2}\sum_u h_\mathbf{x}(u)\sqrt{\frac{h_R(u)}{h_{\mathbf{x_0}}(u)}} \\
&\simeq \tfrac{1}{2}\mathcal{B}_{\mathbf{x_0}} + \tfrac{1}{2}\frac{\sum_S \omega_i k(\|\mathbf{x}-\mathbf{x}_i\|)}{\sum_S k(\|\mathbf{x}-\mathbf{x}_i\|)} \, .
\end{aligned}
$$

The 1st term being independent of $\mathbf{x}$, maximising $\mathcal{B}_\mathbf{x}$ is equivalent to maximising the 2nd term, with:

$$
\omega_i = \sum_u \sqrt{\frac{h_R(u)}{h_{\mathbf{x_0}}(u)}} \delta_u^{f(\mathbf{x}_i)} = \sqrt{\frac{h_R(f(\mathbf{x}_i))}{h_{\mathbf{x_0}}(f(\mathbf{x}_i))}} \tag{3}
$$

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Mean-Shift Algorithm

Mean-Shift Tracking Algorithm

Mean-Shift Tracking

input: $\{y, h_R(u)\}$ (previous frame).

1. Calculate the histogram $h_y(u)$ in current frame (Eq. 2).
2. Calculate the weights ω_i in each point of the support (Eq. 3).
3. mean-shift: Calculate the new position x:

$$x = \frac{\sum_S \omega_i g(||y - x_i||)x_i}{\sum_S \omega_i g(||y - x_i||)} \qquad (4)$$

avec $g(x) = -k'(x)$ (Eq. 1).

4. If $||x - y|| < \varepsilon$, stop. Else $y \leftarrow x$, and go back to step (1).

output: Final position x and a new reference histogram (model) $h_R(u) = h_x(u)$.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Mean-Shift Algorithm

Object Tracking using Mean-Shift

e
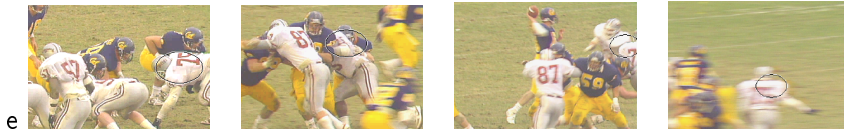
- Support: interior of an ellipse.
- Value space: quantized RGB $16 \times 16 \times 16$.
- Average number of Mean-Shift iterations: 4 (See graph on the right).
- Works of Comaniciu, Ramesh et Meer [Comaniciu03].

iterations per frame.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering
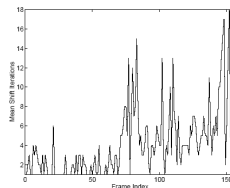
Kalman Filter
Particle Filtering

Predictive Filtering

Kalman Filter

State estimation technique based on prediction and correction steps through linear stochastic equation, optimal under the assumption of Gaussian distribution of the state space.

Condensation algorithm

State estimation technique based on sampling the state space posterior distribution from the visual observation, and iteratively propagating new samples from successive images.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Kalman Filter
Particle Filtering

Kalman Filter

The principle of the Kalman is to estimate the state $\Theta \in \mathbb{R}^n$ of the discrete time process governed by the linear stochastic equation:
$$\Theta_t = A\Theta_{t-1} + BU_t + W_{t-1}$$
using a measurement $X_t \in \mathbb{R}^m$ that relates to Θ as follows:
$$X_t = H\Theta_t + V_t$$
A is a $n \times n$ matrix relating two states at consecutive times.
B is an (opt.) $n \times l$ matrix related to a control input $U \in \mathbb{R}^l$.
H is a $m \times n$ matrix relating the state to the measurement.
V and W are ind., white Gaussian centered random vectors:
$$p(V) \simeq \mathcal{N}(O, R); p(W) \simeq \mathcal{N}(O, Q)$$

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Kalman Filter
Particle Filtering

Kalman Filter algorithm - from [Welsh01]

Init. Start with initial estimates of $\hat{\Theta}_0$ and P_0.
Then for $t > 0$:

Kalman Filter (1) Prediction phase

1 Project the state ahead
$$\hat{\Theta}_t^- = A\hat{\Theta}_{t-1} + BU_t$$

2 Project the error covariance ahead
$$P_t^- = AP_{t-1}\, {}^tA + Q$$

Kalman Filter (2) Correction phase

1 Compute the Kalman gain
$$K_t = P_t^-\, {}^tH(HP_t^-\, {}^tH + R)^{-1}$$

2 Update estimate with new measurement
$$\hat{\Theta}_t = \hat{\Theta}_t^- + K_t(X_t - H\hat{\Theta}_t^-)$$

3 Update the error covariance
$$P_t = (I - K_tH)P_t^-$$

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Kalman Filter
Particle Filtering

Kalman Filtering, Implementations

Order 0 Model

- State $\Theta_t = (x_t, y_t)$, Observation $X_t = (x_t, y_t)$.
- Transition Matrix $A = I_2$; Observation Matrix $H = I_2$.

Order 1 Model

- State $\Theta_t = (x_t, y_t, v_t^x, v_t^y)$, Observation $X_t = (x_t, y_t)$.

- Transition Matrix $A = \begin{pmatrix} 1 & 0 & \delta_t & 0 \\ 0 & 1 & 0 & \delta_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$.

- Observation Matrix $H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Kalman Filter
Particle Filtering

Condensation algorithm

Condensation algorithm [Isard98]

Iterate:
input: $\{s_{t-1}^{(n)}, \pi_{t-1}^{(n)}, c_{t-1}^{(n)}\}_{n \leq N}$ the set of N old samples
output $\{s_t^{(n)}, \pi_t^{(n)}, c_t^{(n)}\}_{n \leq N}$ the set of N new samples

1 **Select** a sample $\hat{s}_t^{(n)}$ as follows:

- select (uniformly) a random number $r \in [0, 1]$
- find the smallest j for which $c_{t-1}^{(j)} \leq r$
- set $\hat{s}_t^{(n)} = s_{t-1}^{(j)}$

2 **Predict** the new sample $s_t^{(n)}$:
$$s_t^{(n)} = \arg\max P(\hat{\Theta}_t = s / \Theta_{t-1} = \hat{s}_t^{(n)})$$

3 **Correct** from the measure and update the current weight:
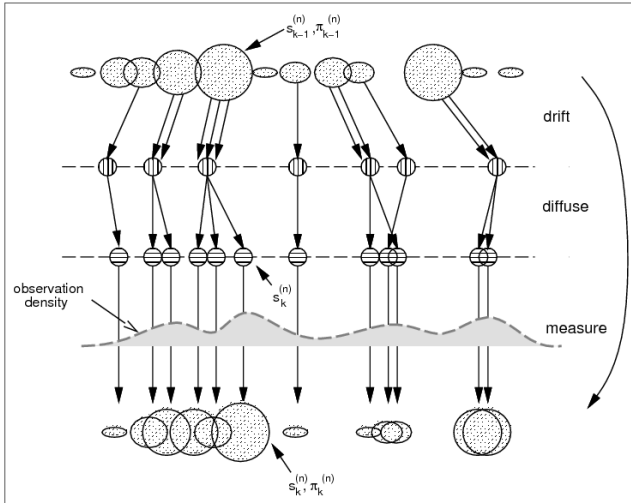$$\pi_t^{(n)} = P(X_t / \Theta_t = s_t^{(n)}), \text{ normalizing so that } \sum_{i \leq N} \pi_t^{(i)} = 1$$

then recompute the cumulative distribution:
$$c_t^{(0)} = 0; c_t^{(n)} = c_t^{(n-1)} + \pi_t^{(n)} (1 \leq n \leq N)$$

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Kalman Filter
Particle Filtering

Condensation algorithm, cont.

from [Isard98]

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Kalman Filter
Particle Filtering

Object Tracking - Conclusion

Prediction

- Allows to estimate the initial state (position, size,...).
- Dynamical model: likelihood, kinematics,...
- Sets a trade-off between innovation (data) and prediction (model).

Observation

- Appearance model: colour, orientation, features...
- Global description: vector, distribution, co-occurence,...
- Discrimination *vs* Invariance.
- Evaluates the possible states around the initial (predicted) state.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Kalman Filter
Particle Filtering

Object Tracking - Conclusion

Learning Visual Tracking

State-of-the-art learning based trackers are mostly
Tracking-by-Detection approaches that combine different methods:

- Offline learning of objects representation (Deep features).
- Online learning of correlation filters.
- Asymmetric Siamese Networks.

Prediction is generally not learned, but managed using classic techniques.

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Kalman Filter
Particle Filtering

Bibliography - Tracking

📄 **[Jalal12]** A.S. JALAL and V. SINGH
The State-of-the-Art in Visual Object Tracking
Informatica 36 (2012) 227-248

📄 **[VOT14]** M. KRISTAN et al.
The Visual Object Tracking VOT2014 challenge results
Visual Object Tracking Workshop 2014 at ECCV2014, 2014

📄 **[Comaniciu03]** D. COMANICIU, V. RAMESH and P.MEER
Kernel-based object tracking
Pattern Analysis and Machine Intelligence, 25(5), 564-575 (2003)

📄 **[Held16]** D.HELD, S. THRUN and S. SARAVESE
Learning to Track at 100 FPS with Deep Regression Networks
ECCV 2016. LNCS vol. 9905

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Kalman Filter
Particle Filtering

Bibliography - Tracking

📄 **[Bolme10]** D.S. BOLME, J.R. BEVERIDGE, B.A. DRAPER and Y.M. LUI,
Visual object tracking using adaptive correlation filters
CVPR (2010) 2544-2550

📄 **[Bertinetto-Valmadre16]** L. BERTINETTO, J. VALMADRE, J.F. HENRIQUES, A. VEDALDI and P.H.S. TORR
Fully-Convolutional Siamese Networks for Object Tracking
ECCV 2016 Workshops, LNCS vol. 9914

📄 **[Li18]** B. LI, J. YAN, W. WU, Z. ZHU and X. HU
High Performance Visual Tracking with Siamese Region Proposal Network
CVPR 2018, pp. 8971-8980

Introduction
Observations and detection
Tracking a distribution
Predictive Filtering

Kalman Filter
Particle Filtering

Bibliography - Predictive filtering

📓 **[Isard98]** M. ISARD and M. BLAKE
CONDENSATION - CONditional DENSity propagATION for visual tracking
Int. Journal of Computer Vision (1998) 29(1), 5-28 (1998)

📓 **[Welsh01]** G. WELSH and G. BISHOP
An Introduction to the Kalman Filter
Tutorial of ACM SIGGRAPH (2001)