Visual Tracking

T. Chateau

Pascal Institute, UMR 6602, UBP/CNRS/IFMA

October 24, 2012

Content

Content

Visual Tracking

Definition

Visual Tracking is the process of locating, identifying, and determining the dynamic configuration of one or many moving (possibly deformable) objects (or parts of objects) in each frame of one or several cameras

Human equivalent

Follow something with your eyes

Visual Tracking (before beginning)

State Vector

The dynamic configuration of the the tracked object at time k is modelled by a State vector denoted:

$$\mathbf{x}_k$$

State Sequence

The state sequence is given by the set (sequence) of State vectors, denoted:

$$\mathbf{X} \doteq \{\mathbf{x}_k\}_{k=1,\dots,K}$$

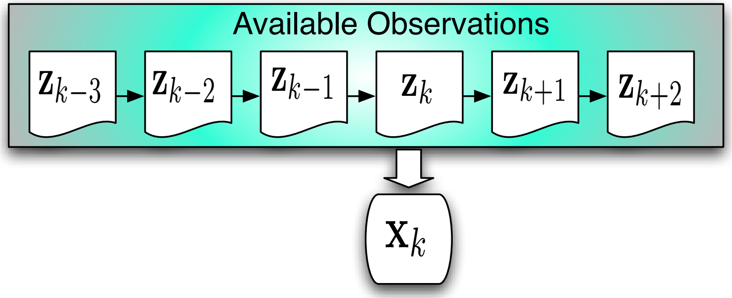
Observation

Observation: $\mathbf{Z} \doteq \{\mathbf{z}_k\}_{k=1,\dots,K}$

On-line and Off-line Tracking

Off-line Tracking (Deferred Tracking)

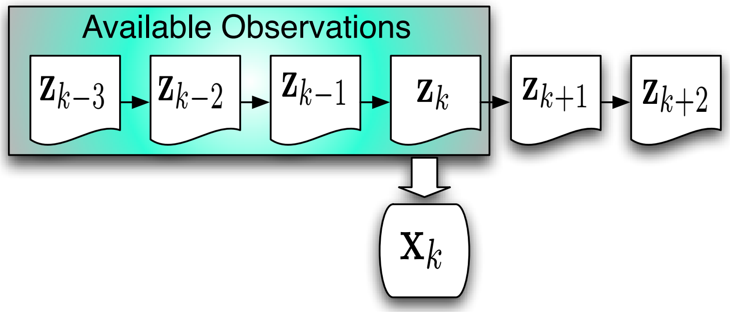
Estimation of the state $\mathbf{x}_k$ uses the entire observation sequence $\mathbf{Z} \doteq \{\mathbf{z}_k\}_{k=1,\dots,K}$



On-line and Off-line Tracking

On-line Tracking

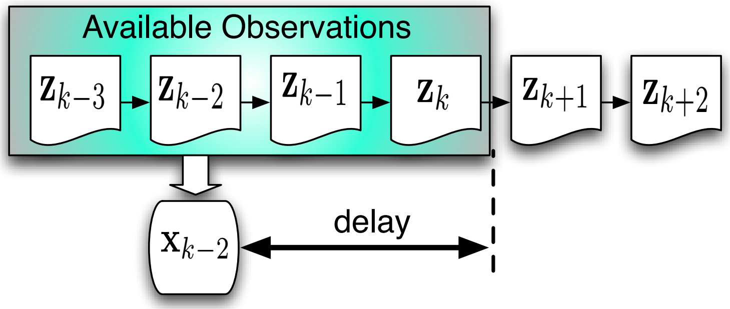
Estimation of the state $\mathbf{x}_k$ uses the current and past observation: $\mathbf{z}_{0:k}$



On-line and Off-line Tracking

Delayed Tracking

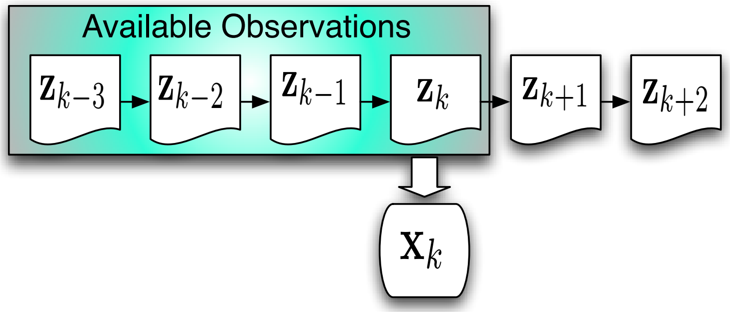
Estimation of the state needs current, past and a part (delay) of future observation



On-line and Off-line Tracking

On-line Tracking

For robotic applications: estimation of the state $\mathbf{x}_k$ uses the current and past observation: $\mathbf{z}_{0:k}$



Why is Visual Tracking Difficult ?

Hidden State

The state $\mathbf{X}$ is a hidden state and must be deduced from observation

Tracking Challenges

- Object Modeling: how to define what an object is in terms that can be interpreted by a computer ?
- Appearance Change: The observation of an object changes according to many parameters (illumination conditions, occlusions, shape variation...)
- Kinematic Modelling: How to inject priors on object kinematic and interactions between objects.

Tracking Challenges: Object Modelling

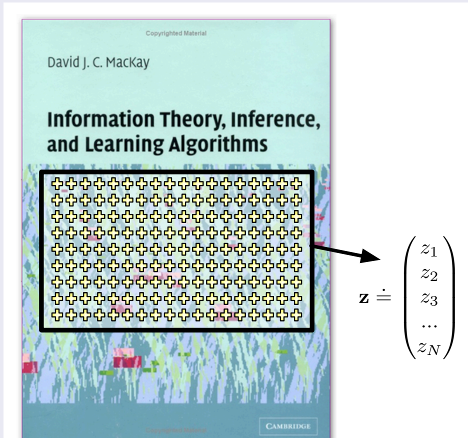
Generic, discriminative model

Build a visual description of the object:

- Generic enough to encode the entire variability of the object
- Discriminative enough to separate the object into the images (cluttered background)

Tracking Challenges: object description

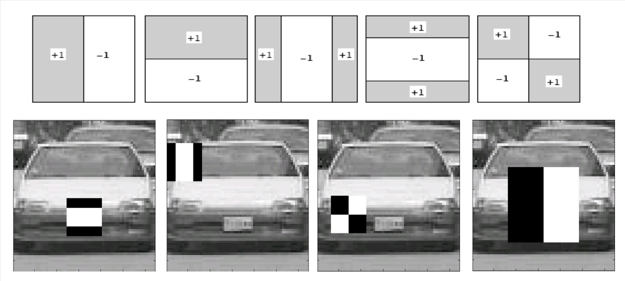
Example 1



Tracking Challenges: object description

Example 2

Generic object tracking (example: vehicle tracking):

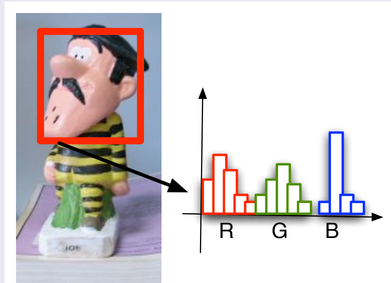


The Image is described with a vector of selected "Haar-like" wavelets

Tracking Challenges: object description

Example 3

color based tracking:



RGB (marginalized histogram)

The Image is described with a color histogram

Tracking Challenges: Appearance Variation

Several Illumination Conditions and poses

Content

The Toy Example

Object Tracking from a single static camera

Estimation of the 2D position of a moving object

The Toy Example

State Vector

$$\mathbf{x}_k \doteq \{x_k, y_k\}$$

, the position of the gravity center of the object (into the image reference plane)

Observation function

Based on a difference image :

Compute diff. image: $\mathtt{I}_k^{diff} = \mathtt{I}_k^{ref} - \mathtt{I}_k$

Update Ref. image: $\mathtt{I}_{k+1}^{ref} = \alpha.\mathtt{I}^{ref} + (1-\alpha).\mathtt{I}_k$

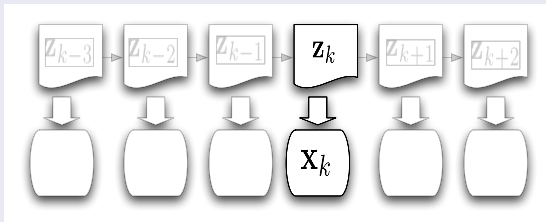
The Toy Example

Example of diff. image

Tracking by Detection

Tracking by Detection

The state $\mathbf{x}_k$ at time k depends only on the observation $\mathbf{z}_k$ at time k



Tracking by Detection

Application to the toy example

$\mathbf{x}_k = \mathbf{f}(\mathbf{z}_k)$ where $\mathbf{f}$ is a function given the position of the foreground pixel which has the most moving neighbours pixels (clustering method not developed here)

Matlab demonstration

Introduction to Visual Tracking
Modelling Visual Tracking
Stochastic Filters for Visual Tracking
Some examples
My First Tracker
Classification of Visual Tracking approaches
Overview of non probabilistic methods
Probabilistic Approaches to Visual Tracking

Tracking by Detection

Conclusion

- Tracking by detection needs a function $\mathbf{x}_k = \mathbf{f}(\mathbf{z}_k)$
- No prior on motion between two images is injected into the algorithm

T. Chateau Blaise Pascal University

Introduction to Visual Tracking
Modelling Visual Tracking
Stochastic Filters for Visual Tracking
Some examples
My First Tracker
Classification of Visual Tracking approaches
Overview of non probabilistic methods
Probabilistic Approaches to Visual Tracking

Injecting priors on motion

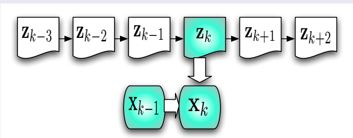
Assumption

We know (prior) a bound on the relative motion between images $\mathtt{I}_{k-1}$ and $\mathtt{I}_k$.

Consequence

The state $\mathbf{x}_k$ at time k depends only on observation $\mathbf{z}_k$ at time k and the previous state $\mathbf{x}_{k-1}$:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{z}_k, \mathbf{x}_{k-1})$$



T. Chateau Blaise Pascal University

Introduction to Visual Tracking
Modelling Visual Tracking
Stochastic Filters for Visual Tracking
Some examples
My First Tracker
Classification of Visual Tracking approaches
Overview of non probabilistic methods
Probabilistic Approaches to Visual Tracking

Injecting priors on motion

Application to the toy example

The observation function is reduced to a Region of Interest (ROI) around the previous estimated state.

Matlab demonstration

T. Chateau Blaise Pascal University

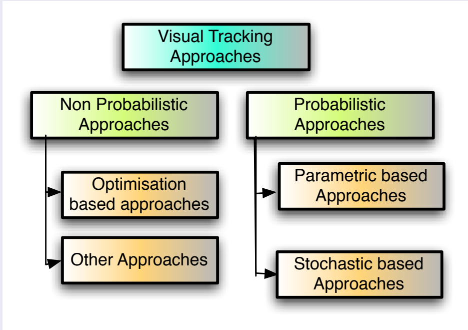
Injecting priors on motion

Conclusion

- Injecting priors on motion reduces the search state space
- The resulting solution is a basic "tracker"

Classification of Visual Tracking approaches

Non-Probabilistic vs Probabilistic Approaches



Tracking as an optimisation problem

State

The State vector is an unknown parameter vector which can be estimated using optimisation techniques :

$$\hat{\mathbf{x}}_k = \arg \min_{\mathbf{x}_k \in \mathcal{X}} \mathcal{E}(\mathbf{x}_k, \mathbf{z}_k)$$

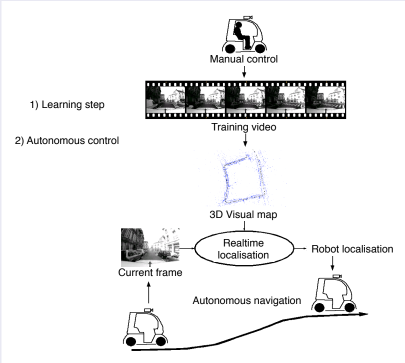
The search space $\mathcal{X}$ is often reduced using priors on motion and previous estimation.

example of Tracking solutions using optimisation techniques

MeanShift, Comaniciu

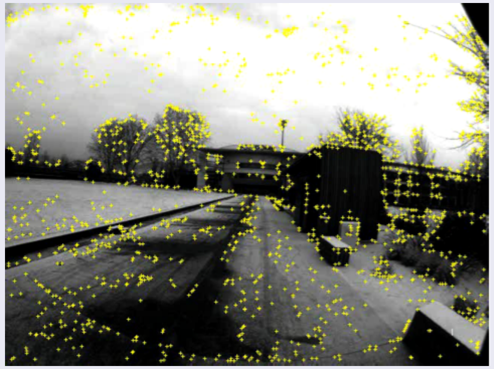
example of Tracking solutions using optimisation techniques

Localisation, Royer (Lasmea)



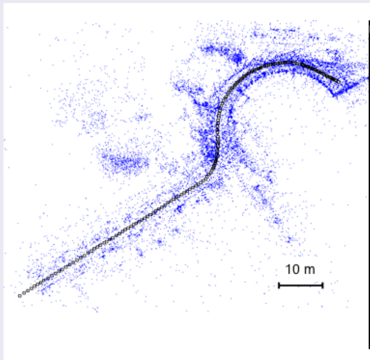
example of Tracking solutions using optimisation techniques

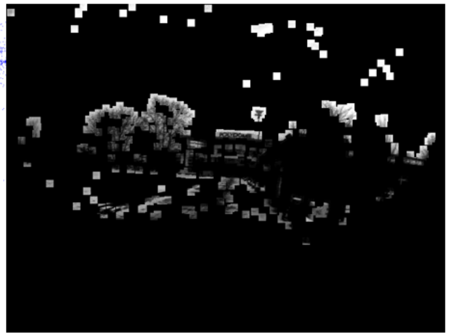
Localisation, Royer (Lasmea)



example of Tracking solutions using optimisation techniques

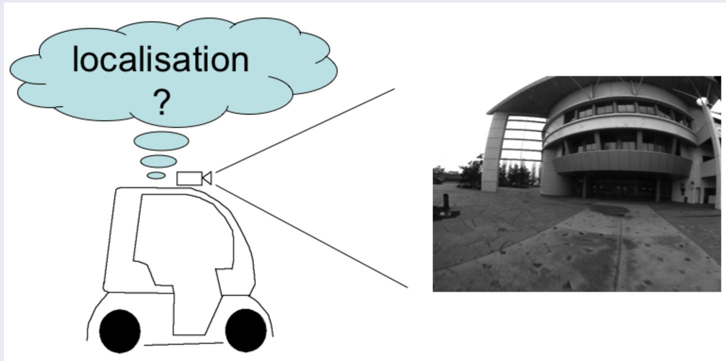
Localisation, Royer (Lasmea)





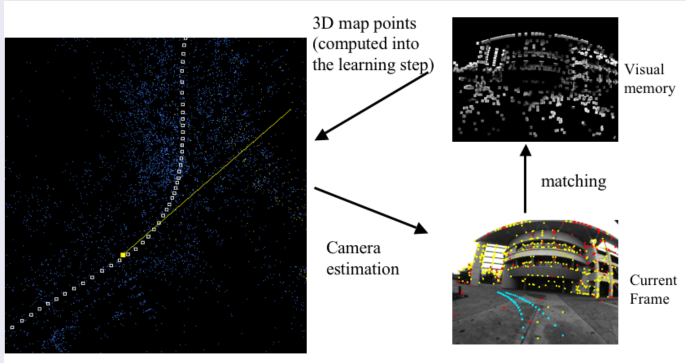
example of Tracking solutions using optimisation techniques

Localisation, Royer (Lasmea)



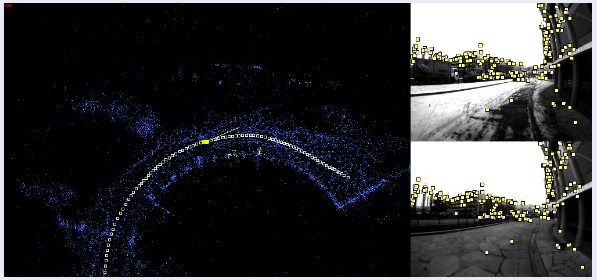
example of Tracking solutions using optimisation techniques

Localisation, Royer (Lasmea)



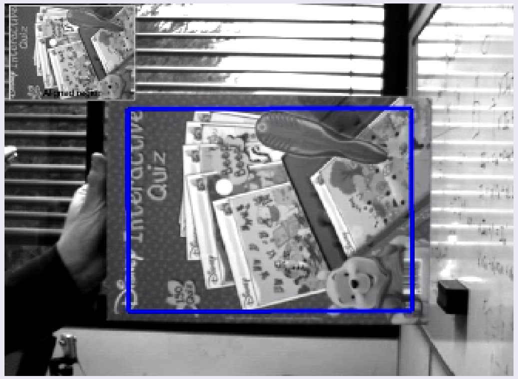
example of Tracking solutions using optimisation techniques

Localisation, Royer (Lasmea)



example of Tracking solutions using optimisation techniques

ESM, Malis, INRIA

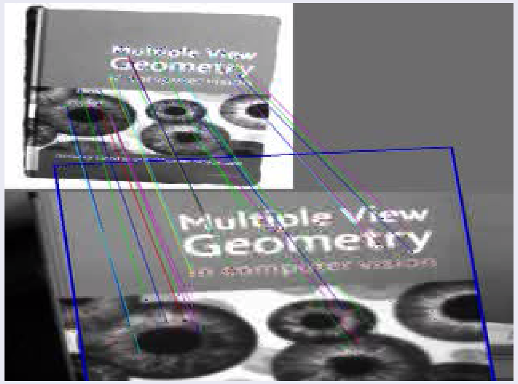


Tracking : Other non-probabilistic approaches

Using Machine Learning, Williams, RVM, Relevance Vector Machine

Tracking : Other non-probabilistic approaches

Tracking by Detection, V. Lepetit, EFPL



Probabilistic Approaches to Visual Tracking

Random Vectors

Both the state $\mathbf{X}$ and the observation $\mathbf{Z}$ are random vectors:

$$\mathbf{X} \in \mathcal{X} \text{ and } \mathbf{Z} \in \mathcal{Z}$$

Joint Probability

- The Probability of a sate sequence is given by:

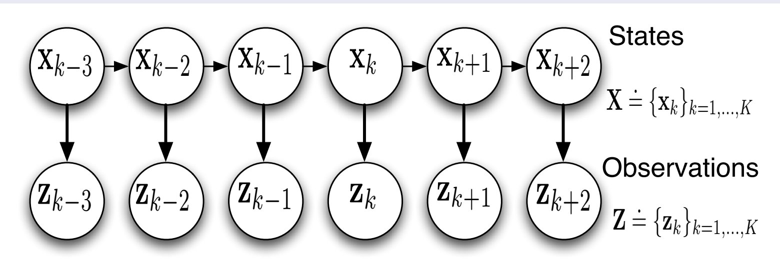
$$p(\mathbf{X}|\mathbf{Z}) = p(\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_K | \mathbf{z}_1; \mathbf{z}_2; \dots; \mathbf{z}_K)$$

- The final output of a Visual Tracking process is an estimate $\hat{\mathbf{X}}$

The Recursive Bayesian Estimation Approach

Dynamic Bayesian Network representation

First order Markovian assumption: the object configuration at time k, $\mathbf{x}_k$, depends only on the previous state $\mathbf{X}_{k-1}$.



Recursive state-space Bayesian estimation approach

Posterior distribution

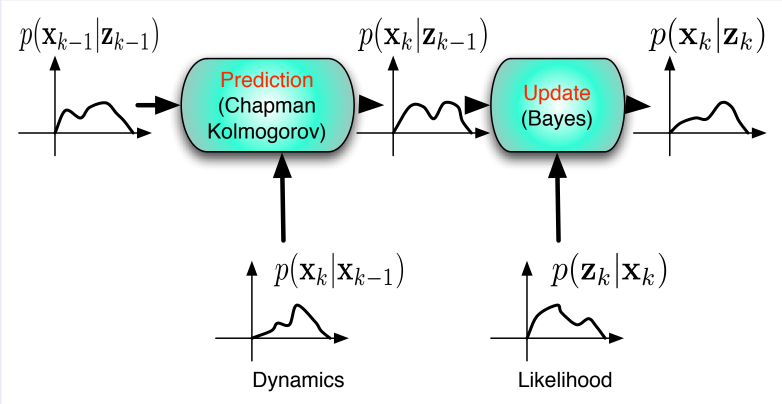
The belief about the current state $\mathbf{x}_k$ is expressed by a probability distribution:

$$p(\mathbf{x}_k|\mathbf{z}_k)\text{: POSTERIOR DISTRIBUTION}$$

How to recursively compute $p(\mathbf{x}_k|\mathbf{z}_k)$?

computing $p(\mathbf{x}_k|\mathbf{z}_k)$ from $p(\mathbf{x}_{k-1}|\mathbf{z}_{k-1})$

A two steps algorithm

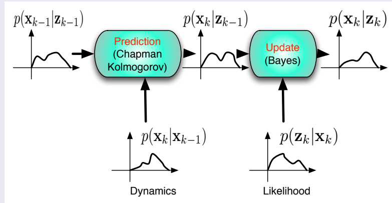


computing $p(\mathbf{x}_k|\mathbf{z}_k)$ from $p(\mathbf{x}_{k-1}|\mathbf{z}_{k-1})$

Prediction step (dynamical model)

Chapman-Kolmogorov equation:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}$$



computing $p(\mathbf{x}_k|\mathbf{z}_k)$ from $p(\mathbf{x}_{k-1}|\mathbf{z}_{k-1})$

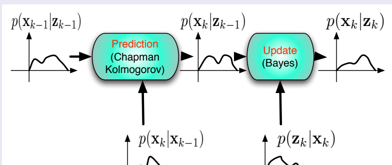
Update step

Bayes theorem:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})}$$

with :

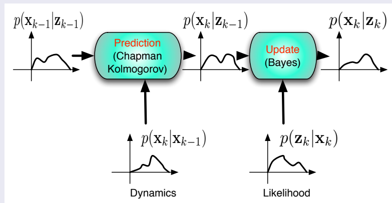
$$p(\mathbf{z}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})d\mathbf{x}_k$$



computing $p(\mathbf{x}_k|\mathbf{z}_k)$ from $p(\mathbf{x}_{k-1}|\mathbf{z}_{k-1})$

Recursive Bayesian filtering distribution

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = C^{-1}p(\mathbf{z}_k|\mathbf{x}_k)\int_{\mathbf{x}_{k-1}} p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}$$



computing $p(\mathbf{x}_k|\mathbf{z}_k)$ from $p(\mathbf{x}_{k-1}|\mathbf{z}_{k-1})$

Partial Conclusion

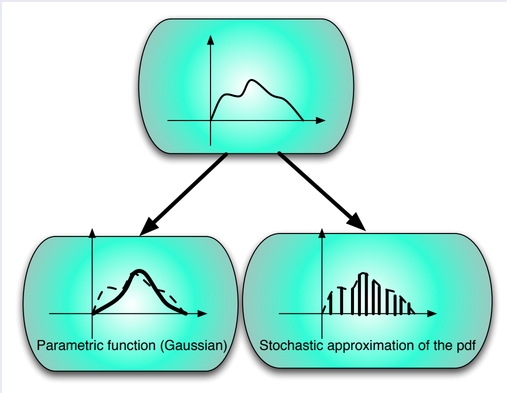
- The recursive bayesian filtering distribution provides an efficient solution to compute the posterior at time k ($p(\mathbf{x}_k|\mathbf{z}_k)$) from the posterior at time $k-1$ ($p(\mathbf{x}_{k-1}|\mathbf{z}_{k-1})$) , the dynamic model ($p(\mathbf{x}_k|\mathbf{x}_{k-1})$), and the likelihood ($p(\mathbf{z}_k|\mathbf{x}_k)$)
- Operations (integrals, products) on pdf have to be done:

Question

how to define probabilities such that operations like product and integration become tractable ?

Modelling pdf

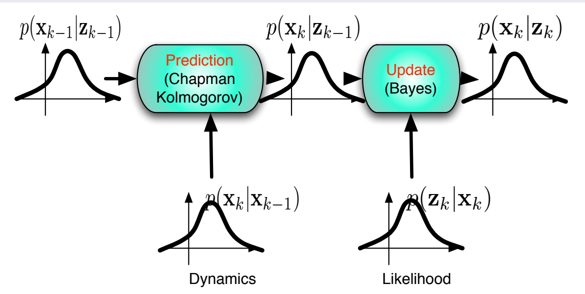
Parametric and stochastic models



Parametric models (Kalman,...)

Kalman filter

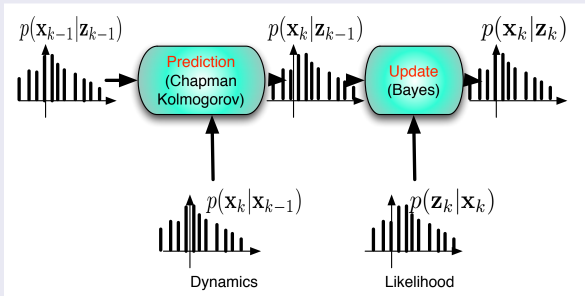
Assumption: all pdf are modelized with Gaussian



Stochastic models (Particle filters,...)

Particle filters

All pdf are approximated by a set of samples.



Probabilistic filters

Partial conclusion

- Kalman filters and derived: we assume that the unknown pdf can be modelized by a parametric function
- Stochastic solutions: approximation of the pdf by a set of particles.

Next

Stochastic approaches to bayesian filter are developed in the next section

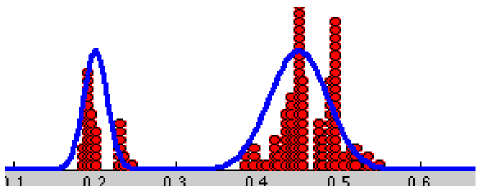
Content

Approximating pdf(s)

Set of particles model

$$p(\mathbf{x}) \approx \{\mathbf{x}^n\}_{n=1,..,N}$$

$$p(\mathbf{x}) \approx \sum_{n=1}^{N} \delta(\mathbf{x} - \mathbf{x}^n)$$

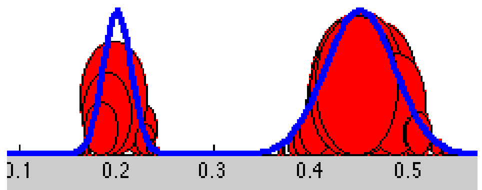


Approximating pdf(s)

Set of weighted particles model

$$p(\mathbf{x}) \approx \{\mathbf{x}^n, \pi^n\}_{n=1,..,N}$$

$$p(\mathbf{x}) \approx \sum_{n=1}^{N} \pi^n \delta(\mathbf{x} - \mathbf{x}^n)$$



SIR (algo. and matlab simulation) (1996)

CONDENSATION Algorithm

Init: particles $\{(\mathbf{x}_0'^n, \mathbf{1}/N)\}_{n=1}^N$ according to the initial distribution $\mathbf{x}_0$

for $k = 1, ..., K_{end}$ **do**

Prediction : generation of $\{(\mathbf{x}_k^n, \mathbf{1}/N)\}_{n=1}^N$ from $p(\mathbf{x}_k|\mathbf{X}_{k-1} = \mathbf{x}_{k-1}'^n)$

Observation : estimation of the weight vector according to the image $\{(\mathbf{x}_k^n, \boldsymbol{\pi}_k^n)\}_{n=1}^N$ with $\boldsymbol{\pi}_k^n \propto \mathbf{p}(\mathbf{z}_k|\mathbf{x}_k = \mathbf{x}_k^n)$

Sampling : build $\{(\mathbf{x}_{k-1}'^n, \mathbf{1}/N)\}_{n=1}^N$ from $\{(\mathbf{x}_0^n, \boldsymbol{\pi}_0^n)\}_{n=1}^N$ using Importance Sampling

Estimation : $\hat{\mathbf{x}}_k \doteq \frac{1}{N}\sum_{n=1}^N \mathbf{x}_k^n$

end for

Output: Estimated state sequence $\{\hat{\mathbf{x}}_k\}_{k=1,\ldots,K_{end}}$

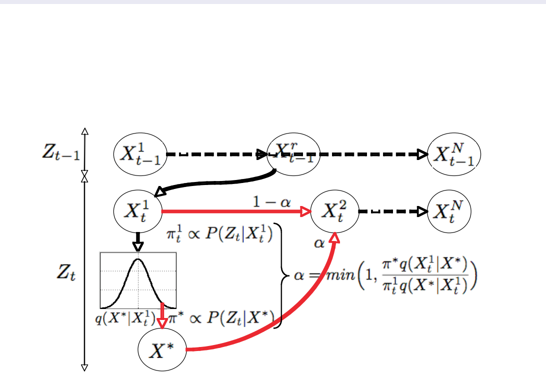
SIR: Conclusion

Conclusion

- Particle filters approximate non gaussian pdf
- CONDENSATION is a "parallel" algorithm.
- The power of exploration is conditioned by an efficient sampling (many sampling strategies have been proposed)
- matlab illustration (SIR, CONDENSATION)

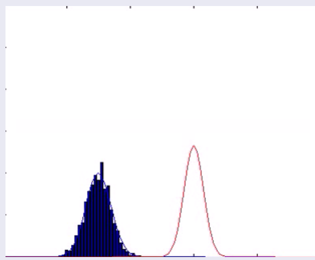
MCMC (Markov Chain Monte Carlo)

Examples



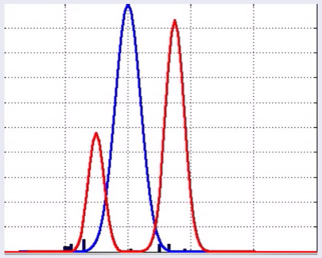
MCMC (Markov Chain Monte Carlo)

Method: build a chained set of particles (Markov Chain Monte Carlo)



MCMC (Markov Chain Monte Carlo)

Method: build a chained set of particles (Markov Chain Monte Carlo)



MCMC: Conclusion

Conclusion

- MCMC approximate non gaussian pdf
- MCMC are sequential algorithms.
- Efficient sampling strategies based on partionned sampling can be proposed
- MCMC are used in for high dimensional tracking problems

Multi-object visual tracking MOT

Challenges

- The state vector has a variable dimension:
- The exploration process must jump from one dimension to an other.

One solution: RJMCMC

- Reversible Jump Monte-Carlo Markov Chain is a solution to track a varying number of objects

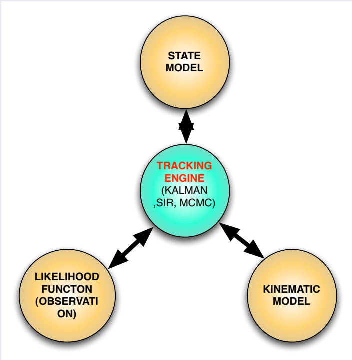
RJMCMC

RJMCMC

- The size of the state vector is variable according to the pdf associated to the number of objects
- position updating proposals,
- dimension move proposals (add an object, remove an object)

Conclusion

Conclusion

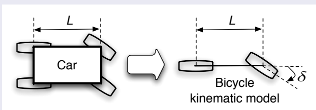


Content

What do we want to do ?

Visual tracking of a vehicle from a static camera

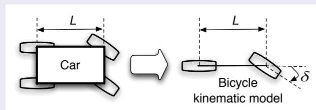
- The dynamic model of the object is known.
- We want to estimate velocity and steering angle of the vehicle.



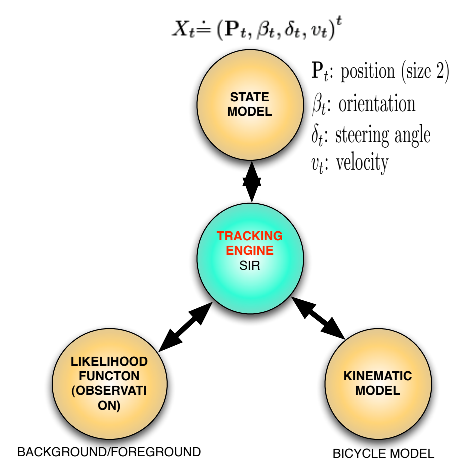
What do we want to do ?

Visual tracking of a vehicle from a static camera

- The dynamic model of the object is known.
- We want to estimate velocity and steering angle of the vehicle.

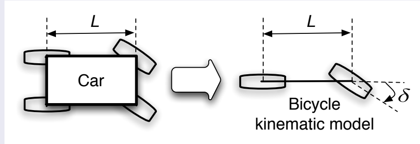


Tracking scheme



Kinematic model

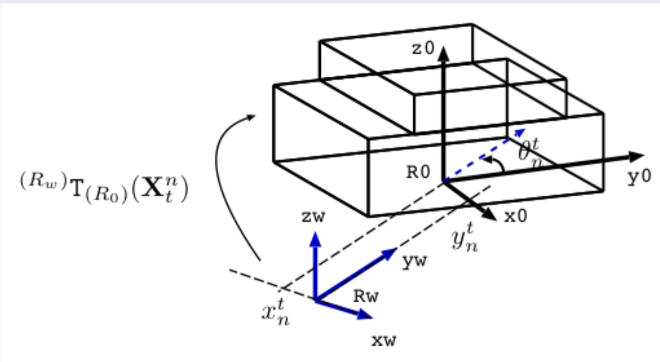
Bicycle model



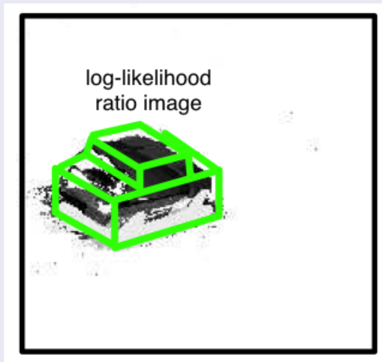
$$\begin{aligned} \dot{x} &= v.\cos\beta \\ \dot{y} &= v.\sin\beta \\ \dot{\beta} &= \frac{v}{L}.\tan\delta \end{aligned} \tag{1}$$

- v: velocity
- x, y : position
- β: orientation

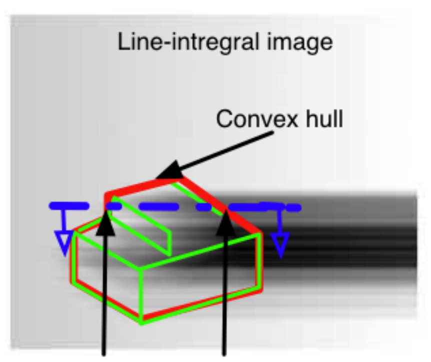
Observation function



Observation function

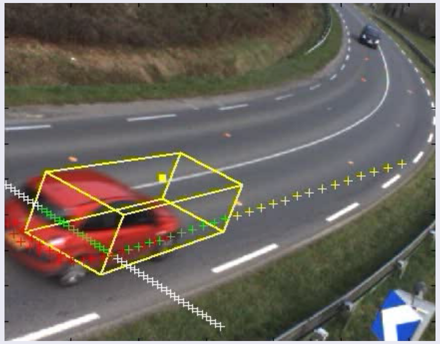


Observation function



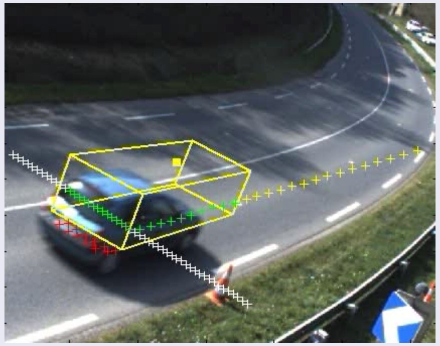
Results

Results



Results

Results



Précision (cm)

Speed km/hr	Vision ave/std	Rangefinder ave/std	Sensor merge ave/std
40	**0.25**/0.18	**0.65**/0.54	**0.17**/0.10
60	**0.19**/0.16	**0.72**/0.67	**0.09**/0.06
80	**0.18**/0.15	**0.33**/0.22	**0.14**/0.10

Multi-object tracking

Multi-object tracking

Introduction to Visual Tracking
Modelling Visual Tracking
Stochastic Filters for Visual Tracking
Some examples
Tracking a vehicle from a static camera
Multi-object tracking
Tracking with classifiers

What do we want to do ?

Visual tracking of a varying number of objects

- real time tracking and identification of a variable number of objects in 3D,
- no prior knowledge of objects appearance,
- robustness to partial and heavy occlusions ,
- robustness to heavy scale changes,
- no prior knowledge of object entrance locations.

T. Chateau Blaise Pascal University

Introduction to Visual Tracking
Modelling Visual Tracking
Stochastic Filters for Visual Tracking
Some examples
Tracking a vehicle from a static camera
Multi-object tracking
Tracking with classifiers

What do we want to do ?

Visual tracking of a varying number of objects

- real time tracking and identification of a variable number of objects in 3D,
- no prior knowledge of objects appearance,
- robustness to partial and heavy occlusions ,
- robustness to heavy scale changes,
- no prior knowledge of object entrance locations.

T. Chateau Blaise Pascal University

Introduction to Visual Tracking
Modelling Visual Tracking
Stochastic Filters for Visual Tracking
Some examples
Tracking a vehicle from a static camera
Multi-object tracking
Tracking with classifiers

What do we want to do ?

Visual tracking of a varying number of objects

- real time tracking and identification of a variable number of objects in 3D,
- no prior knowledge of objects appearance,
- robustness to partial and heavy occlusions ,
- robustness to heavy scale changes,
- no prior knowledge of object entrance locations.

T. Chateau Blaise Pascal University

What do we want to do ?

Visual tracking of a varying number of objects

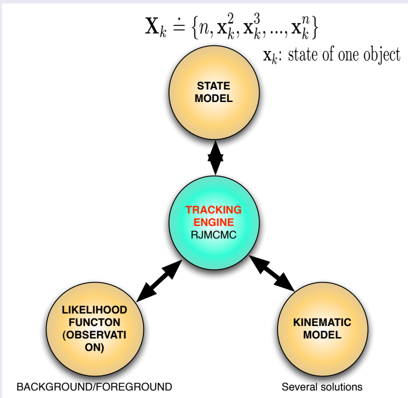
- real time tracking and identification of a variable number of objects in 3D,
- no prior knowledge of objects appearance,
- robustness to partial and heavy occlusions ,
- robustness to heavy scale changes,
- no prior knowledge of object entrance locations.

What do we want to do ?

Visual tracking of a varying number of objects

- real time tracking and identification of a variable number of objects in 3D,
- no prior knowledge of objects appearance,
- robustness to partial and heavy occlusions ,
- robustness to heavy scale changes,
- no prior knowledge of object entrance locations.

Tracking scheme



Proposals

RJMCMC Proposals

- **Enter:** add an object (data driven)
- **Leave:** remove an object (data driven)
- **Object position Update:** choose and object and propose a spatial move associated to this object.

Results

Demonstration

Introduction to Visual Tracking
Modelling Visual Tracking
Stochastic Filters for Visual Tracking
Some examples
Tracking a vehicle from a static camera
Multi-object tracking
Tracking with classifiers

Tracking with classifiers

Tracking with classifiers

T. Chateau Blaise Pascal University

Introduction to Visual Tracking
Modelling Visual Tracking
Stochastic Filters for Visual Tracking
Some examples
Tracking a vehicle from a static camera
Multi-object tracking
Tracking with classifiers

What do we want to do ?

Visual tracking of an object class from a moving camera

- We want to recognize and track a moving object from a moving monocular camera, at realtime (30fps).
- We want to track and recognize an object using only a generic model.

T. Chateau Blaise Pascal University

Introduction to Visual Tracking
Modelling Visual Tracking
Stochastic Filters for Visual Tracking
Some examples
Tracking a vehicle from a static camera
Multi-object tracking
Tracking with classifiers

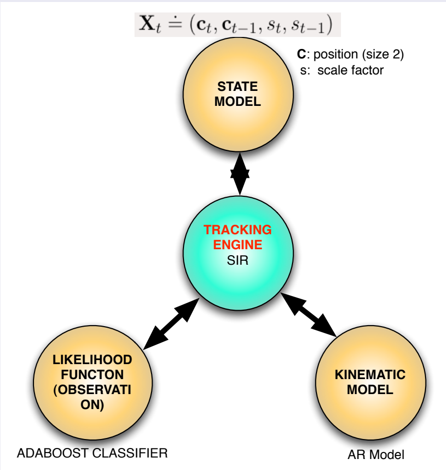
What do we want to do ?

Visual tracking of an object class from a moving camera

- We want to recognize and track a moving object from a moving monocular camera, at realtime (30fps).
- We want to track and recognize an object using only a generic model.

T. Chateau Blaise Pascal University

Tracking scheme



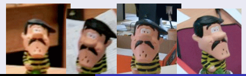
Why is it a challenge ?

A moving object

- Variation of the appearence of the object (3D object)

- Variation of the illumination conditions

- Cluttered background



Why is it a challenge ?

A moving camera

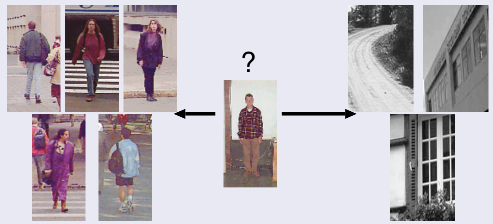
- Background/Foreground segmentation can not be done easily.

Difficult with a moving camera

Why is it a challenge ?

Recognize an object from a generic model

This is an object recognition problem



Why is it a challenge ?

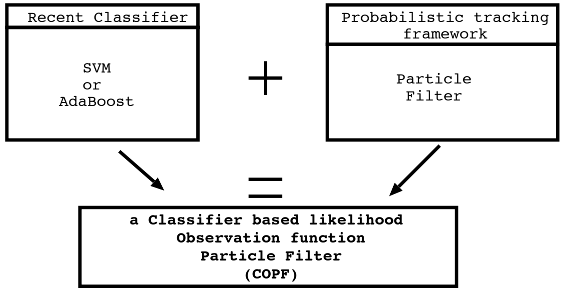
Track an object at a realtime framerate

This is a realtime tracking problem

A solution ?

using classifiers into a probabilistic tracking framework

Bring together recent classifiers (Adaboost, SVM) and a particle filter



Using Classifiers for Real-Time Tracking: why ?

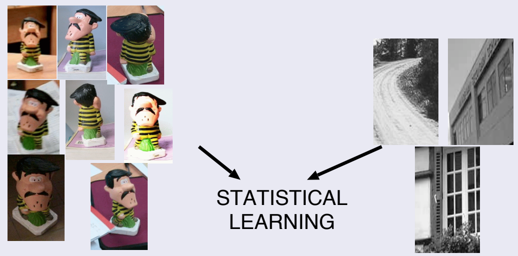
Offline learning of the object to be tracked

- Trackers can be designed to track categories of objects (pedestrians, vehicles),

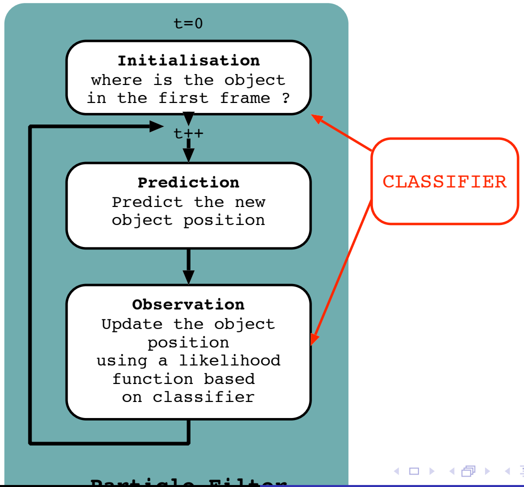
Using Classifiers for Real-Time Tracking: why ?

Offline learning of the object to be tracked

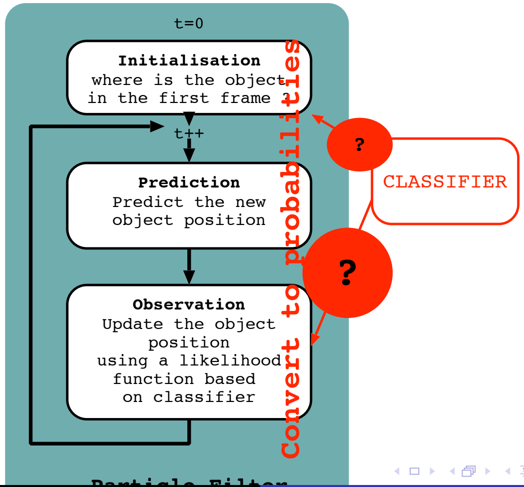
- Objects are modelized with a collection of views, representing variation of the object appearance,



Overview of the method



Overview of the method



The Observation Function

output of a classifier $m(\mathbf{f})$

Assumption : $m(\mathbf{f}) \in]-\infty; +\infty[$ with

$$m(\mathbf{f}_1) < m(\mathbf{f}_2) \longrightarrow P(class|\mathbf{f}_1) < P(class|\mathbf{f}_2)$$

***Platt scaling*, 1999**

Estimate a sigmoid, from a learning database in order to produce calibrated probabilities from the output of the classifier :

$$P(\text{positive}|m(\mathbf{f})) = \frac{1}{1 + \exp(A.m(\mathbf{f}) + B)} \qquad (2)$$

The Observation Function

output of a classifier $m(\mathbf{f})$

Assumption : $m(\mathbf{f}) \in]-\infty; +\infty[$ with

$$m(\mathbf{f}_1) < m(\mathbf{f}_2) \longrightarrow P(class|\mathbf{f}_1) < P(class|\mathbf{f}_2)$$

***Platt scaling*, 1999**

Estimate a sigmoid, from a learning database in order to produce calibrated probabilities from the output of the classifier :

$$P(\text{positive}|m(\mathbf{f})) = \frac{1}{1 + \exp(A.m(\mathbf{f}) + B)} \qquad (2)$$

Platt scaling

Estimation of the sigmoid parameters

A et B are two parameters to be estimated from a learning database (m_i, y_i) with $(y_i \in \{0;1\})$

non-linear estimation

minimization of the cross-entropy error function:

$$argmin_{(A,B)}\{-\sum_i y_i \log(p_i) + (1-y_i)\log(1-p_i)\}, \quad (3)$$

with

$$p_i = \frac{1}{1+\exp(A.m_i + B)} \quad (4)$$

Platt scaling

Estimation of the sigmoid parameters

A et B are two parameters to be estimated from a learning database (m_i, y_i) with $(y_i \in \{0;1\})$

non-linear estimation

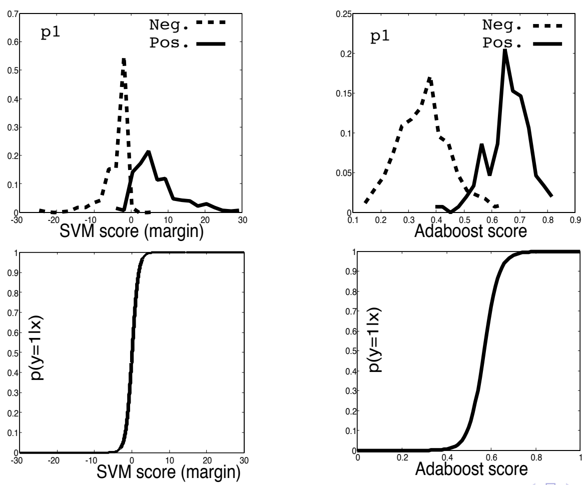
minimization of the cross-entropy error function:

$$argmin_{(A,B)}\{-\sum_i y_i \log(p_i) + (1-y_i)\log(1-p_i)\}, \quad (3)$$

with

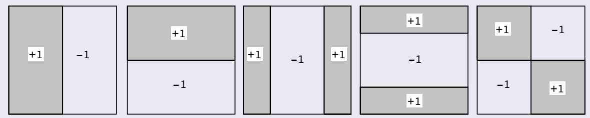
$$p_i = \frac{1}{1+\exp(A.m_i + B)} \quad (4)$$

Results



Features

Haar based wavelets



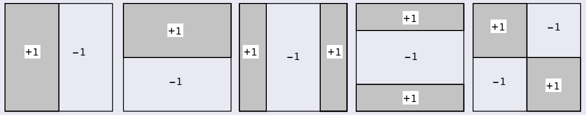
A Window of interest is descibed by a vector of the output of five filters applied to several positions and scales.

Large number of features

example: For 3 scales and a 128×64 pixels image, the number of features is about 40000.

Features

Haar based wavelets



A Window of interest is descibed by a vector of the output of five filters applied to several positions and scales.

Large number of features

example: For 3 scales and a 128×64 pixels image, the number of features is about 40000.

features reduction

Using Adaboost

- **Weak Classifier**: one threshold for each feature
- **Evaluation Fonction**: minimization of the number of samples assigned to the bad class
- **Stopping criteria**: number of features to retain (rounds of boosting)

features reduction

Using Adaboost

- **Weak Classifier**: one threshold for each feature
- **Evaluation Fonction**: minimization of the number of samples assigned to the bad class
- **Stopping criteria**: number of features to retain (rounds of boosting)

features reduction

Using Adaboost

- **Weak Classifier**: one threshold for each feature
- **Evaluation Fonction**: minimization of the number of samples assigned to the bad class
- **Stopping criteria**: number of features to retain (rounds of boosting)

features reduction

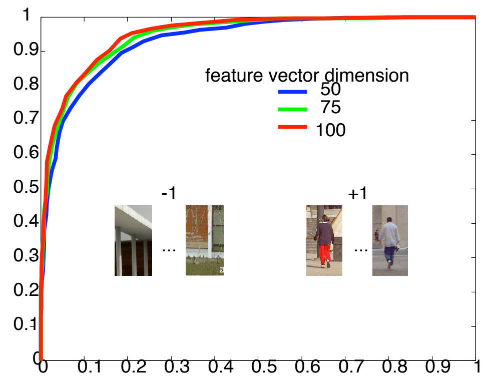
Using Adaboost

- **Weak Classifier**: one threshold for each feature
- **Evaluation Fonction**: minimization of the number of samples assigned to the bad class
- **Stopping criteria**: number of features to retain (rounds of boosting)

Some of the selected features

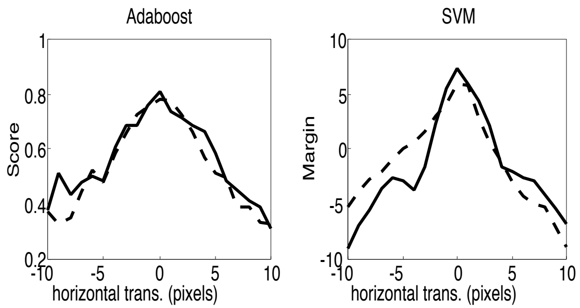
Result for a recognition step (Adaboost)

Result for a recognition step (Adaboost)



Roc curve for three different sizes of feature vector
Using a pedestrian database with 2000 samples for training and 2000 samples for testing

Classifier score evolution near the true position of the object



Particle filter : $p(\mathbf{X}_t|\mathbf{Z}_{0:t}) \doteq \{(\mathbf{X}_t^n, \pi_t^n)\}_{n=1}^N$

State vector

$$\mathbf{X}_t \doteq (\mathbf{c}_t, \mathbf{c}_{t-1}, s_t, s_{t-1}) \quad (5)$$

with $\mathbf{c} \doteq (x, y)$ location of a bounding box whitin the image and associated scale factor s_t

Dynamics

$$\mathbf{X}_{t+1} = \mathtt{A}\mathbf{X}_t + \mathtt{B}\mathbf{v}_t \ , \ \ \mathbf{v}_t \sim \mathcal{N}(0, \Sigma) \quad (6)$$

Likelihood function

$$P(\mathbf{Z}_t|\mathbf{X}_t = \mathbf{X}_t^n) = \frac{1}{1 + \exp\left\{\hat{A}.m(\mathbf{F}^*(\mathbf{c}_t^n + s_t^n\mathtt{W})) + \hat{B}\right\}} \quad (7)$$

Particle filter : $p(\mathbf{X}_t|\mathbf{Z}_{0:t}) \doteq \{(\mathbf{X}_t^n, \pi_t^n)\}_{n=1}^N$

State vector

$$\mathbf{X}_t \doteq (\mathbf{c}_t, \mathbf{c}_{t-1}, s_t, s_{t-1}) \quad (5)$$

with $\mathbf{c} \doteq (x, y)$ location of a bounding box whitin the image and associated scale factor s_t

Dynamics

$$\mathbf{X}_{t+1} = \mathtt{A}\mathbf{X}_t + \mathtt{B}\mathbf{v}_t \ , \ \ \mathbf{v}_t \sim \mathcal{N}(0, \Sigma) \quad (6)$$

Likelihood function

$$P(\mathbf{Z}_t|\mathbf{X}_t = \mathbf{X}_t^n) = \frac{1}{1 + \exp\left\{\hat{A}.m(\mathbf{F}^*(\mathbf{c}_t^n + s_t^n\mathtt{W})) + \hat{B}\right\}} \quad (7)$$

Particle filter : $p(\mathbf{X}_t|\mathbf{Z}_{0:t}) \doteq \{(\mathbf{X}_t^n, \pi_t^n)\}_{n=1}^N$

State vector

$$\mathbf{X}_t \doteq (\mathbf{c}_t, \mathbf{c}_{t-1}, s_t, s_{t-1}) \quad (5)$$

with $\mathbf{c} \doteq (x, y)$ location of a bounding box whitin the image and associated scale factor s_t

Dynamics

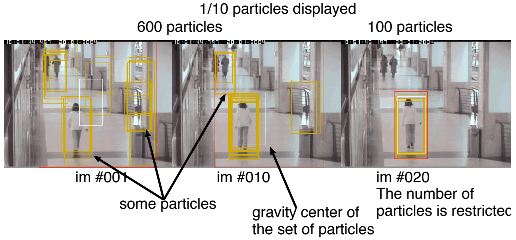
$$\mathbf{X}_{t+1} = \mathtt{A}\mathbf{X}_t + \mathtt{B}\mathbf{v}_t \ , \ \ \mathbf{v}_t \sim \mathcal{N}(0, \Sigma) \quad (6)$$

Likelihood function

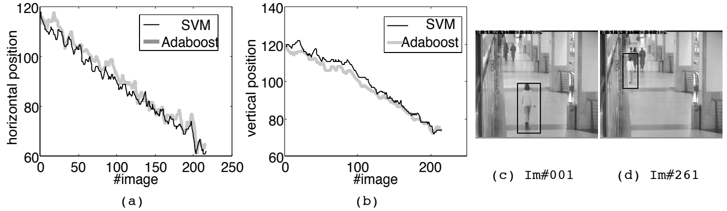
$$P(\mathbf{Z}_t|\mathbf{X}_t = \mathbf{X}_t^n) = \frac{1}{1 + \exp\left\{\hat{A}.m(\mathbf{F}^*(\mathbf{c}_t^n + s_t^n\mathtt{W})) + \hat{B}\right\}} \quad (7)$$

Initialization

1. The classifier is called for a set of regular positions and scales whitin the image
2. A set of particles is initialized from the positions ans scales associated to the highest outputs of the classifier
3. The number of particles is restricted when the particle distribution is around the object to be tracked.

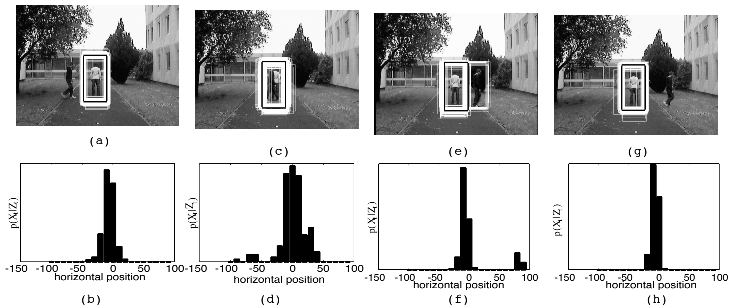


Comparing SVM and Adaboost classifiers



Tracking a generic object (videos)

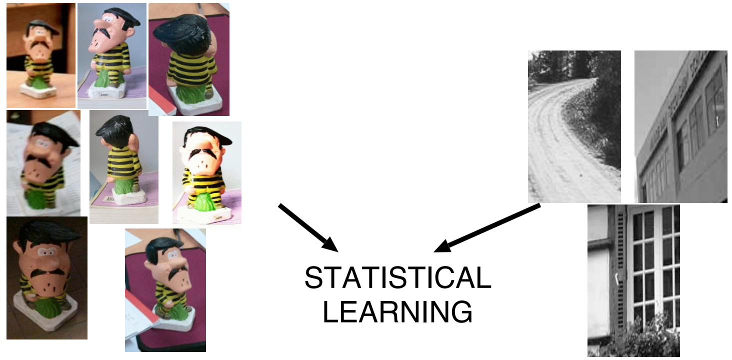
Tracking a generic object: occlusion (videos)



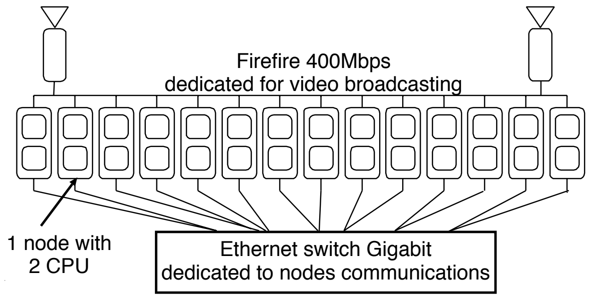
Tracking a generic object: occlusion (videos)

(a) (36.2s) (b) (37.0s) (c) (37.5s) (d) (38.5s)

Tracking a specific object (videos)



Parallel implementation: the Babylon Project

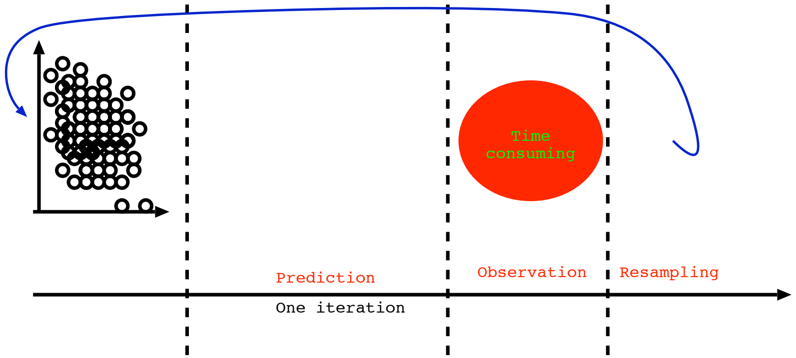


Node: XServe PPC G5

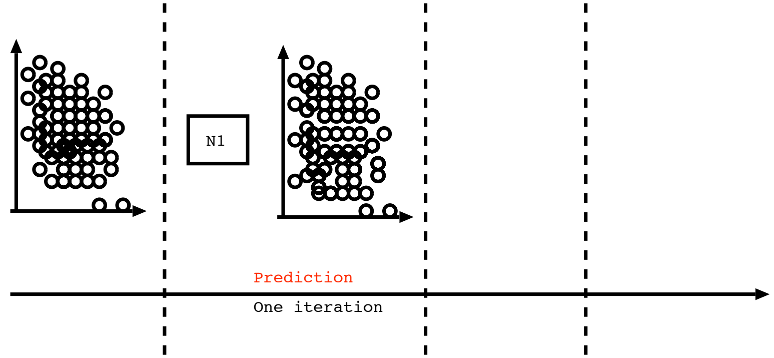
- Bi-processor
- SIMD achived by the AltiVec Extention

The Babylon Project

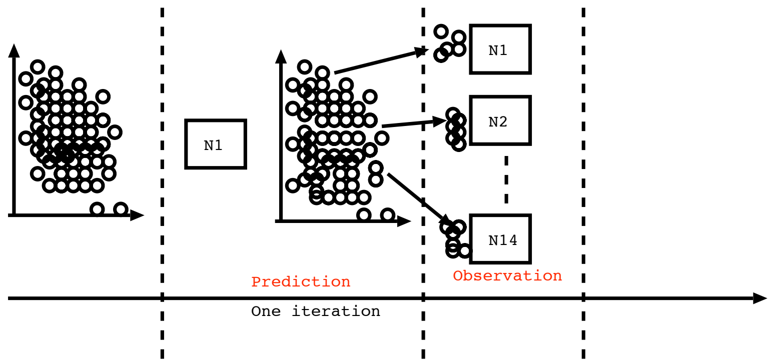
Parallel implementation



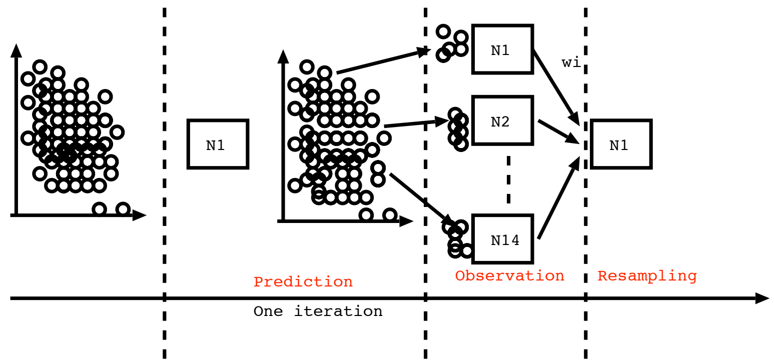
Parallel implementation



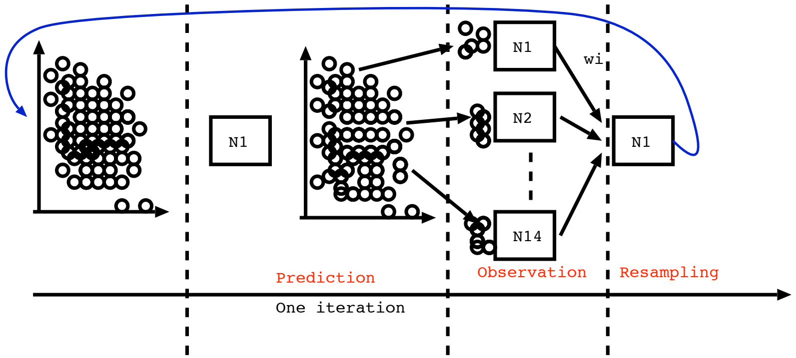
Parallel implementation



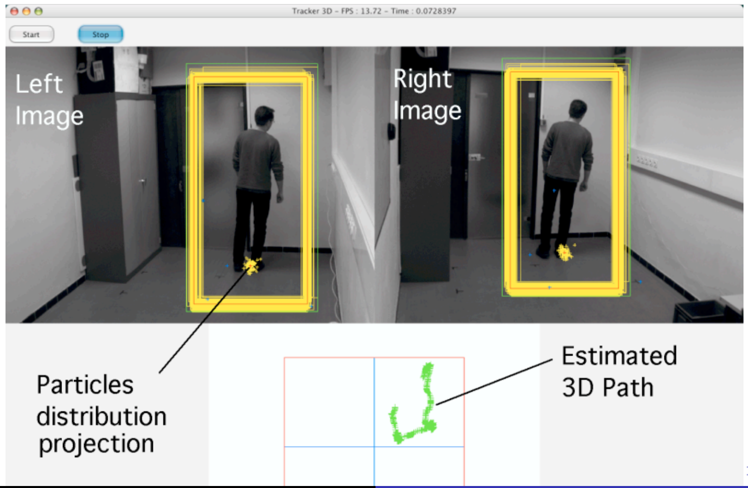
Parallel implementation



Parallel implementation



3D Tracking



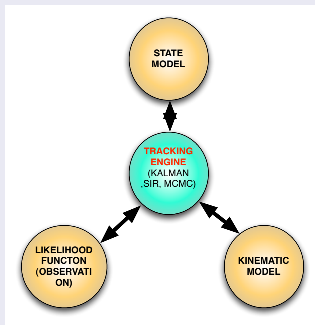
3D Tracking

Performances

	200	500	1000	5000	10000
Seq.	$0.0609s$	$0.1439s$	$0.2874s$	$1.6393s$	$3.8462s$
	16.40	6.95	3.48	0.61	0.26
Par.	$0.0231s$	$0.0265s$	$0.0313s$	$0.0858s$	$0.1567s$
	43.31	37.72	31.9	11.66	6.38
Gain	$\times 2.7$	$\times 5.42$	$\times 9.16$	$\times 19.43$	$\times 24.5$

- 20 FPS with 2000 particles .
- Linear evolution of performances according to the number of nodes.

Conclusion



Thanks to F. Bardet, D. Ramadasan, C. Tournayre, G. Jacob, E. Royer,....