Image motion

Jorge Marques, 2008

finding a template

Suppose we wish to find a known template T(x,y) in a given image I(x,y).

This problem is known as template matching.

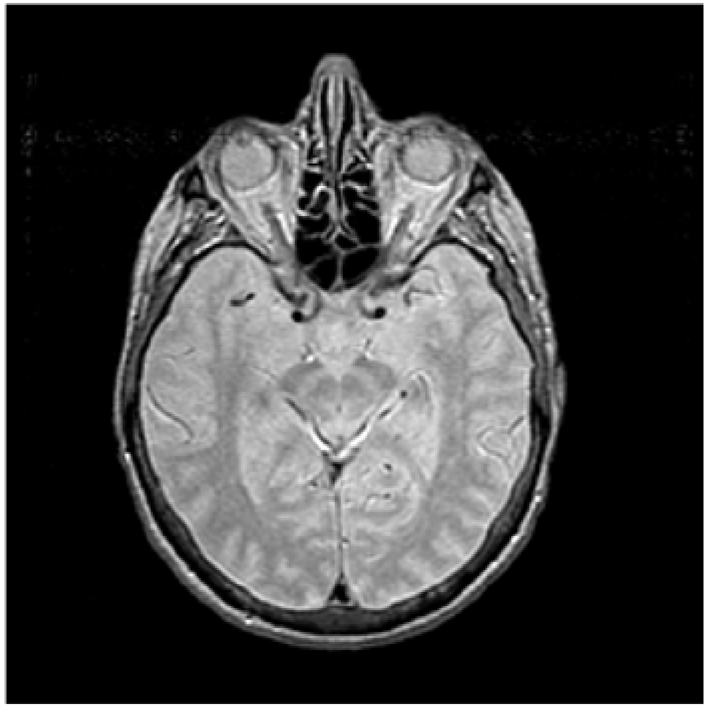
template

FC Barcelona

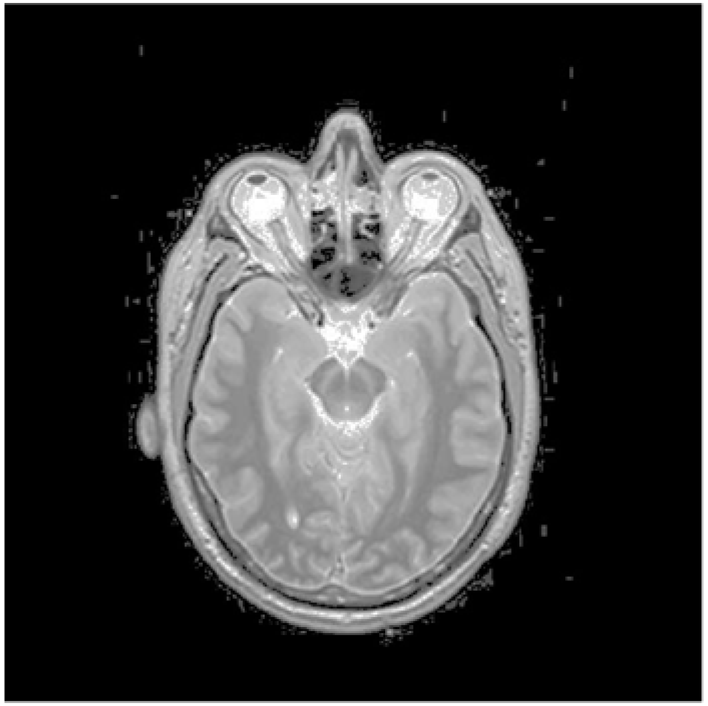
image

the template can be small or large

alignment (MRI images)



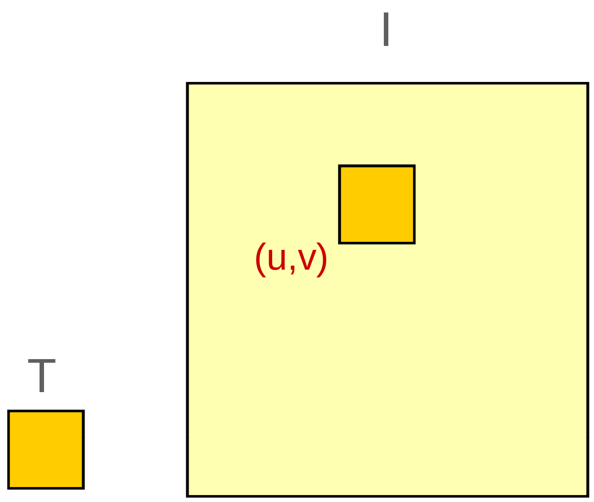
atlas



test slice

Adapted from Kubic, Unser, 2003

template matching



template $T(x,y) \quad x,y = 0, \ldots, B-1$

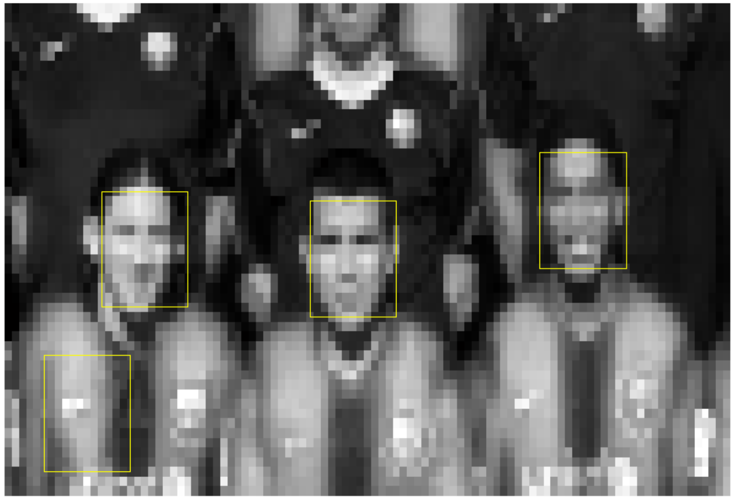
image $I(x,y) \quad x,y = 0, \ldots, N-1$

displacement $t = (u,v)$

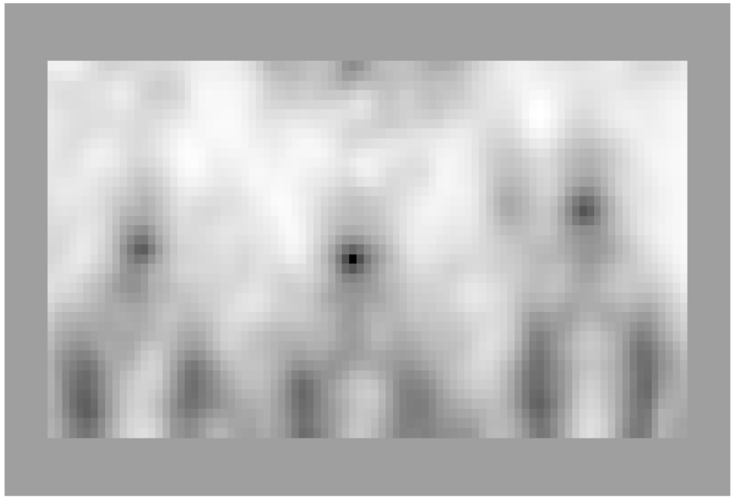
The solution is based on two steps:

- define a matching criterion M (e.g., cross correlation)
- find local maxima/minima (e.g., exhaustive search)

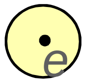
object detection



matching criterion: M



nonlinear optimization



Non-minimum suppression:

$$M(t_0) < M(t) \quad \text{for all t in a vicinity of radius r of d}$$

Thresholding: $M(t_0) < \lambda$ λ threshold

matching criteria

cross-correlation

$$R(u,v) = \sum_{x,y=0}^{B-1} T(x,y) I(x+u, y+v)$$

sum of square differences (SSD)
(l_2 norm, squared)

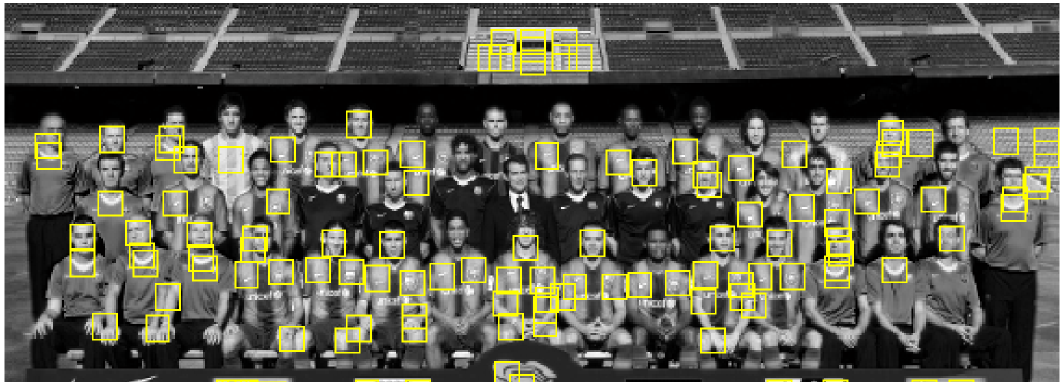
$$E(u,v) = \sum_{x,y=0}^{B-1} [T(x,y) - I(x+u, y+v)]^2$$

sum of absolute differences (SAD)
(l_1 norm)

$$E(u,v) = \sum_{x,y=0}^{B-1} |T(x,y) - I(x+u, y+v)|$$

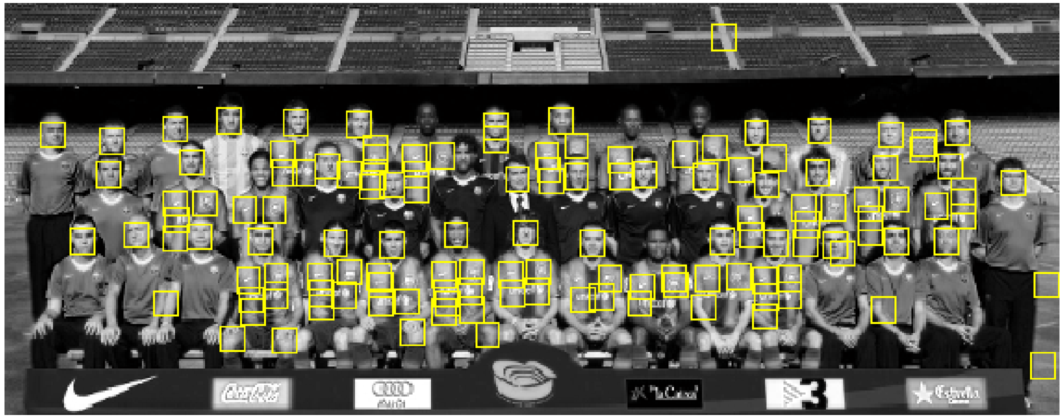
Non integer displacements can be considered. Image interpolation is required in this case.

cross-correlation



SSD

SAD



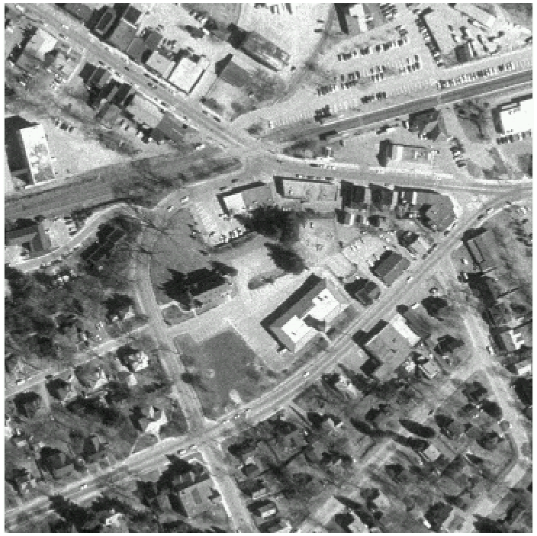
There are better ways to detect faces (e.g., Viola & Jones) !!

limitations

Template matching has weaknesses:

- not invariant to rotations and scaling → more general transformations
- not invariant to illumination changes → modify matching criteria to improve robustness
- time consuming
- template adaptation is tricky

problem formulation



Matlab

Image alignment

Given 2 (or more) images I, T we wish to estimate a transformation which maps the first into the second

$$(x, y) \rightarrow (x', y') \qquad (x', y') = W(x, y; \theta)$$

according to some criterion.

This can be done using:

feature based methods:

based on the alignment of feature points (marks)

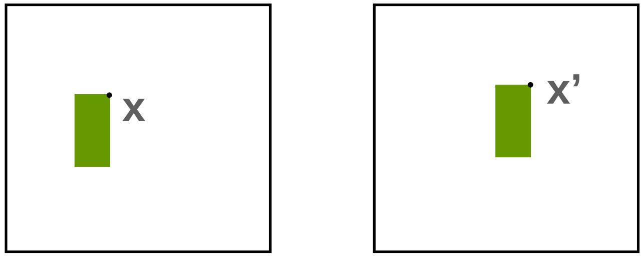
image based methods:

based on the alignment of image intensity or color

What geometric transformations can we use?

translation & rigid body

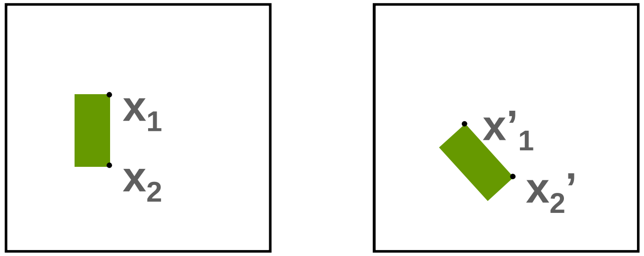
translation



$$W(x;\theta) = x + t \qquad \theta = t$$

2 degrees of freedom

rigid body



$$W(x;\theta) = Rx + t \qquad \theta = (R, t)$$

3 degrees of freedom

rotation matrix

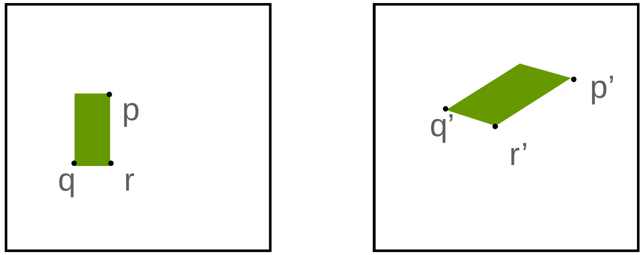
$$RR^T = R^T R = I$$

$$\det(R) = 1$$

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

afinne and projective transformations

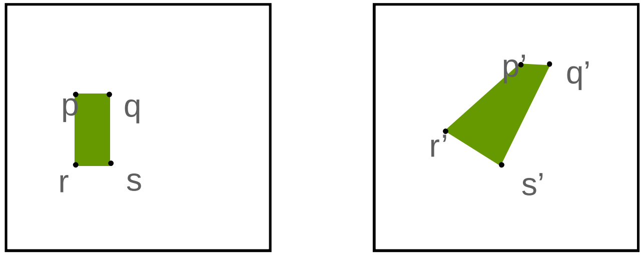
affine transformation



$$W(x;\theta) = Ax + t \qquad \theta = (A, t)$$

6 degrees of freedom

projective transformation (homography)



$$W(x,\theta) = \begin{bmatrix} \frac{p_1 x + p_2 y + p_3}{p_7 x + p_8 y + p_9} \\ \frac{p_4 x + p_5 y + p_6}{p_7 x + p_8 y + p_9} \end{bmatrix} \qquad \theta = (p_1, \ldots, p_9)$$

8 degrees of freedom

projective and polynomial transformations

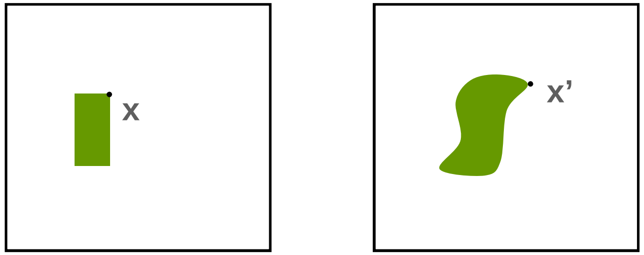
projective (contd.)

$$x' = \frac{\tilde{x}^T p_1}{\tilde{x}^T p_3} \qquad y' = \frac{\tilde{x}^T p_2}{\tilde{x}^T p_3}$$

$$p_1 = [p_1\ p_2\ p_3]^T \qquad p_2 = [p_4\ p_5\ p_6]^T$$

$$p_3 = [p_7\ p_8\ p_9]^T \qquad \tilde{x} = [x\ y\ 1]^T$$

polynomial



$$W(x,\theta) = \begin{bmatrix} \sum_{p,q:p+q\le n} a_{pq} x^p y^q \\ \sum_{p,q:p+q\le n} b_{pq} x^p y^q \end{bmatrix}$$

The estimation of coefficients is numerically ill conditioned

others …. e.g., free form deformations

properties

	DoF	Preserves lines?	Preserves Paralelism?	Preserves Angles?	Preserves length?
translation	2	Yes	Yes	Yes	Yes
Rigid body	3	Yes	Yes	Yes	Yes
Affine	6	Yes	Yes	X	X
Projective	8	Yes	X	X	X
Polynomial	(n+2)(n+1)/2	X	X	X	X

can we align images using intensity?

image based methods

Problem:

Given two images T, I we wish to find a geometric transformation **W(x)** which maps points of the first image into points of the second, such that

I(**W**(**x**))≈T(**x**).

color constancy

Most popular criterion (SSD)

$$E(\theta) = \sum_{\mathrm{x}} [T(\mathrm{x}) - I(W(\mathrm{x};\theta))]^2$$

Note: the sum is for all the points x in which both images T(x), I(W(x)) overlap.

The minimization of E is a non linear problem!!

Lucas-Kanade (translation motion)

Criterion $$E(u,v) = \sum_{\mathrm{x}} [T(\mathrm{x}) - I(\mathrm{x}+\mathrm{t})]^2$$

Parameter update $$\mathrm{t} = \mathrm{t}_0 + \Delta t$$

First order approximation of the image $$I(\mathrm{x}+\mathrm{t}) = I(\mathrm{x}+\mathrm{t}_0) + \nabla I(\mathrm{x}+\mathrm{t}_0)^T \Delta \mathrm{t}$$

Lucas Kanade algorithm (recursion)

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & I_y^2 \end{bmatrix} \Delta \mathrm{t} = \begin{bmatrix} \sum (T(\mathrm{x}) - I(\mathrm{x}+\mathrm{t}_0)) I_x \\ \sum (T(\mathrm{x}) - I(\mathrm{x}+\mathrm{t}_0)) I_y \end{bmatrix}$$

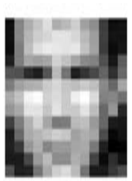
$$\mathrm{t} \leftarrow \mathrm{t}_0 + \Delta t$$

$R\Delta t = r$

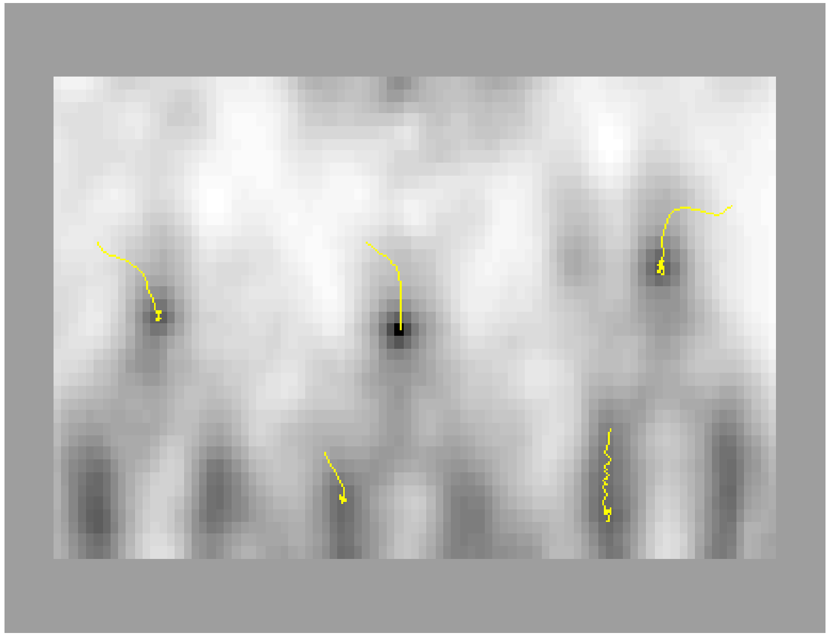
$$\mathrm{t} \leftarrow \mathrm{t}_0 + \Delta t$$

I_x, I_y are the partial derivatives of I at $x+t_0$.

convergence from several starting points



SSD criterion



The SSD criterion is not explicitly computed in the L-K algorithm.

proof

Let us minimize
$$E = \sum_x [T(\mathrm{x}) - I(\mathrm{x}+\mathrm{t}_0) - \nabla I(\mathrm{x}+\mathrm{t}_0)^T \Delta t]^2$$

A necessary condition is

$$\frac{dE}{d\Delta t} = 0 \qquad \sum_x [T(\mathrm{x}) - I(\mathrm{x}+\mathrm{t}_0) - \nabla I(\mathrm{x}+\mathrm{t}_0)^T \Delta \mathrm{t}]\nabla I(\mathrm{x}+\mathrm{t}_0) = 0$$

$$\sum_x \nabla I(\mathrm{x}+\mathrm{t}_0)\nabla I(\mathrm{x}+\mathrm{t}_0)^T \Delta \mathrm{t} = \sum_x [T(\mathrm{x}) - I(\mathrm{x}+\mathrm{t}_0)]\nabla I(\mathrm{x}+\mathrm{t}_0)$$

Defining
$$\nabla I(\mathrm{x}+\mathrm{t}_0) = \begin{bmatrix} I_x(\mathrm{x}+\mathrm{t}_0) \\ I_y(\mathrm{x}+\mathrm{t}_0) \end{bmatrix}$$

We obtain
$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_y I_x & \sum I_y^2 \end{bmatrix} \Delta \mathrm{t} = \begin{bmatrix} \sum (T(\mathrm{x}) - I(\mathrm{x}+\mathrm{t}_0)) I_x \\ \sum (T(\mathrm{x}) - I(\mathrm{x}+\mathrm{t}_0)) I_y \end{bmatrix}$$

discussion

L-K strong points

- uses all the available information
- It is simple
- appropriate for tracking
- can be extended to deal with general motion models

L-K weak points

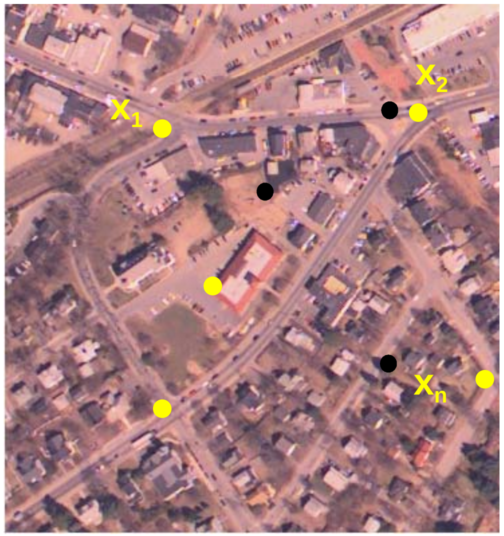
- no guarantee that the optimal solution is obtained
- the solution depends on the initialization ⟶ use multiple scales
- convergence is difficult if the number of parameters is high
- solution depends on the illumination ⟶ Illumination can be estimated

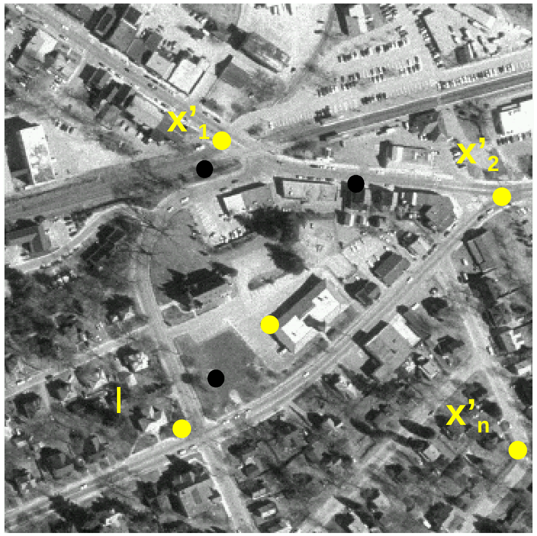
can we align images from sparse prototypes?

feature based matching

Problem:

Given two sets of points $\{\mathbf{x}_i\}$, $\{\mathbf{x}'_j\}$ detected in the images T, I, we wish to find a geometric transformation W that maps the points $\{x_i\}$ into the points $\{x'_j\}$.





$$\mathrm{x}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}^T \qquad \mathrm{x}'_i = \begin{bmatrix} x_i' \\ y_i' \end{bmatrix}$$

we assume that the correspondence is known $\mathrm{x}_i \leftrightarrow \mathrm{x}'_i$

approach

Define a matching criterion e.g.,

$$E(\theta) = \sum_i \| x'_i - W(x_i;\theta) \|^2$$

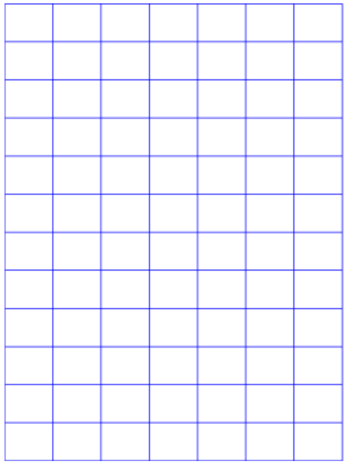
SSD criterion

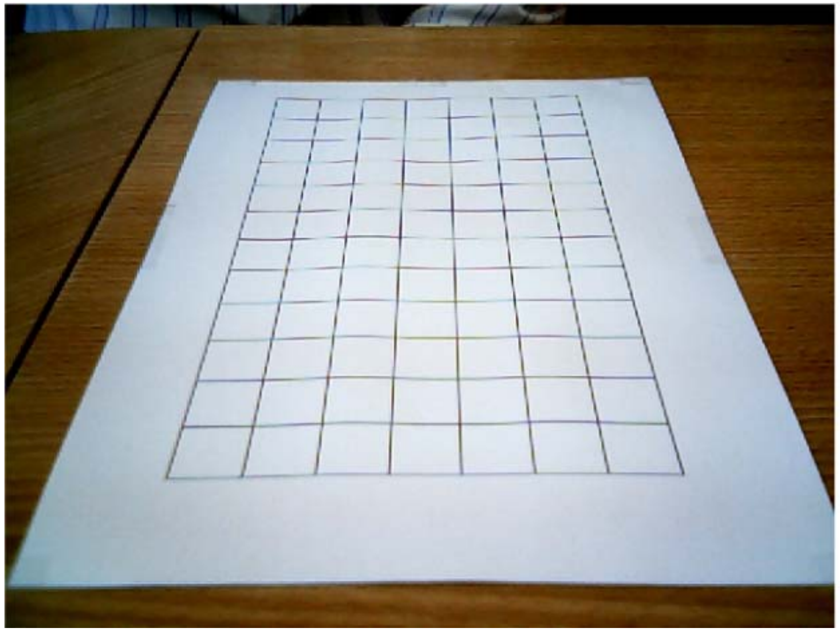
Minimize the criterion with respect to θ using a closed form or a numeric algorithm.

Note: there are other matching e.g., l_1 norm.

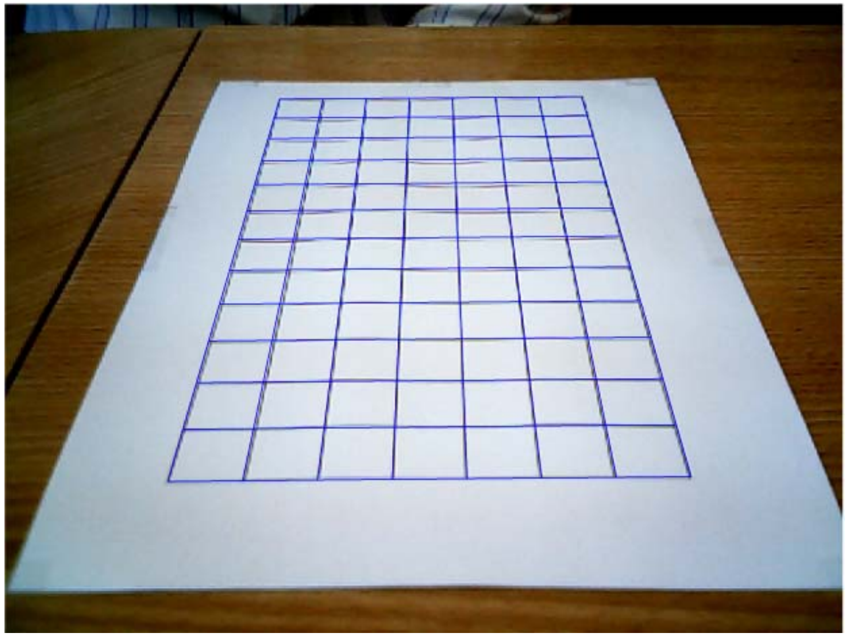
example

input





output



alignment using a projective transform

estimation of an homography

Homography

$$x' = f(x, p)$$

$$x' = \frac{p_1 x + p_2 y + p_3}{p_7 x + p_8 y + p_9}$$

$$y' = \frac{p_4 x + p_5 y + p_6}{p_7 x + p_8 y + p_9}$$

$$\| p \| = 1$$

is a nonlinear function of the unknown parameters.

The minimization of the SSD criterion is difficult !!

$$E(p) = \sum_i \| x'_i - f(x_i, p) \|^2$$

Idea: use another (simpler) criterion instead

$$(p_7 x + p_8 y + p_9) x' = (p_1 x + p_2 y + p_3)$$

$$(p_7 x + p_8 y + p_9) y' = (p_4 x + p_5 y + p_6)$$

algebraic error

$$e = \begin{bmatrix} (p_1 x + p_2 y + p_3) - (p_7 x + p_8 y + p_9) x' \\ (p_4 x + p_5 y + p_6) - (p_7 x + p_8 y + p_9) y' \end{bmatrix}$$

$$E'(p) = \sum_i \| e_i \|^2 \qquad \| p \| = 1$$

estimation of the projective transform (2)

minimize

$$E' = p^T M^T M p$$

with restriction $p^T p = 1$

$$M = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x_1 \\ \vdots & \vdots & \vdots & & & & \vdots & \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x_n \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 & -y_1 \\ \vdots & \vdots & \vdots & \vdots & & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_n x_n & -y'_n y_n & -y_n \end{bmatrix}$$

This problem can be easily solved using Lagrange multipliers:

p is the eigenvector of matrix $M^T M$ associated to the smallest eigenvalue.

The whole algorithm can be written in 1 (long) line of Matlab!

proof

Lagrangian function

$$L = E' - \lambda(p^T p - 1) = p^T M^T M p - \lambda(p^T p - 1)$$

$$\frac{dL}{dp} = 0 \quad \Rightarrow \quad M^T M p - \lambda p = 0$$

$$M^T M p = \lambda p$$

p is na eigen vector of matrix $M^T M$

which one? $E = p^T M^T M p = \lambda p^T p = \lambda$

choose λ_{min}

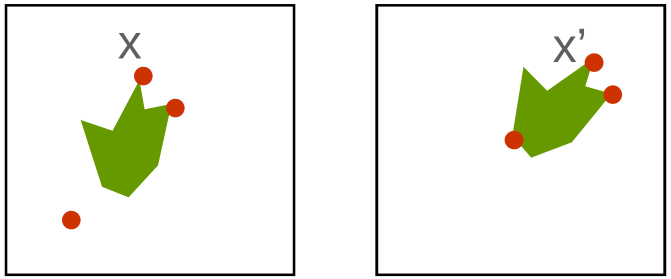
other transformations?

The other transformations (translation, affine, polynomial) are easily estimated by the minimization of the SSD criterion E.

Only the rigid body transformation is a bit more difficult because matrix R is not free. It is a rotation matrix: $R^TR=RR^T=I$ and the SSD criterion must be opyimized under this restriction.

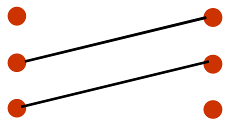
This problem can be solved using the singular vector decomposition of the data.

unknown correspondence



This is a difficult problem!

We need to estimate a permutation matrix.



$$p = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

which minimizes the matching criterion E.

tough!

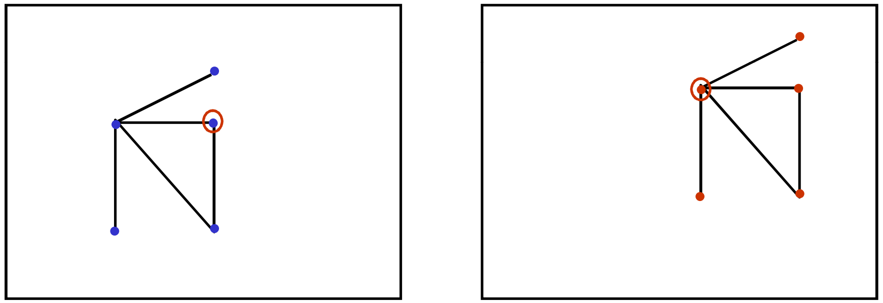
See the paper by Maciel & Costeira, PAMI03

suboptimal approaches are used instead!

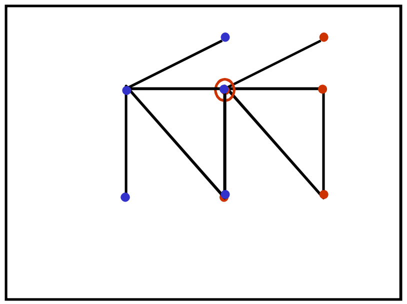
ransac

RANSAC stands for Random Sample Consensus (Fischler, Bolles, 1981)

It is based on hypothesis generation and classification of data points as inliers and outliers.



estimate translation



only 2 points are matched!

bad attempt!

ransac (2)

Objective: to estimate a transform W(x,q) with 2n degrees of freedom.

Algorithm

Hypotheses generation

randomly select n pairs of points (x_i, x'_k)

estimate the geometric transformation $W(x,\theta)$

Compute the number of points which were correctly aligned (support) i.e., such that

$$| x'_k - W(x_i, \theta) | < \varepsilon$$

Model selection: choose the transformation with largest support

Refinement: improve the estimate of θ by applying the least squares method to the subset of points which are well aligned.

example - registration

Afine transform

(3 marks)

(Matlab demo)

example - mosaicing

homography

(4 marks)

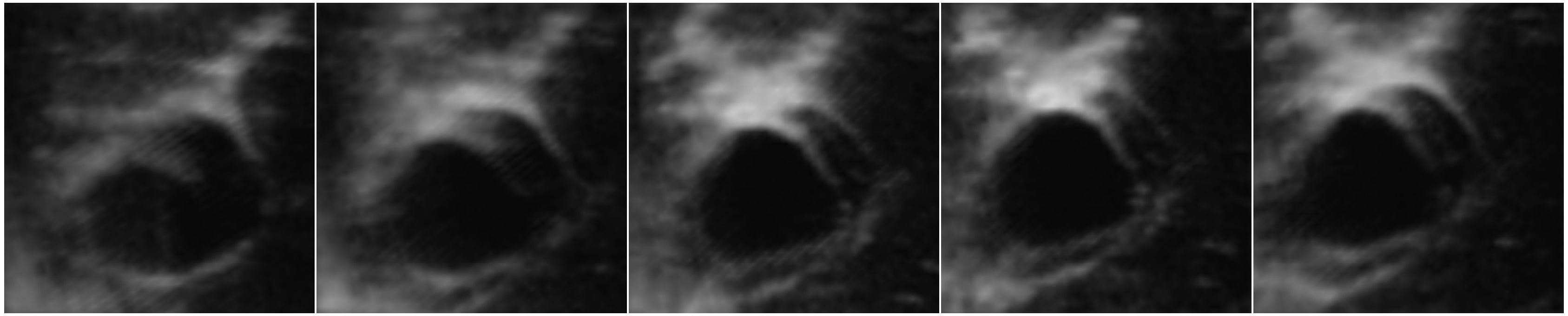
exemplo (cont.)

mosaicing

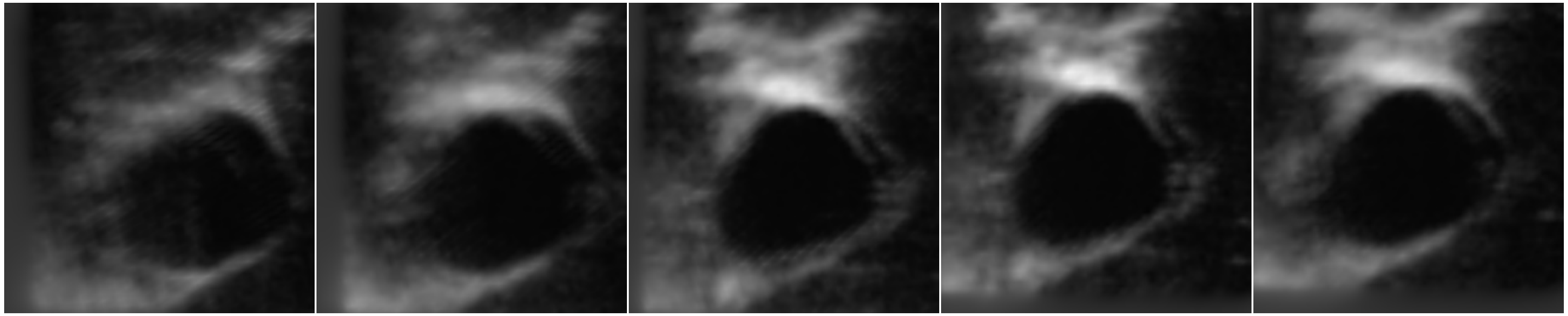
mosaicing → alignment + fusion

3D ultrasound

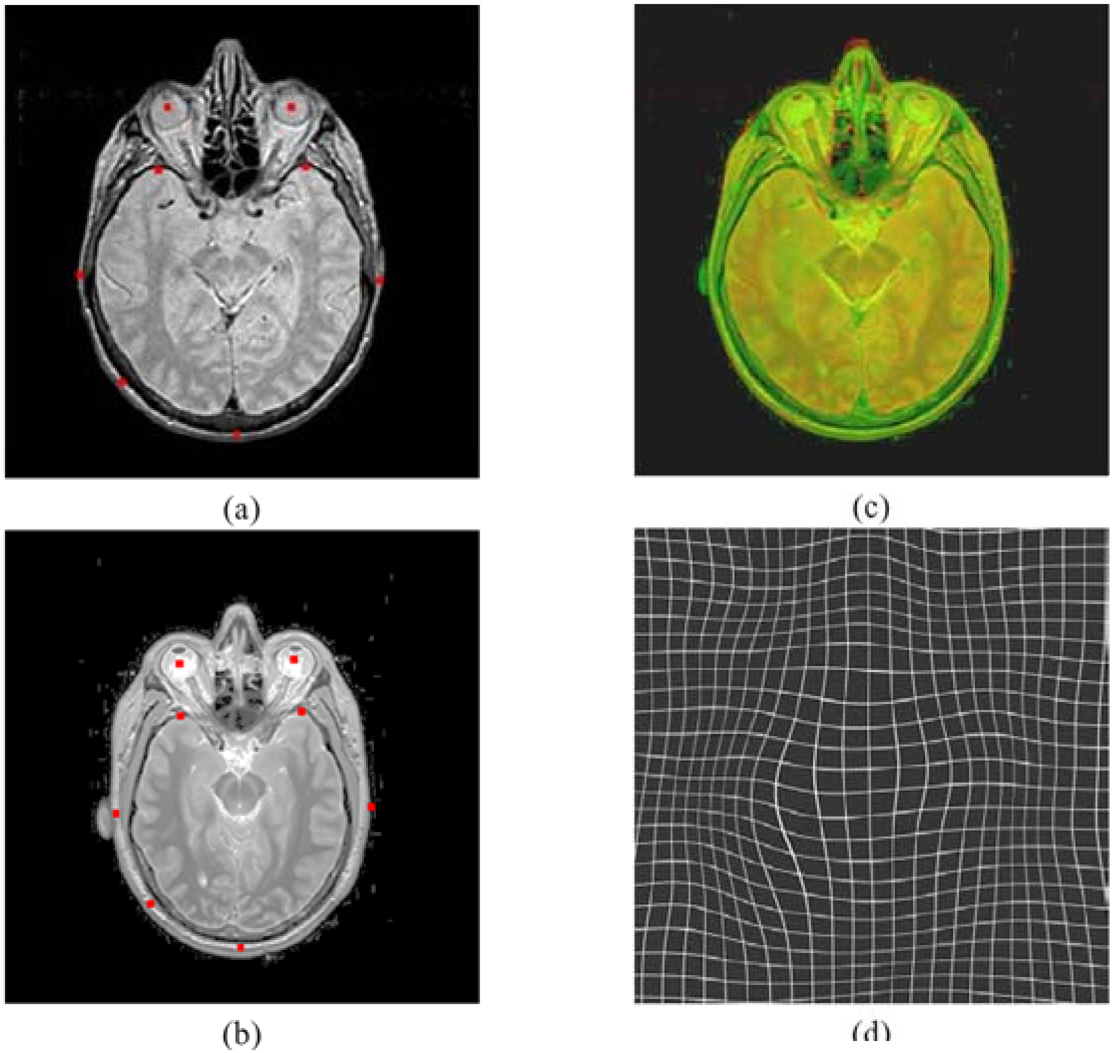
without alignment



with alignment



non-rigid alignment



Kybic, Unser, 2003

region tracking

Two steps region detection
region tracking

Region detection

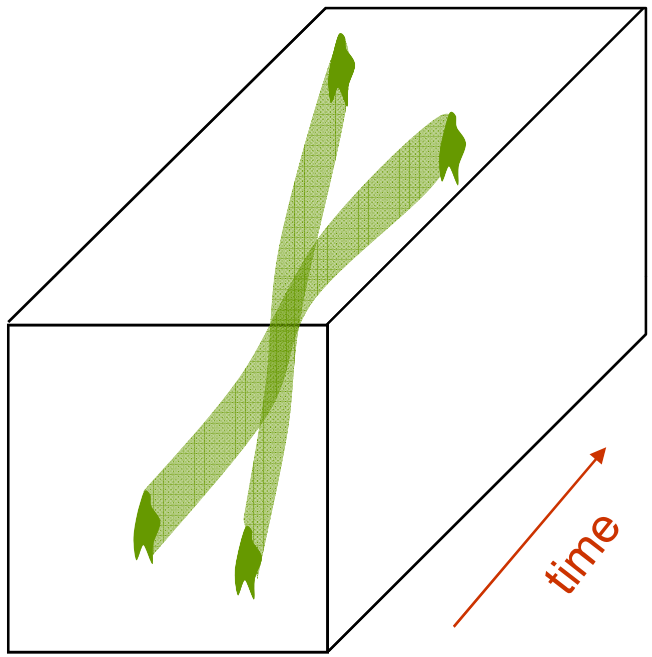
problem

goal:

- detect all moving objects

assumptions

- static camera
- static background
- show illumination changes

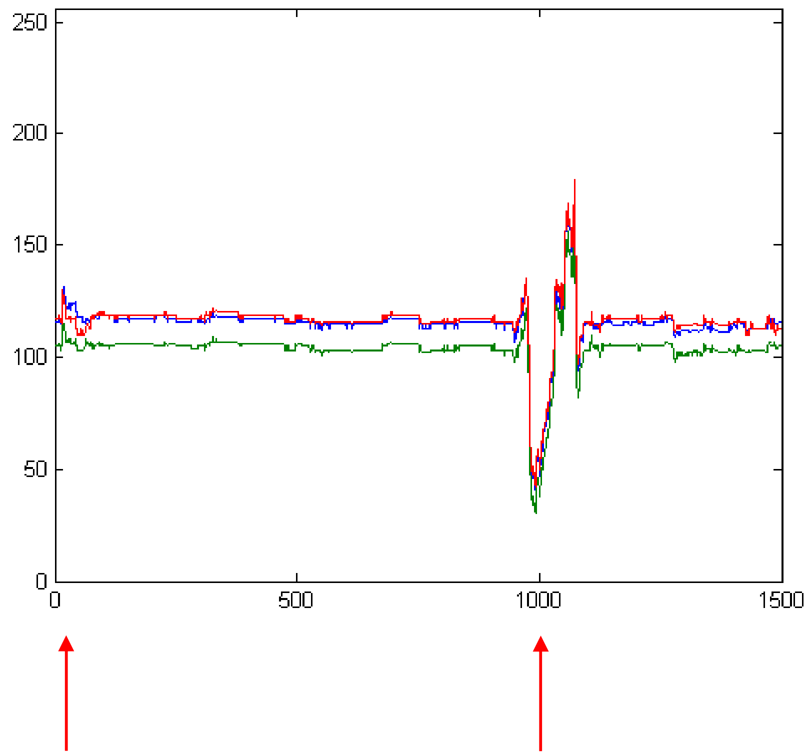


Evolution of pixel color

pixel

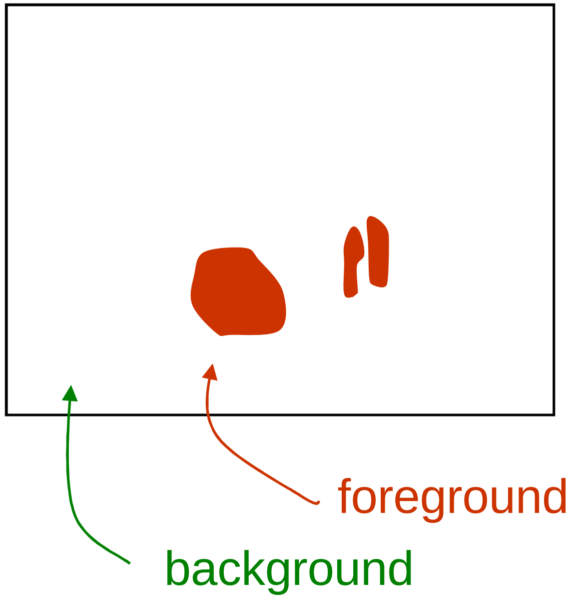
t=10

t=1000



Background subtraction

background image



Pixel classification

If $|I(x,y)-B(x,y)| < \varepsilon$, the pixel is classified as background pixel. Otherwise it is classified as active.

Basic background subtraction

The basic background subtraction classifies a pixel I(x,y) as active if

$$A(x,y) = 1 \qquad \text{if } |I(x,y) - B(x,y)| > \lambda$$
$$A(x,y) = 0 \qquad \text{otherwise}$$

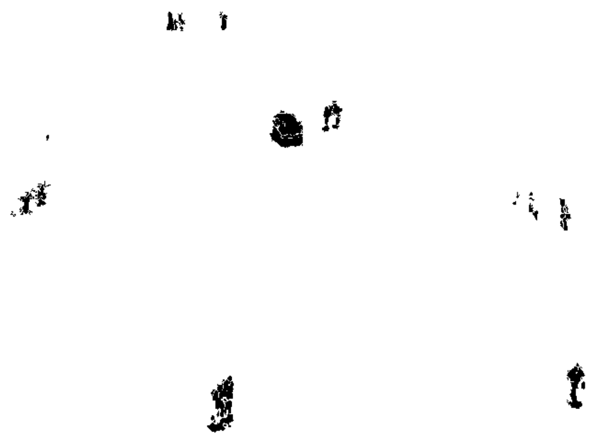
Image A(x,y) is very noisy. It has many small regions classified as active and some true objects appear fragmented in several regions.

Morphologinal post-processing is usually done. Typically we compute all conected components and eliminate all the small regions.

Example

I

B

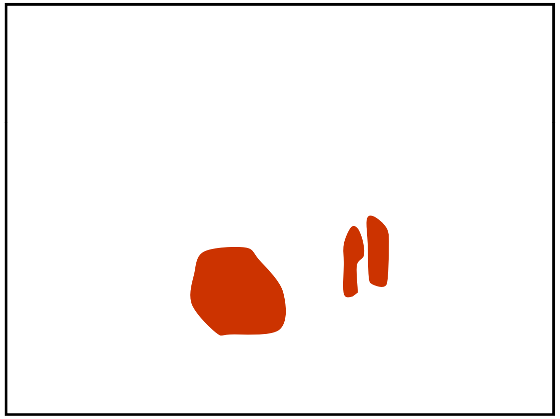


eliminação de regiões pequenas

How to deal with time-varying illumination?

Illumination changes can be compensated by the adaptation of the background image.

Only the pixels belonging to the background regiion should be adapted.



$$B(x,y,t) = \alpha B(x,y,t-1) + (1-\alpha) I(x,y,t)$$ background pixels

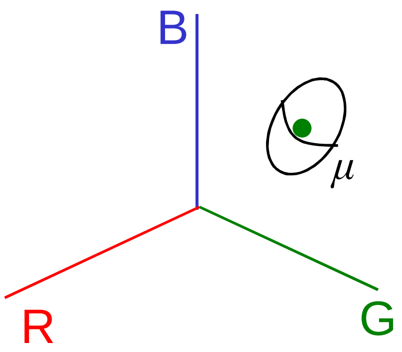
$$B(x,y,t) = B(x,y,t-1)$$ foreground pixels

Gaussian background model

(see Wren et al., 1997)

Cackground pixels are corrupted by noise. We can model each pixel as a random variable with Gaussian distribution

$$I(x,y) \sim N(\mu(x,y), R(x,y))$$



pixel classification

$p(I(x,y)) \geq \lambda$ $\Rightarrow$ background pixel

$p(I(x,y)) < \lambda$ $\Rightarrow$ foreground pixel

$$p(I(x,y)) = \frac{1}{(2\pi)^{3/2} \det(R)^{1/2}} \, e^{-\frac{1}{2}(I(x,y)-\mu(x,y))^T R^{-1} (I(x,y)-\mu(x,y))}$$

Estimation of the Gaussian model

batch

$$\mu(x,y) = \frac{1}{T}\sum_{t=1}^{T} I(x,y,t)$$

$$R(x,y) = \frac{1}{T}\sum_{t=1}^{T}(I(x,y,t)-\mu(x,y))(I(x,y,t)-\mu(x,y))^T$$

adaptive

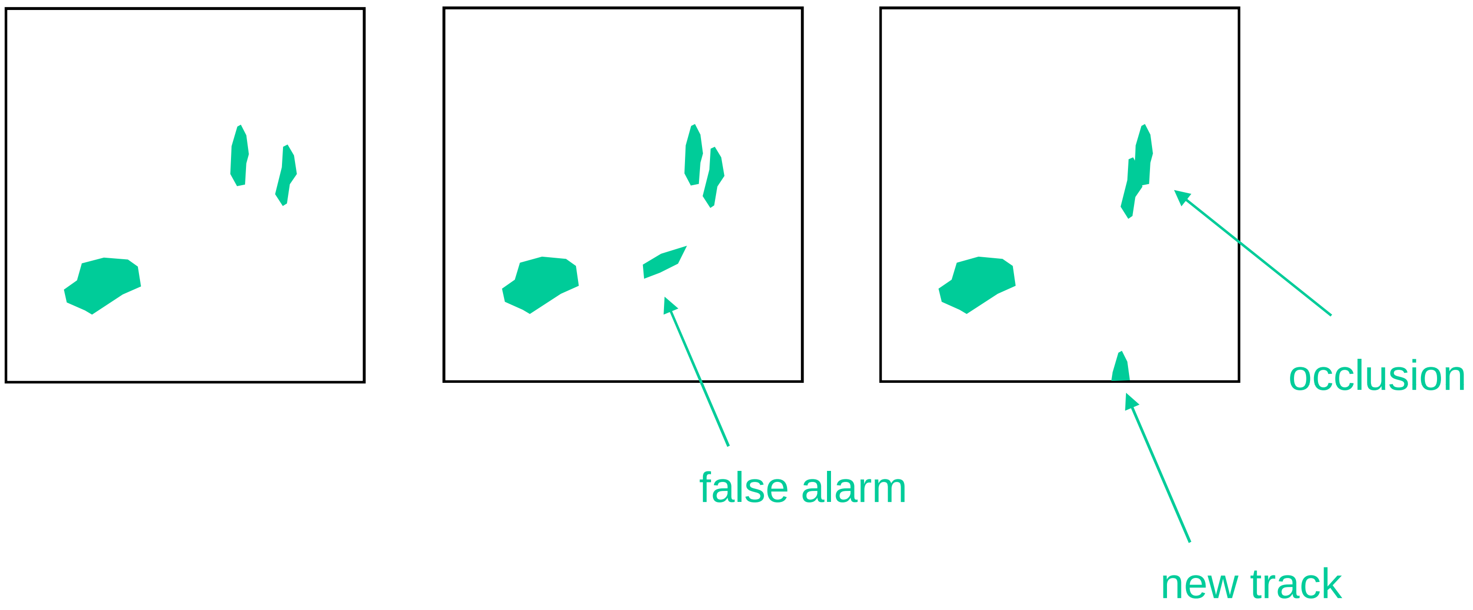
$$\mu(x,y,t) = \alpha\mu(x,y,t-1) + (1-\alpha)I(x,y,t)$$

$$R(x,y,t) = \alpha R(x,y,t-1) + (1-\alpha)(I(x,y,t)-\mu(x,y,t-1))(I(x,y,t)-\mu(x,y,t-1))^T$$

Only background pixels should be used

region tracking

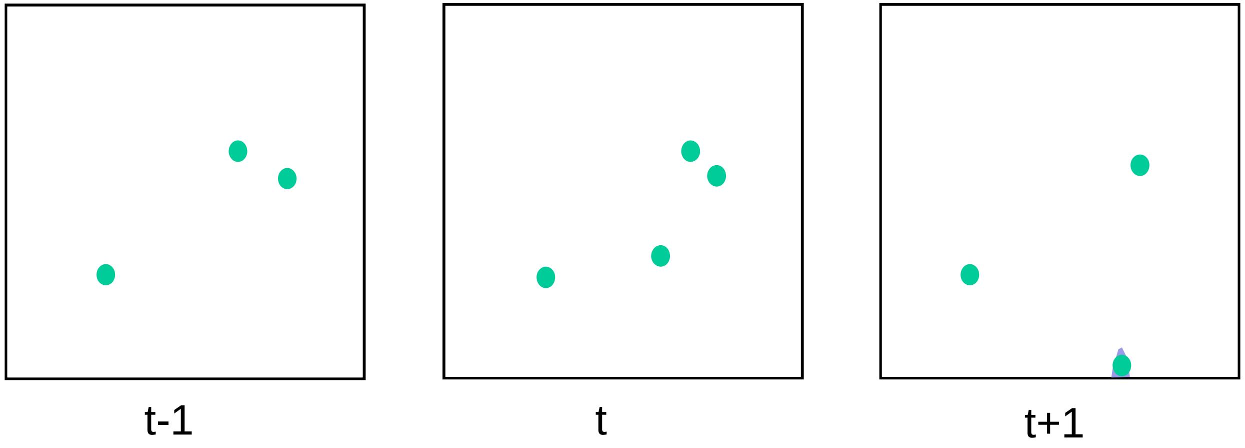
region tracking



Goal: find the trajectory of each object along multiple frames

Dificulties: misdetections, false alarms, occlusions, object splits and merges, new tracks

point tracking



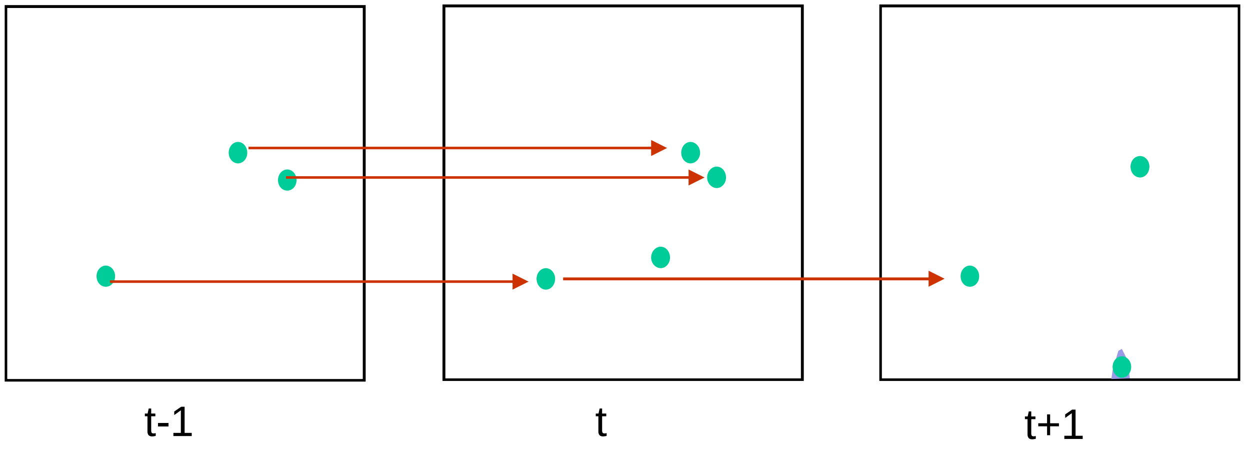
Data $D = \{ (t, p_i^t) \}$ p_i^t position of the i-th region at frame t

Track is a sequence of points detected at different (usually consecutive) frames

$$T = \{(t_1, x_1), (t_2, x_2) \ldots (t_n, x_n)\} \qquad (t_i, x_i) \in D, \quad t_i < t_{i+1}$$

$(t_{i+1} = t_i + 1)$

point association



available methods:

Statistical: propagate uncertainty and assume a dynamic model for the target trajectories (e.g., Kalman or PDA filter)

Deterministic: based on assignment costs and do not require dynamic models (e.g., graph based methods)

hypotheses

typical assumptions

(a) only regions detected in consecutive frames can be associated

(b) regions should correspond to a single target (and vice-versa)

(c) new objects may appear (track birth)

(d) objects can disapear or be occluded (track death)

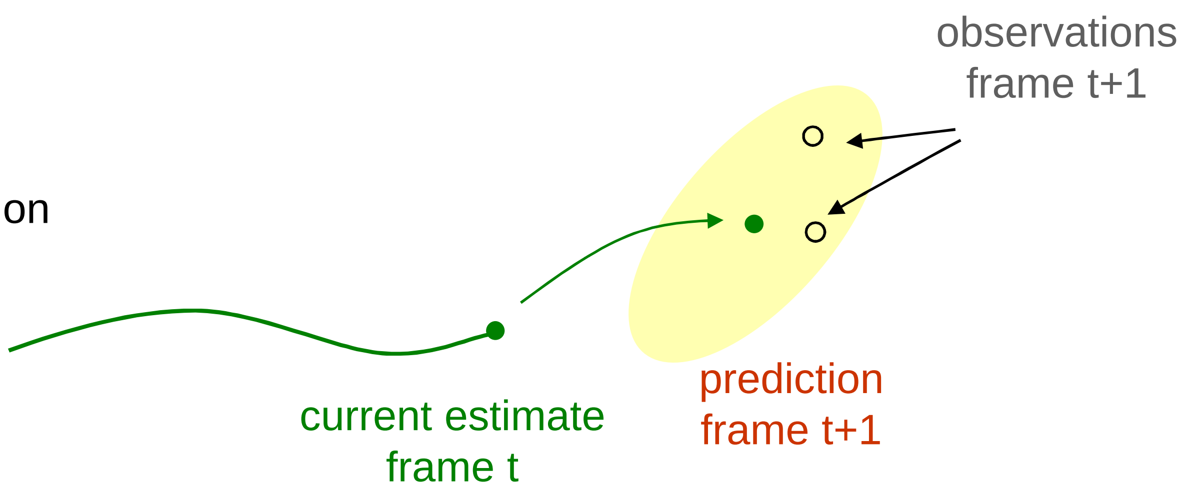
(b') objects can overlap and form groups

statistical methods

Statistical methods assume we know a set of tracks and wish to extend them in new frames.

Involve 3 steps:

- prediction
- data association
- update



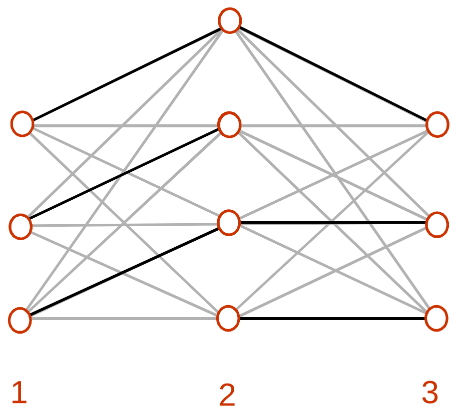
Difficulties:

- data association problem
- initialization of new tracks

Methods:

- nearest–neighbor Kalman filter
- probabilistic data association filter
- joint probabilistic data association filter
- particle filter

methods based on graphs

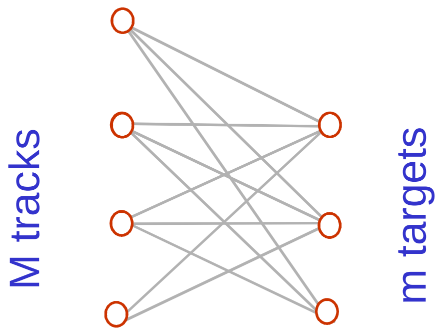


Nodes cooorespond to the detected objects in each frame and the links define a solution for the association problem

Each admissible link has a cost $C_t(i,j)$ (unconnected nodes also have a cost).

Veenman et al

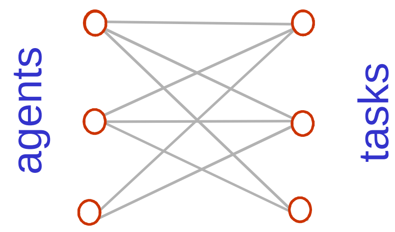
(PAMI 2001)



This method deals with pairs of frames and formulates the association of targets to existing tracks as an assignment problem if M=m.

assignment problem

Problem: there are M agents and m tasks (M=m); we wish to assign one agent to one task minimizing the total cost



$$C = \sum_{i,i=1}^{m} a_{ij} c_{ij}$$

Restrictions

$$\sum_{i=1}^{m} a_{ij} = \sum_{j=1}^{m} a_{ij} = 1 \qquad a_{ij} \in \{0,1\}$$

c_{ij} is the cost of assigning agent i to task j and a_{ij} is a binary variable wich is equal to 1 if and only if agent i is assigned to task j.

The minimization of C under these restrictions is a linear programming problem for which there are very efficient algorithms e.g., Hungarian method.

Example

tasks

$$C = \begin{bmatrix} 3 & 2 & 0 \\ 2 & 1 & 5 \\ 5 & 2 & 5 \end{bmatrix} \qquad A = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

agents

total cost: 0+2+2=4

cost matrix

In tracking, the association cost can be defined in different ways. Two popular choices are

distance criterion $a_{ij} = \| p_i^{t-1} - p_j^t \|$

prediction error $a_{ij} = \| p_i^{t-1} + v_i^{t-1} - p_j^t \|$

v_i^{t-1} displacement vector computed from a previous assignment. (cannot be used in track initialization)

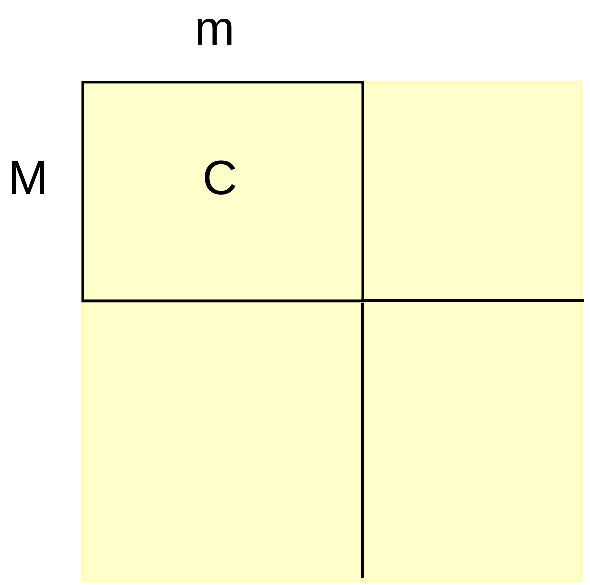
Birth and death of tracks

The previous method does not account for new tracks but it has been extended to allow birth and death of tracks

Consider a problem in which all the targets are new. In this case, all the M tracks should die are all the m targets correspond to new tracks.

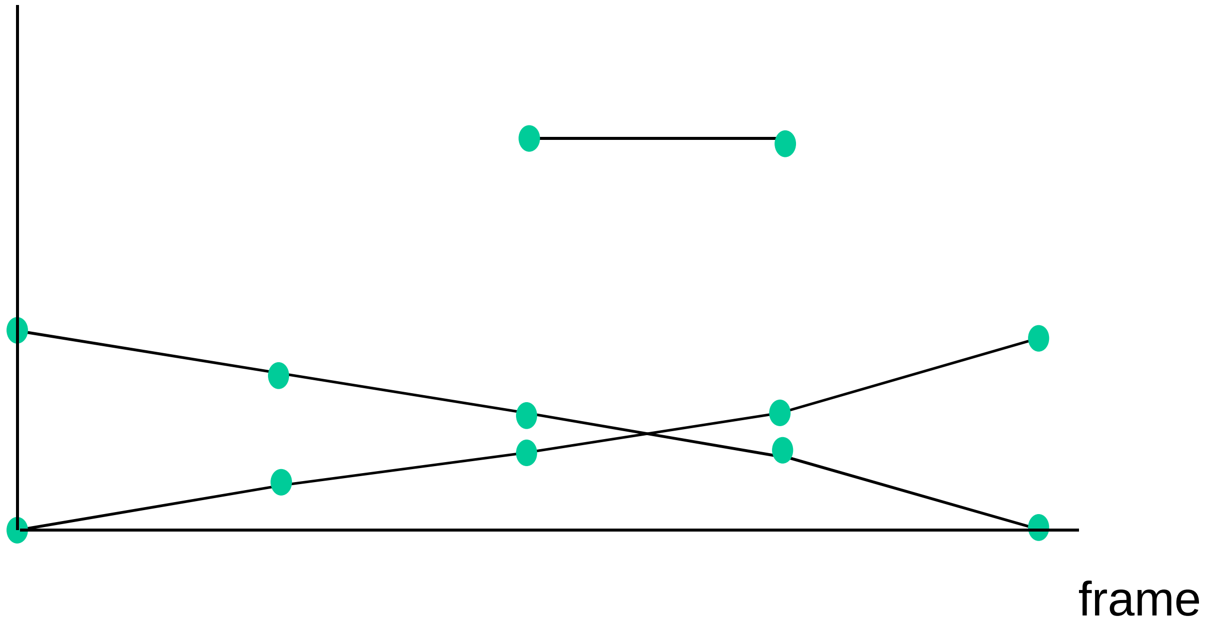
How can we do this in the previous framework?

solution: add M virtual targets and m virtual tracks



$$c_{ij} = c_{high} \quad \text{if } i > M \text{ or } j > m$$

Example 1d



costs were computed using the prediction error, except at the beginning of each track.

How can we deal with groups?

tracking of pedestrians in groups

work of Pedro Jorge

Dealing with groups

Problem

Goal: track all pedestrians in the presence of occlusions and groups

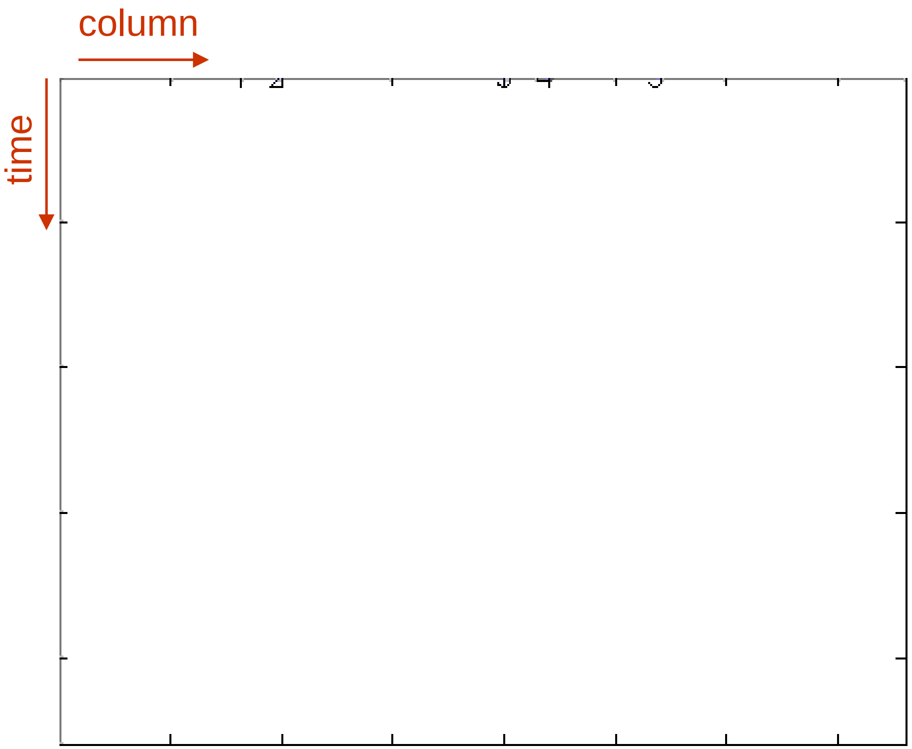
Bottom up approach

Hypothesis: 1) low level algorithms perform well most of the time

2) difficult cases should be solved by a higher level module
(e.g., occlusions, group merging and splitting)

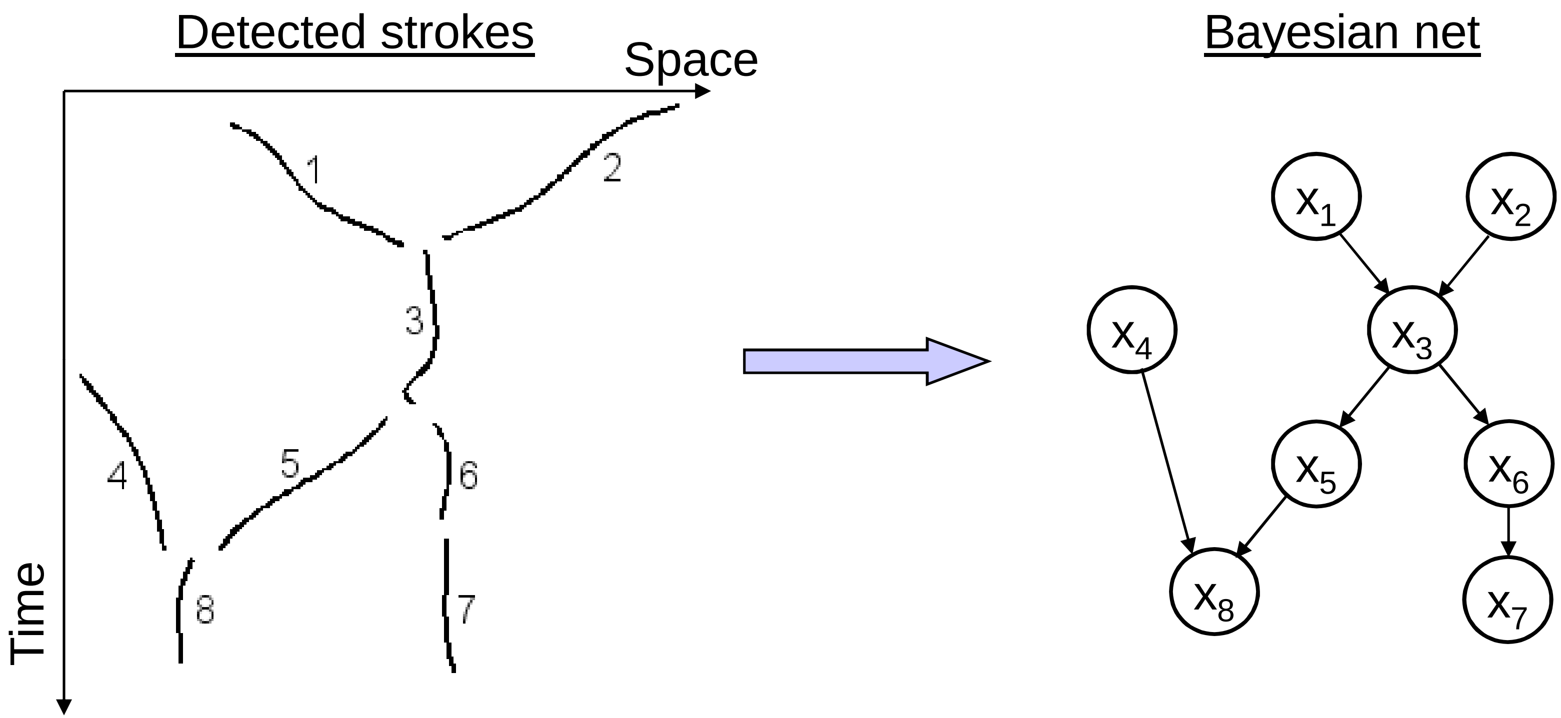
Low level processing

bakground subtraction + region matching (mutual favorite pairing)



how do we assign a color to each track (stroke)?

Problem formulation: labeling

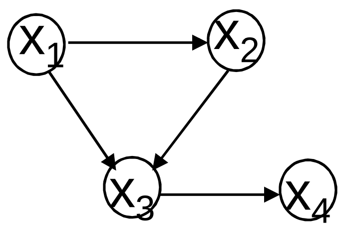


Physical constrains: causality and max. occlusion time and speed

What is a Bayesian network?

It is a probabilistic model for a set of variables $x_1, \ldots, x_n$

Drect dependences are represented by a graph



p_i are the parents of node x_i

Hypothesis (Markov property)

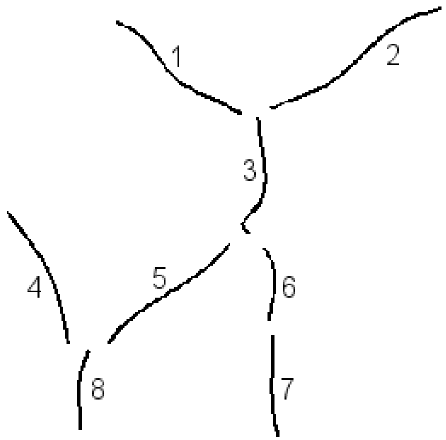
$$p(x_i \mid x_1 \ldots x_{i-1}) = p(x_i \mid p_i)$$

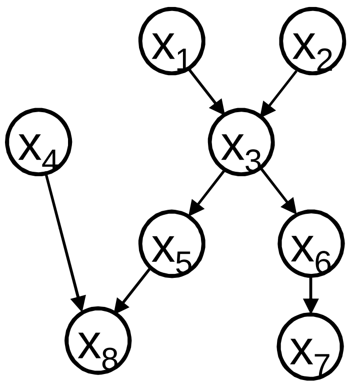
$P(x_1,x_2,x_3,x_4) = P(x_4|x_3)\, P(x_3|x_1,x_2)\, P(x_2|x_1)\, P(x_1)$

Factorization

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i \mid p_i)$$

Bayesian network generation





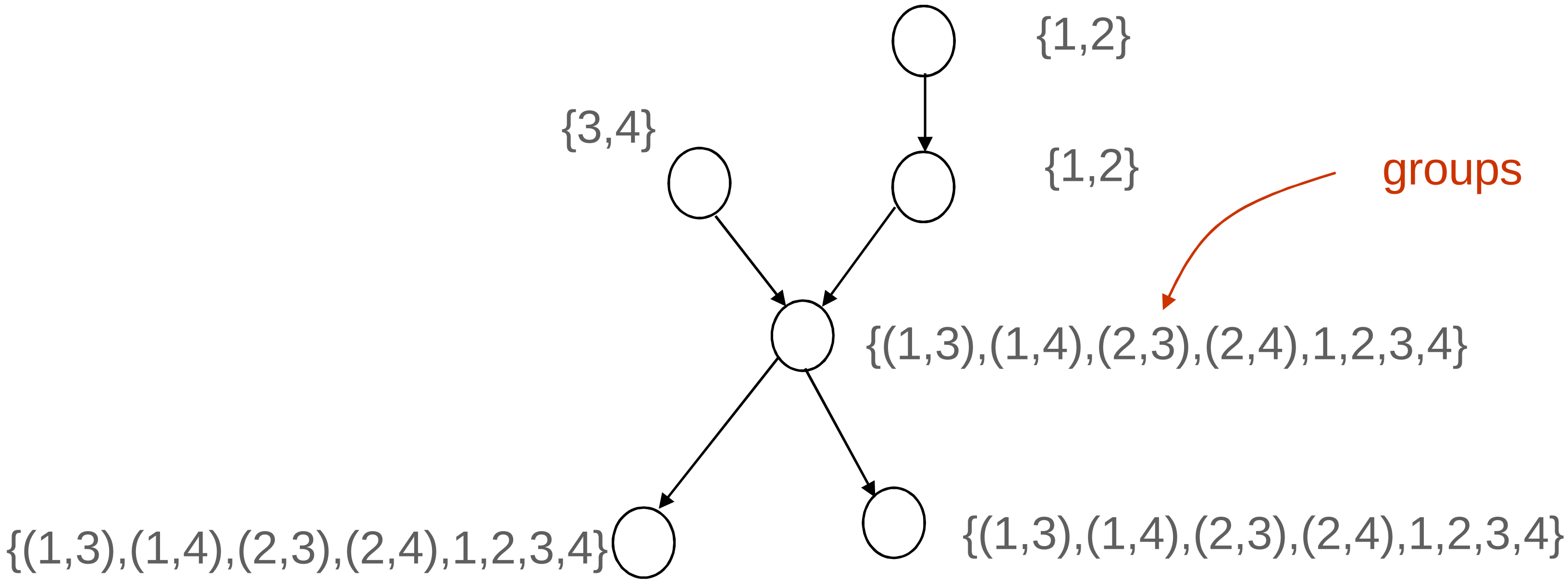
each node is associated to a stroke

links are created between nodes which are close and could be associated to the same object

We compute the set of admissible labels and probability distribution for each node.

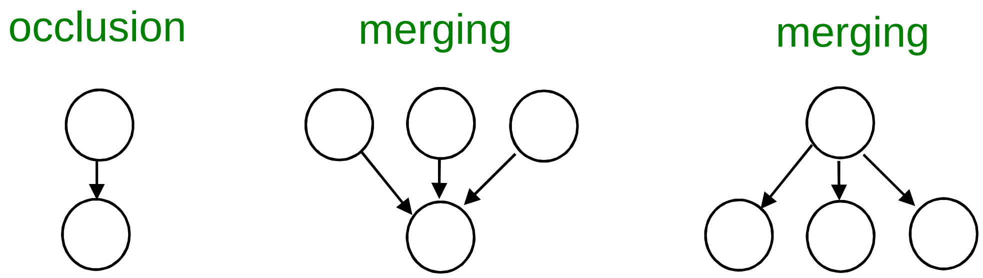
Admissible Labels

Each node has a set of admissible labels

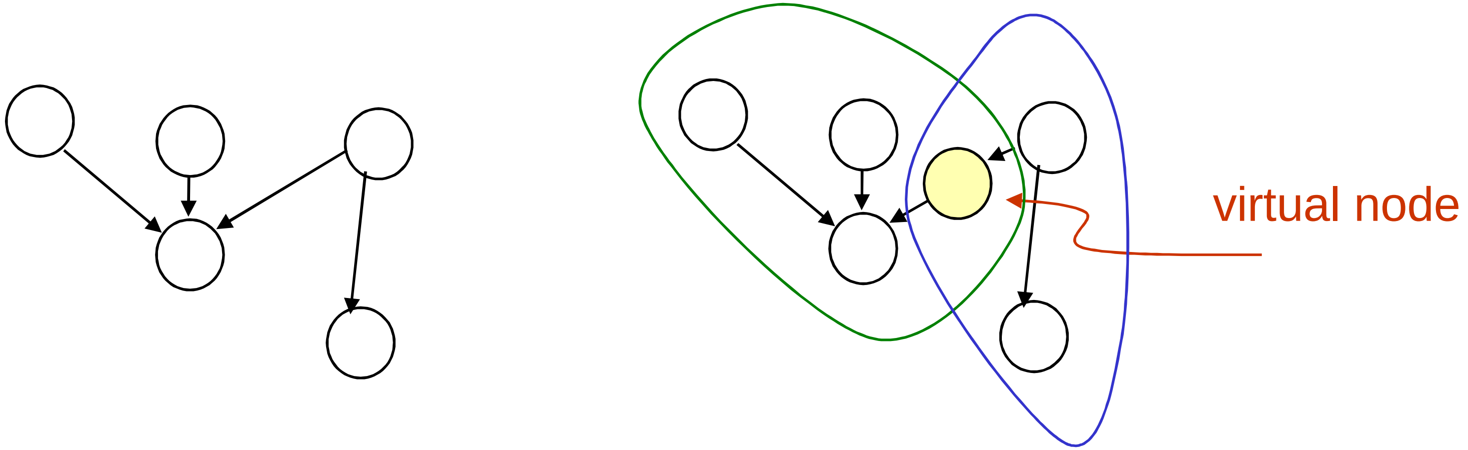


Basic Blocks

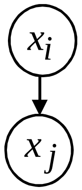
Three basic blocks



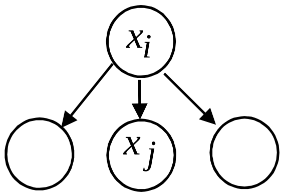
How can we deal more difficult cases (merge-split)?



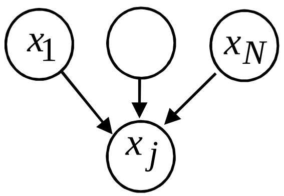
Conditional probability tables



$$P(x_j \mid x_i) = \begin{cases} P_{occ} & x_j = x_i \\ P_{new} & x_j = new \end{cases}$$



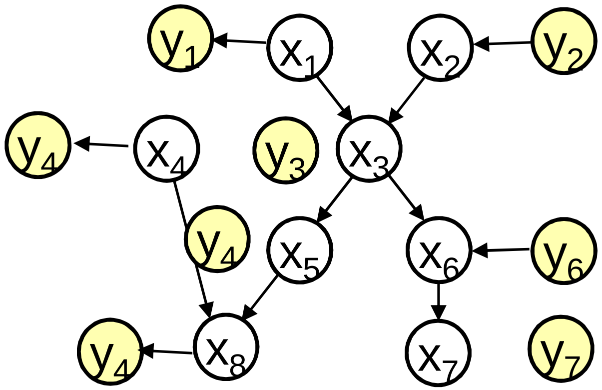
$$P(x_j \mid x_i) = \begin{cases} P_{split} / (2^{N_i} - 2) & x_j \subset P(x_i) \setminus \{x_i\} \\ P_{occ} & x_j = x_i \\ P_{new} & x_j = new \end{cases}$$



$$P(x_j \mid x_1, \ldots, x_N) = \begin{cases} P_{occ} & x_j = x_1 \text{ or } \ldots \text{ or } x_j = x_N \\ P_{new} & x_j = new \\ P_{merge} / L & otherwise \end{cases}$$

L – number of different group merges

Observations



Each label node has an observation node associated

In this work we extract the 3 most significant colors of the image region and compare with the most significant colors of the model

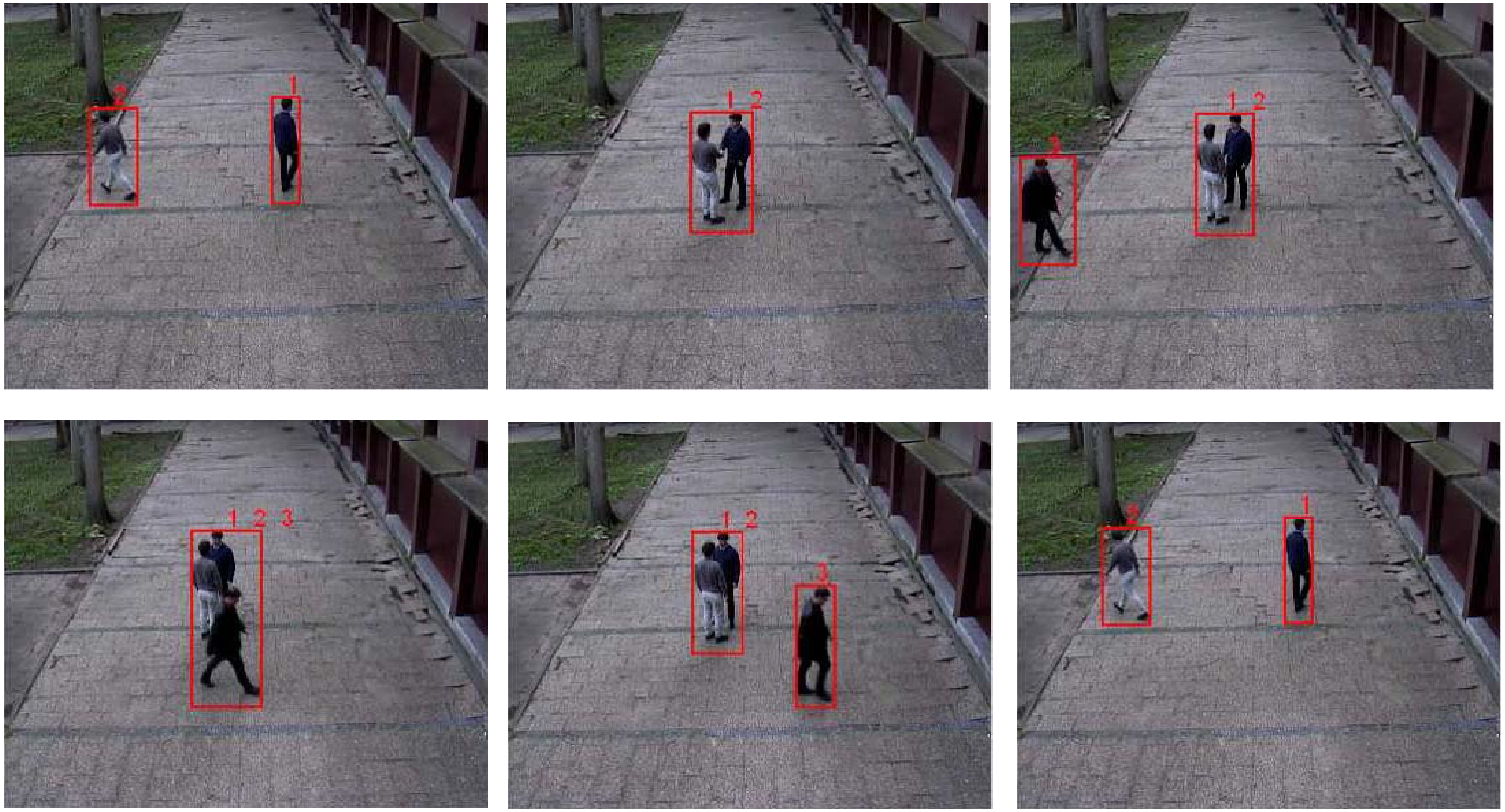
Inference

What is the label distribution in each model?

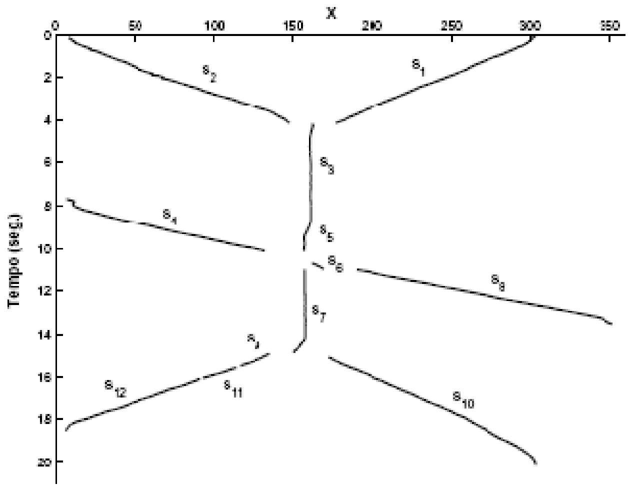
This is the role of inference! Inference can be done using the junction tree algorithm.

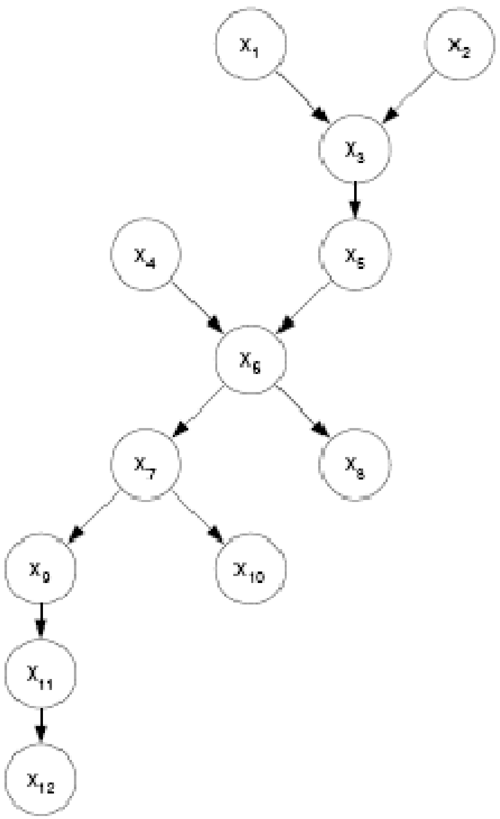
We have used Kevin Murphy's toolbox for Matlab.

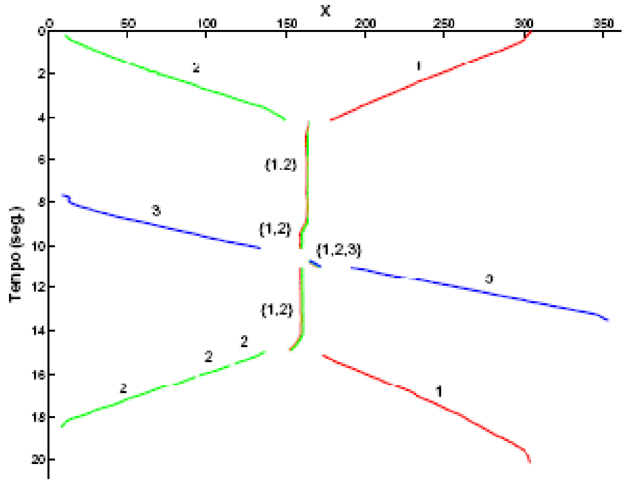
Example



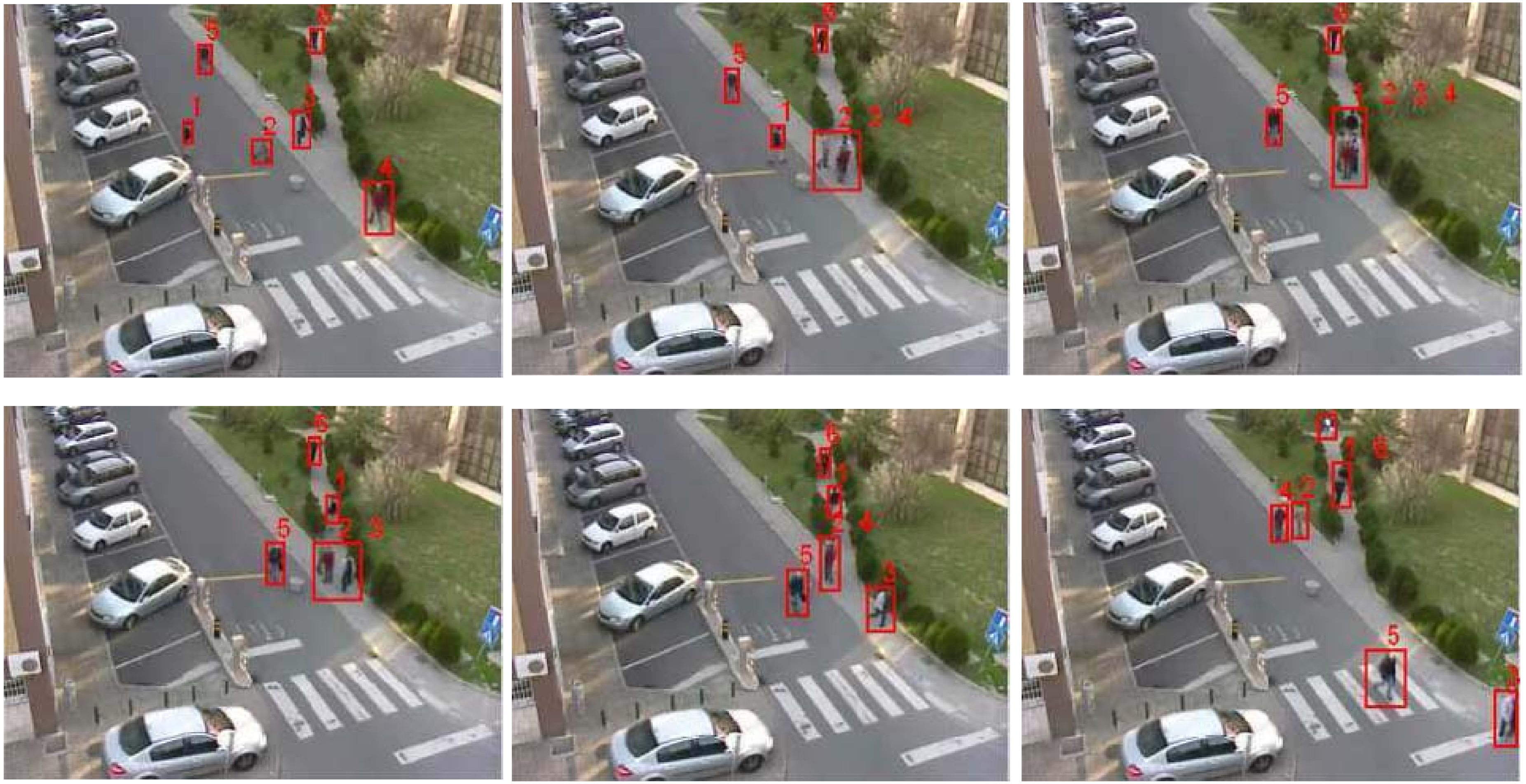
Example (2)


X
Tempo (seg.)

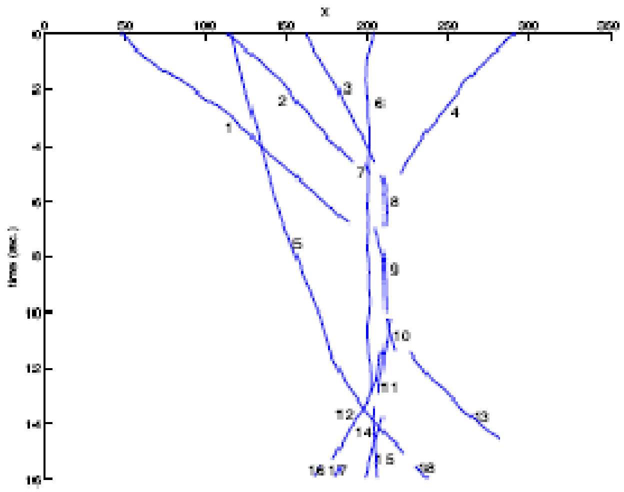

x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_9
x_{10}
x_{11}
x_{12}

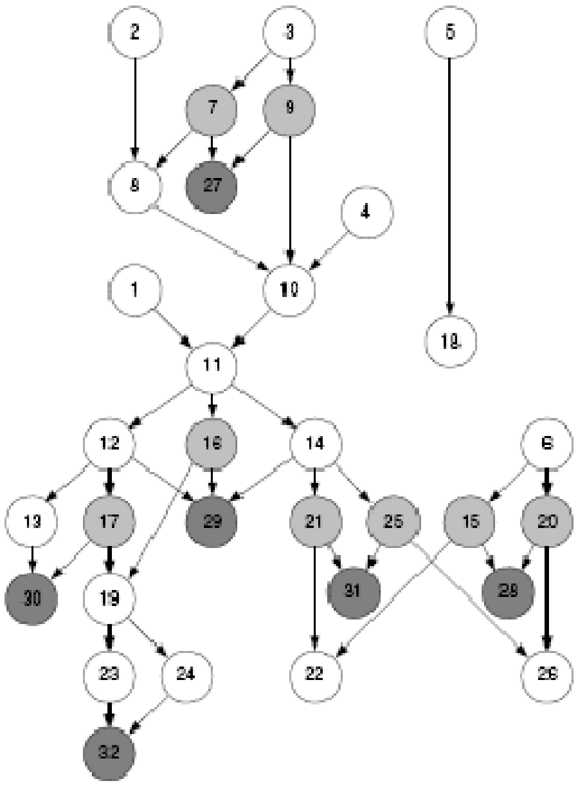

X
Tempo (seg.)
(1,2)
(1,2)
(1,2,3)
(1,2)

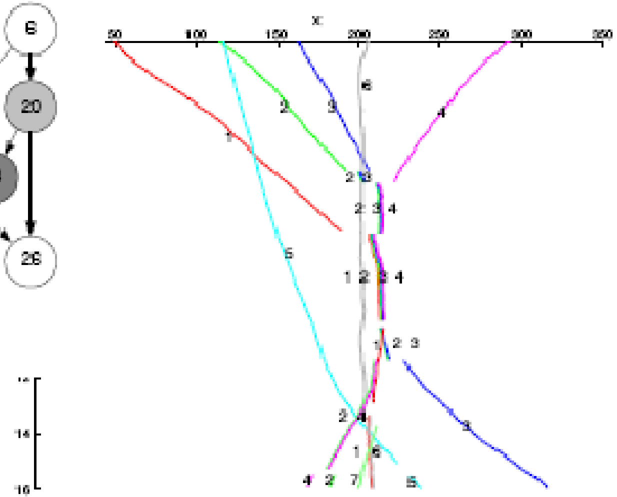
Example



Example(2)







Examples