Tracking people using video images and active contours

Instructor:

Oleg Kuybeda

Students:

Nadav Granot

Zohar Tal

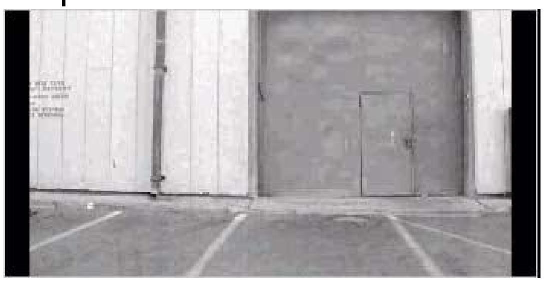
Objective

Tracking people using video images.

- **Input**: A video sequence taken from a standing camera.
- **Output**: Detection and tracking of people that are within the frame of the camera, in the video sequence. Their approximated contour will be drawn on the output image.

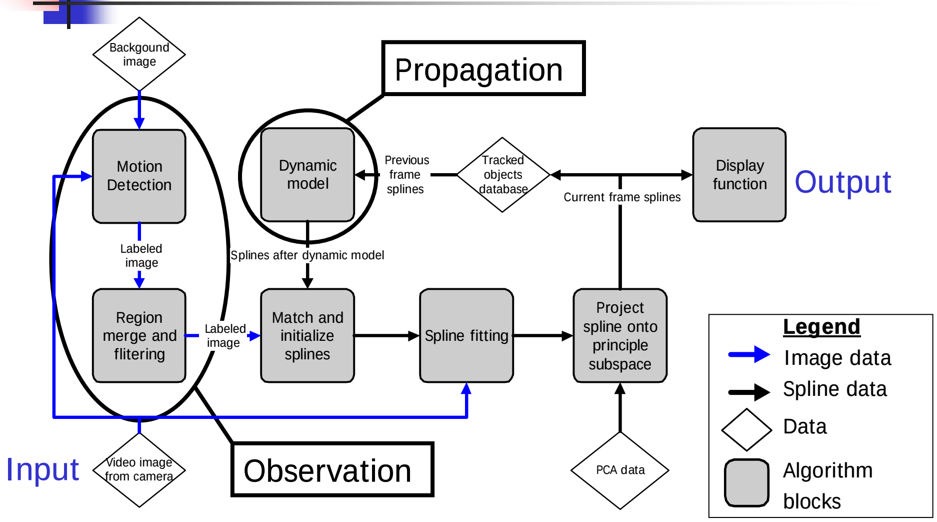
Example: Tracking one person

Proposed tracker



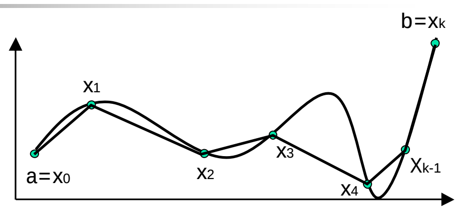
Leeds tracker

Tracker flow



Splines

Suppose we have a set of points, and we want to calculate a curve between them:



One option is obvious – the linear:

$$(1)\quad P(x) = P_i(x) \qquad x_i < x < x_{i+1} \qquad i = 0,1,..,k-1$$

$$(2)\quad P_i^{(j)}(x_i) = P_{i+1}^{(j)}(x_i) \qquad j = 0,1,..,r-1$$

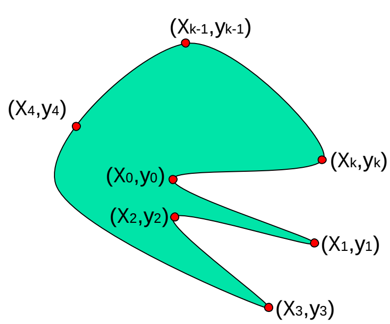
Commonly used parameters:

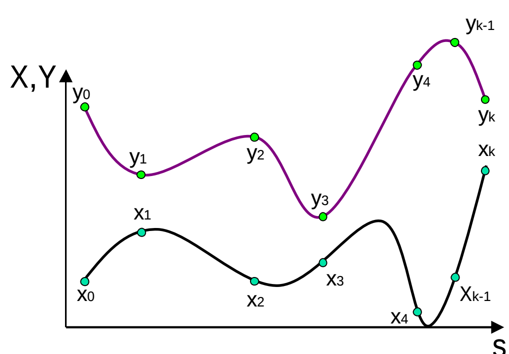
- m – order of the polynoms P_i
- r – number of derivatives
- k – number of intervals in the curve

But, we can add more dimensions and more restrictions, by changing (2) into a set of equations

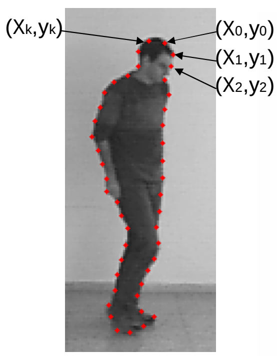
Contour representation using splines

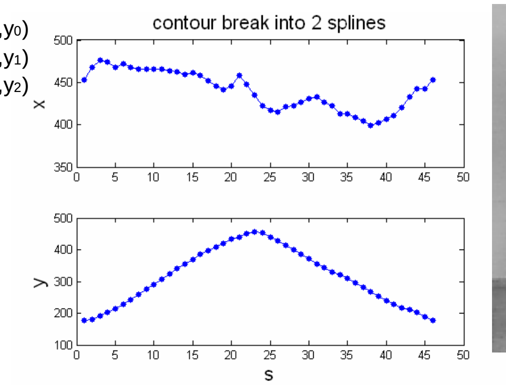
- A multi dimensional function can be represented using spline theory, by creating splines coordination wise.
- There is a restriction – The sampling axis should be the same for all coordinates

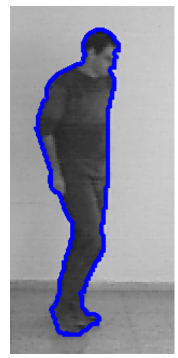




Example: A real contour





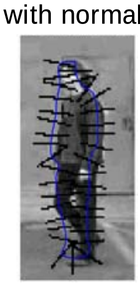


Spline fitting

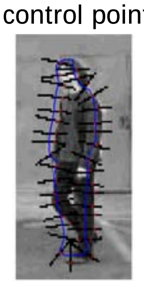
Old spline

Sampled points of old spline

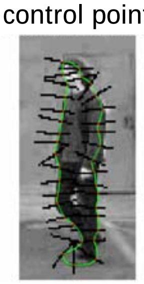
Old spline with normals



Old spline with new control points

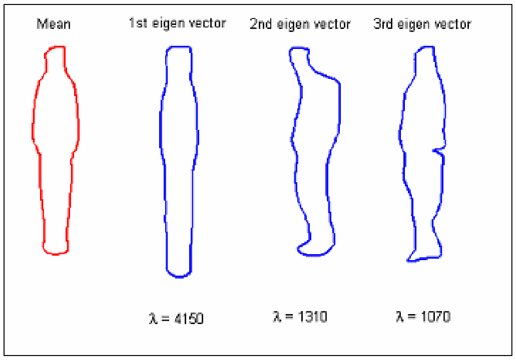


New spline with new control points



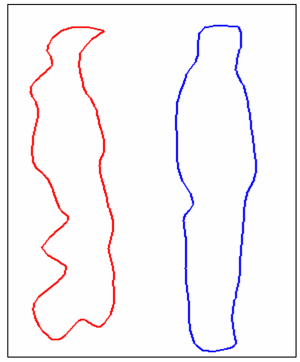
Project spline onto principle subspace

PCA database



Effect of PCA filtering

Original Projected

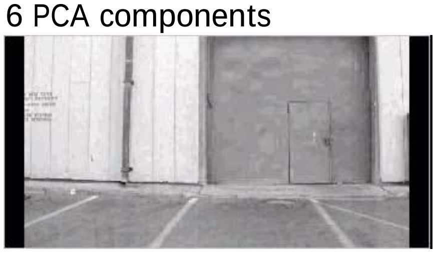


Simulation analysis

- Example 1 – influence of the number of PCA components on the tracking

1 PCA component

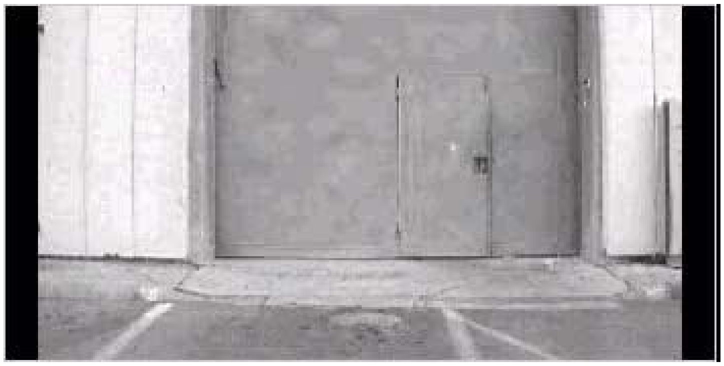
6 PCA components



2 PCA components

Simulation analysis

- Example 2 – Tracking more than 1 person



Simulation analysis

- Example 3 – Tracking a person in a mixed environment

Simulation analysis

- Example 3 – Tracking a person in a mixed environment

Future development

- Technical improvements
- Behavior recognition layer
  - Security
  - Marketing
  - Traffic analysis
  - Non-Human objects

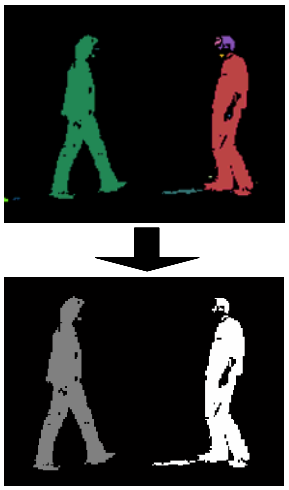
The End

Motion detection

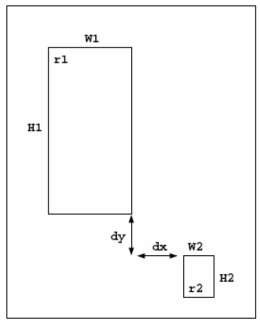
- Differentiate
- Threshold
- label

- Remark: In this project, background estimation is not considered an issue

Region merge and filtering



- Help us “not to lose our heads”
- Remove some clutter



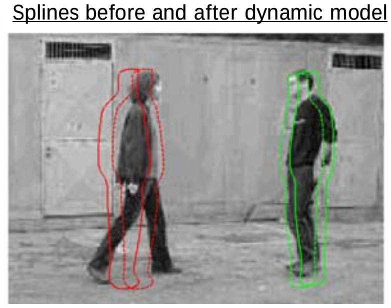
  - Remove very small regions
  - Calculate bounding box for each remaining region
  - Merge regions, using equation (1)
  - Remove regions that are smaller than a human

$$(1)\quad D_{r_1,r_2} = \frac{dx+dy}{\max(W_{r_1},H_{r_1})+\max(W_{r_2},H_{r_2})+1}$$

Dynamic model

- Model is implemented only for motion, not for contour propagation
- Head position is taken as the reference point
- A more complex model can be implemented for better robustness

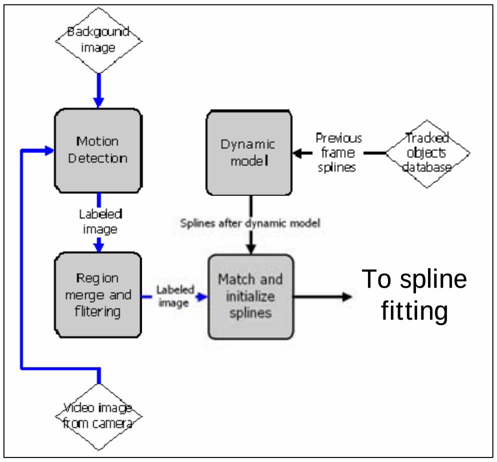
Splines before and after dynamic model



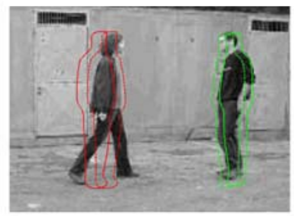
Solid line – Previous frame spline

Dashed line – Predicted splines

Match and initialize splines



- Match detected regions with tracked people
- Add new detected people
- Remove lost trackings



Splines - definition

$$(1)\quad P(x) = P_i(x) \qquad x_i < x < x_{i+1} \qquad i = 0,1,..,k-1$$

$$(2)\quad P_i^{(j)}(x_i) = P_{i+1}^{(j)}(x_i) \qquad j = 0,1,..,r-1$$

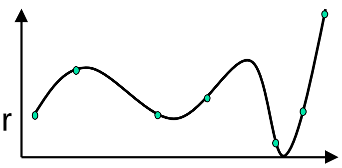
A spline is a Piecewise Polynomial, given by (1) and (2), in which r=m (simple spline) or r<m (regular spline).

Commonly used splines:

- Linear Spline : r=m=1
- Quadratic Spline: r=m=2
- Cubic Spline: r=m=3

Representation of splines

- In the traditional way, the position of the set of points (also known as breakpoints), and the polynom coefficients for each piece are the set of parameters that should be calculated.



- One wants to represent the spline in a method that addresses 2 main demands:
  - Computationally simple.
  - A local change in the set of points will make only a local change in the spline, leaving the rest of it unchanged.

B-Splines - Definition

- A recursive definition:

$$N_{i,0}(x) = \begin{cases} 1 & x_i \le x \le x_{i+1} \\ 0 & else \end{cases}$$

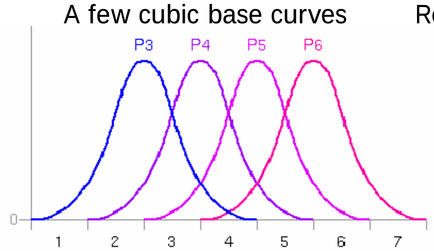
$$N_{i,m}(x) = \frac{x - x_i}{x_{i+m} - x_i} \cdot N_{i,m-1}(x) + \frac{x_{i+m+1} - x}{x_{i+m+1} - x_{i+1}} \cdot N_{i+1,m-1}(x)$$

- The family of curves $N_{i,m}(x)$ can be used to represent a spline as a linear combination of the curves in the family (the base curves):

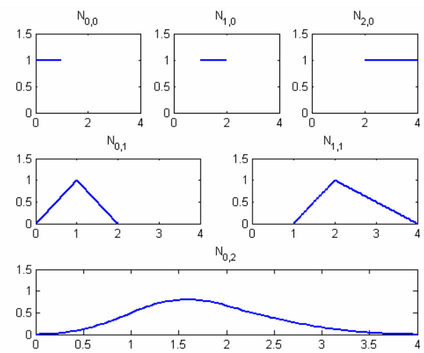
$$P(x) = \sum_{i=-m}^{k-1} a_i \cdot N_{i,m}(x)$$

A little more about base curves

A few cubic base curves



Recursive build of a quadratic base curve



Comments:

- Easy representation:

$$P(x) = \sum_{i=-m}^{k-1} a_i \cdot N_{i,m}(x)$$

- A change in one point would require recalculation of only m+1 intervals, while the rest of the spline is unchanged.

Representation dimensionality

- A typical human contour would require about 45 intervals, depending on distance from camera and amount of contour details required.
- This means that a B-Spline representation of the contour would require about twice as much coefficients (a spline for each dimension).
- Since human contours are very similar to one another, the dimension of the vector used for contour representation can be largely reduced, using Principal Component Analysis - PCA.
- The result is a representation vector of about 5 coefficients.
- By reducing the number of coefficients to 5, we do component filtering to the fitted splines, thus allowing better clutter handling.

Building the PCA database

- In a controlled environment (white background, dark clothes), record all human gestures.
- Use this video to acquire the shadow of a person in all positions.
- Sample the contour in n points, and use them for splines calculation.
- Use spline coefficients as input to the PCA algorithm.

back

PCA Calculation

- Given an observation matrix M, in which every column is an observation and every row is a component, construct the covariance matrix by:

$$\Lambda = \frac{1}{n} M M^T \qquad M : [m, n]$$

- Find eigen values λ_i of the covariance matrix, by solving:

$$\left| \Lambda - I\lambda_i \right| = 0$$

- Find eigen vectors v_i of the covariance matrix, by solving:

$$\left(\Lambda - I\lambda_i \right) \cdot v_i = 0$$

back

Relevant articles and work

- Michael Isard and Andrew Blake, Condensation - conditional density propagation for visual tracking, Int. J. Computer Vision (1998).
- Nils T. Seibel, Design and implementation of people tracking algorithms for visual surveillance applications, university of Reading (2003).
- Christoph Kiefer, Qualitative and Quantitative Evaluation of the Reading People Tracker, Swiss Federal Institute of Technology (2004).