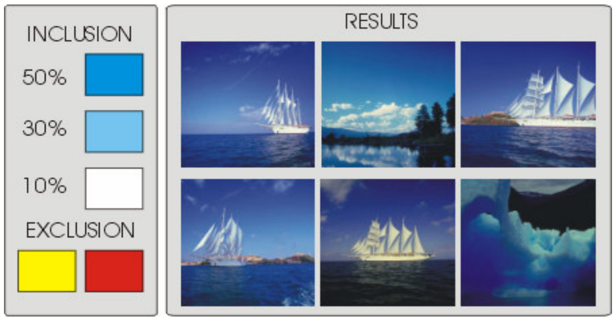
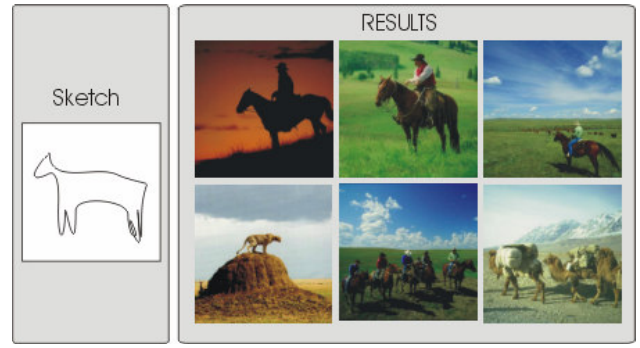
QUERY TYPES

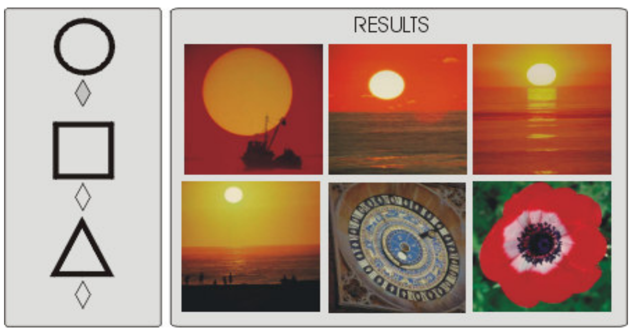
COLOR

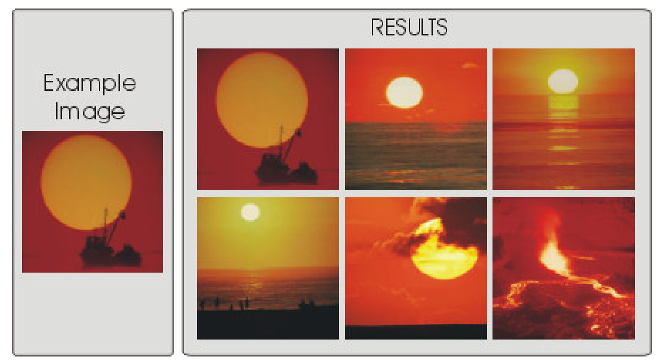




SKETCH

SHAPE





EXAMPLE

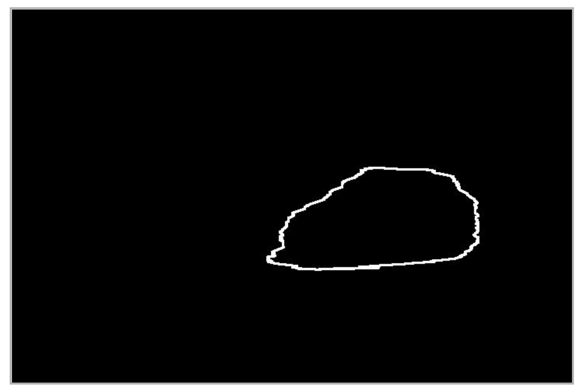
TEXTURE

Feature Extraction (Shape and Motion trajectory)

- Background Subtraction used to extract moving foreground objects *

- Gaussian Mixture Model is used to model the background

- Shape extracted from median frame of the video shot



- Centroids of moving object used to extract the motion trajectory

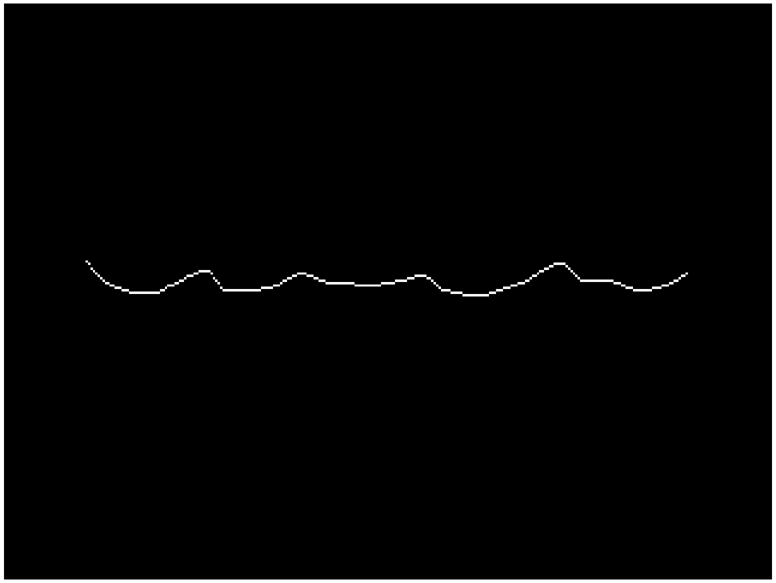
*** OpenCV, "http://sourceforge.net/projects/opencvlibrary/"***

Motion Trajectory Extraction

Input video

Video showing contour & centroid

Trajectory points



Trajectory after interpolation

Shape Representation – Curvature Scale Space

- Powerful and general shape analysis tool developed comprehensively by Mokhtarian and standardized in MPEG-7

- Multi Scale Curvature based shape representation for planar curves

- Steps involved
  - Smooth the contour with a Gaussian to get evolved curves at different scales
  - Find zero crossing curvature of the evolved curves which form the CSS image
  - The maxima in the CSS image is used for matching two contours

- Given planar curve (shape contour)

$$\Gamma = \{(x(u), y(u)) \mid u \in [0,1]\}$$

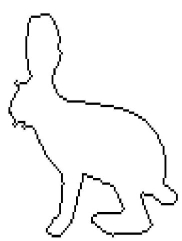
- Evolved planar curve is defined by

$$\Gamma_\sigma = \{(X_\sigma(u), Y_\sigma(u)) \mid u \in [0,1]\}$$

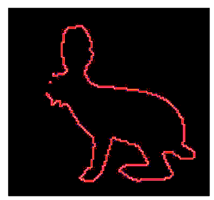
$$X_\sigma(u) = x(u) \otimes g_\sigma(u)$$

$$Y_\sigma(u) = y(u) \otimes g_\sigma(u)$$

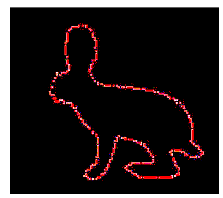
$$g_\sigma(u) = \frac{1}{\sigma\sqrt{2\pi}} e^{-u^2/2\sigma^2}$$



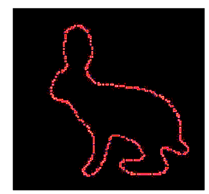
Evolved curves at different scales



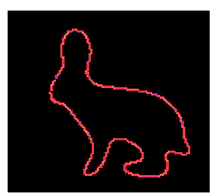
σ =0.5



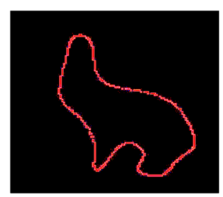
σ =1



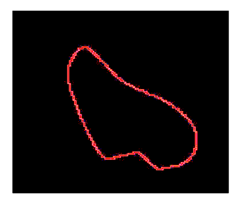
σ =2



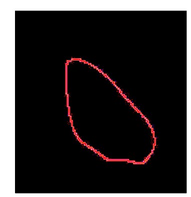
σ =4



σ =8



σ =16



σ =32

σ -> standard deviation of Gaussian used for smoothing

- Curvature of Γ_σ (evolved curve) is given by

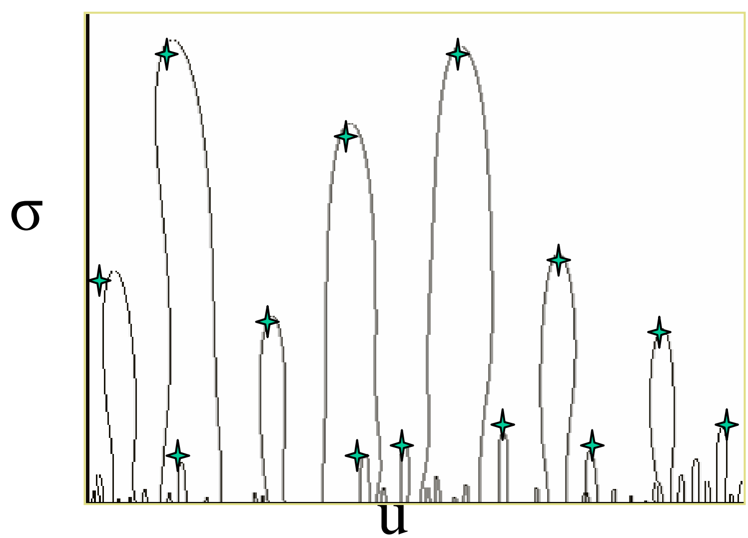
$$k_\sigma(u) = \frac{X_{\sigma,u}(u)Y_{\sigma,uu}(u) - Y_{\sigma,u}(u)X_{\sigma,uu}(u)}{(X_{\sigma,u}(u)^2 + Y_{\sigma,u}(u)^2)^{3/2}}$$

$$X_{\sigma,u}(u) = x(u) \otimes g_{\sigma,u}(u)$$
$$X_{\sigma,uu}(u) = x(u) \otimes g_{\sigma,uu}(u)$$

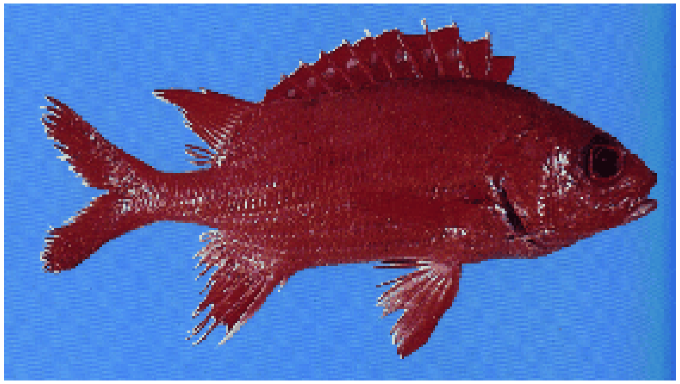
- Solution of equation $k_\sigma(u) = 0$

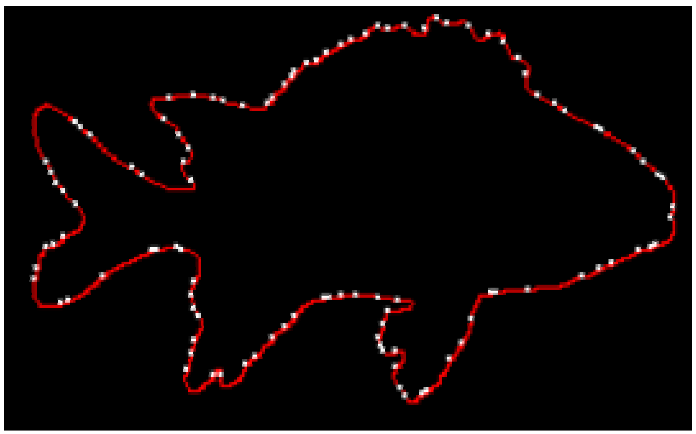
gives zero crossing curvature which forms Curvature Scale Space Image

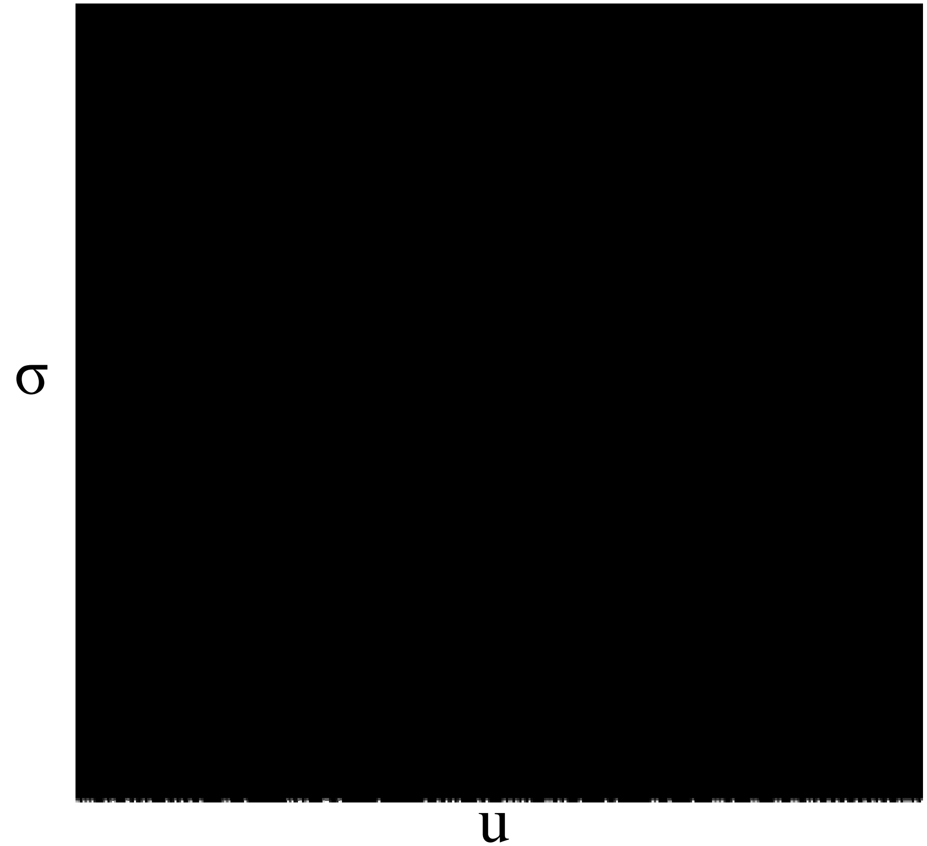
CSS Image



✦ ⟶ Maxima used to represent shape of video object







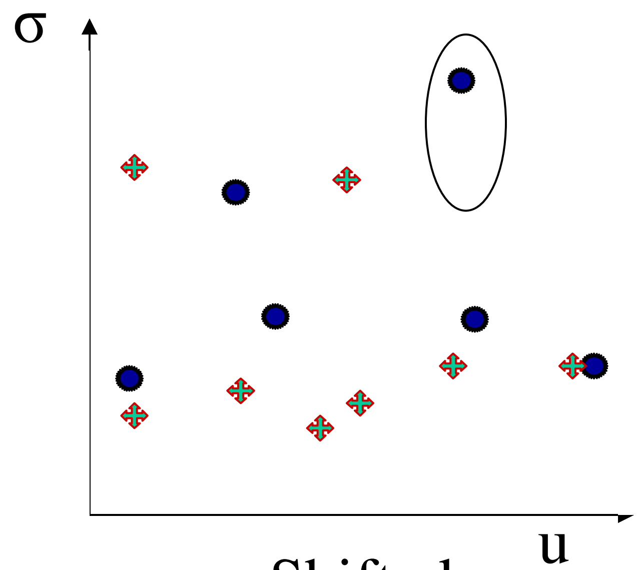
There are two curvature zero crossings on every concave and convex part of shape

As curve becomes smoother these points approach each other

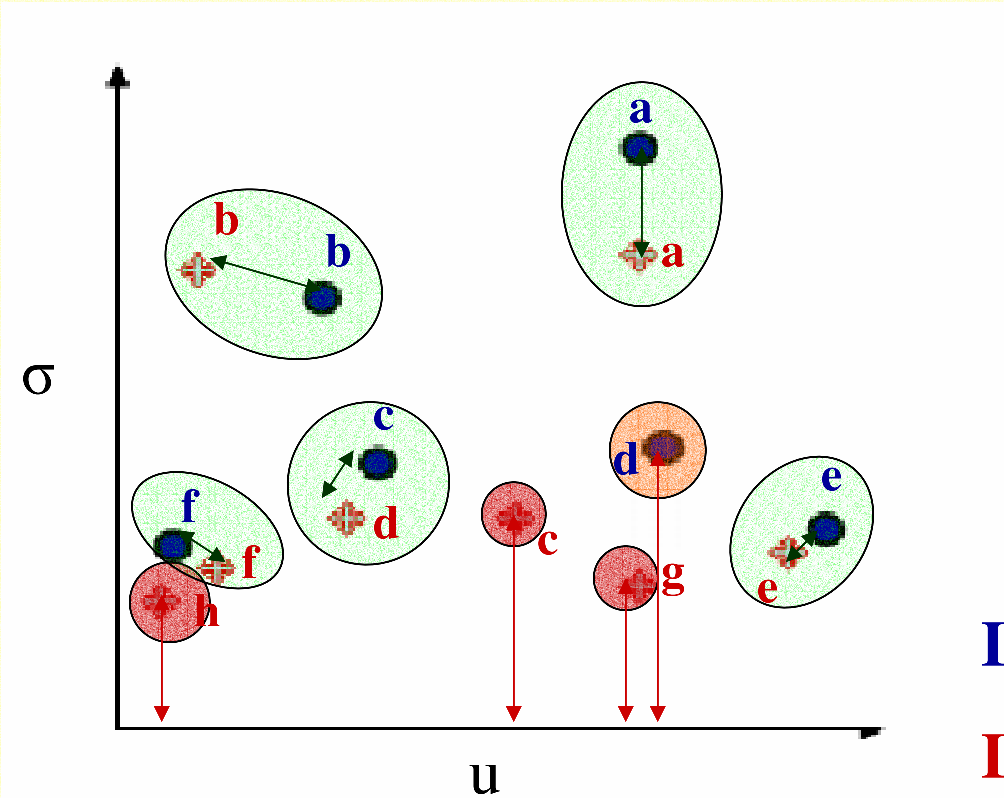
At certain level two points join and represent maximum of relevant contour

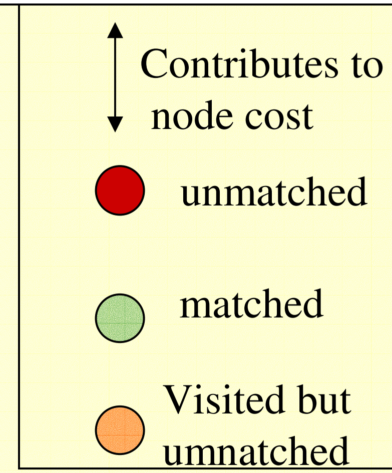
CSS Matching

- Features used for matching are maxima of curvature zero crossing contours
- Task is to find correspondence between two sets of maxima



Shifted





List 1: a b c d e f

List 2: a b d e f

- Maxima are said to be matched when their u coordinates lie within a reasonable distance (20%)

- For all ***matched maxima***, cost is the ***euclidean distance*** between the pair
- For all ***unmatched maxima***, cost is the ***vertical distance(σ)*** of the maxima

Motion Descriptors

- Motion Activity Descriptors
- Camera Motion Descriptors
- Motion Trajectory Descriptors
- Parametric Motion Descriptors

Motion Activity Descriptor

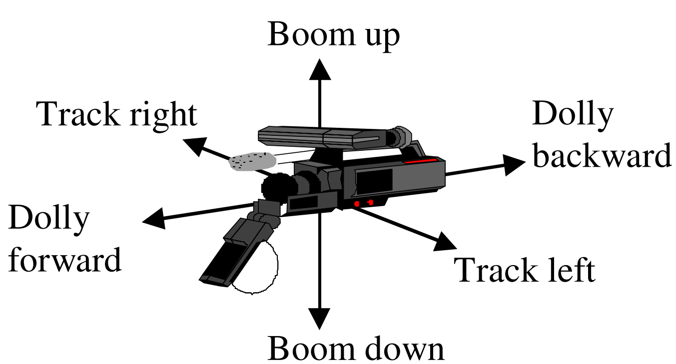
- Captures ‘*intensity of action*’ or ‘*pace of action*’
- Based on standard deviation of motion vector magnitudes
- Quantized into a 3-bit integer

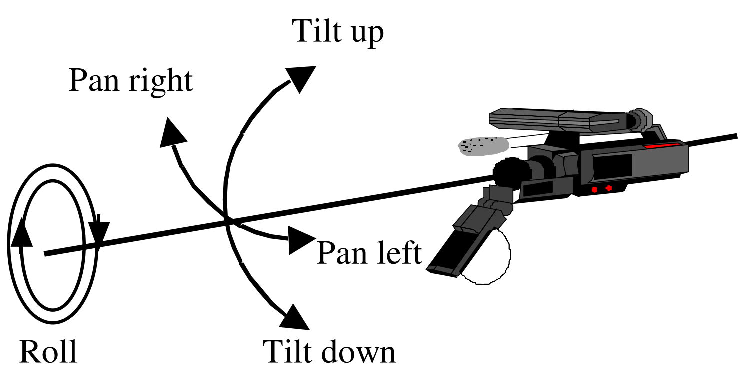
Parametric Motion

- Characterizes the evolution of regions over time
- Uses 2D geometric transforms

Camera Motion Descriptor

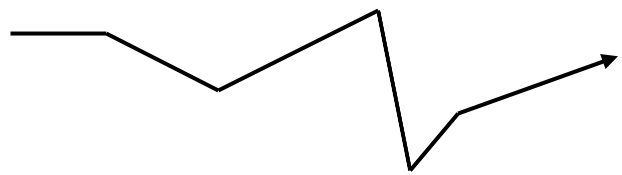
- Describes the movement of a camera or a virtual view point
- Supports 7 camera operations



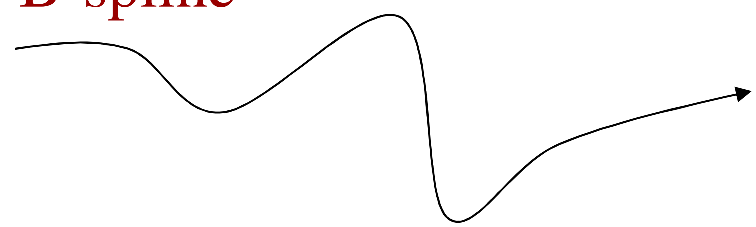


Motion trajectory representation

Point representation

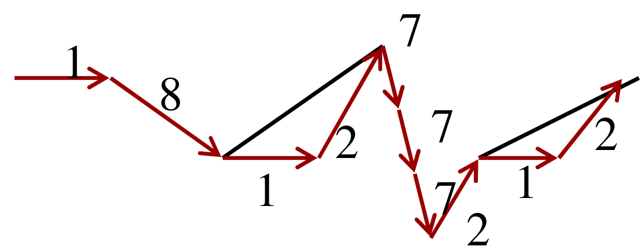


B-spline



Chain code

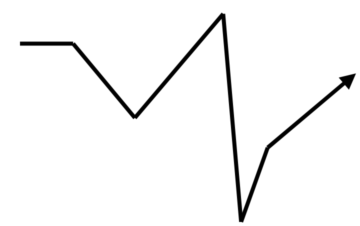
Primitive	→	↗	↑	↖	←	↙	↓	↘
Code	1	2	3	4	5	6	7	8



Code : 1812777212

Differential chain code

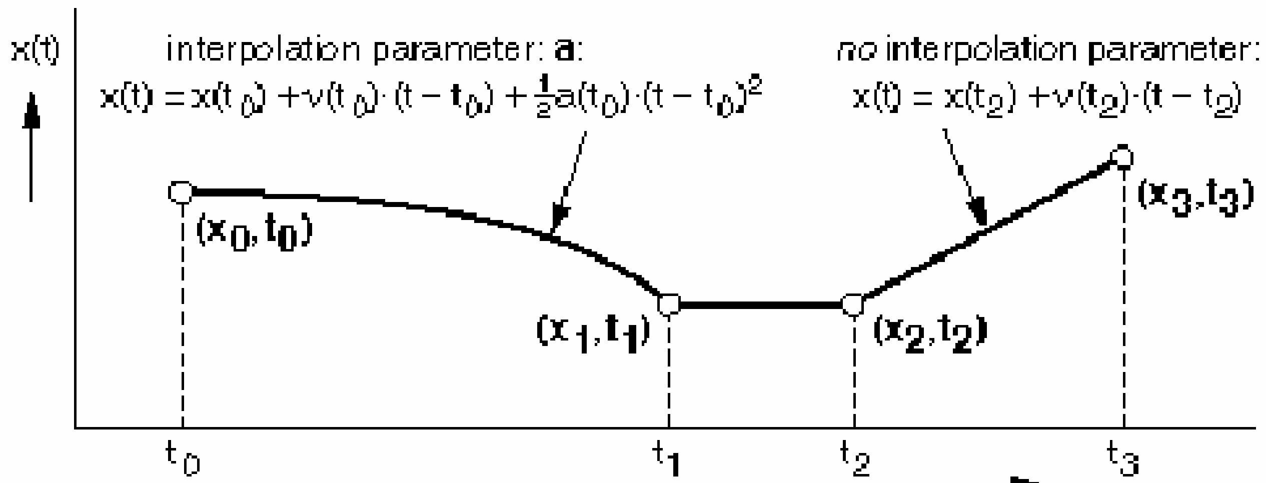
Relative condition	Right shorter	Right equal	Right longer	Left shorter	Left equal	Left longer
Code	1	2	3	4	5	6



Code: 36343

Motion Trajectory-MPEG-7

- Describes the movement of one representative point of a specific region
- A set of key-points (x, y, z, t)
- A set of interpolation functions describing the path



Trajectory Representation – Polynomial

- Motion trajectory model is a first- or second-order piecewise approximation along time, for each spatial dimension.

- Trajectory is divided into sub trajectories based on the key points (high curvature points) detected

- Polynomial curve is fitted for every sub trajectory

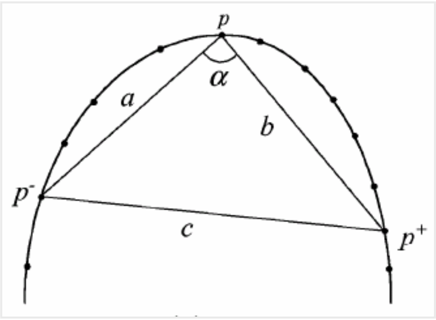
- The representation gives a good approximation of the trajectory

Trajectory Representation (Contd)…

Key point Detection

- Find high curvature points whose angle < 150^0

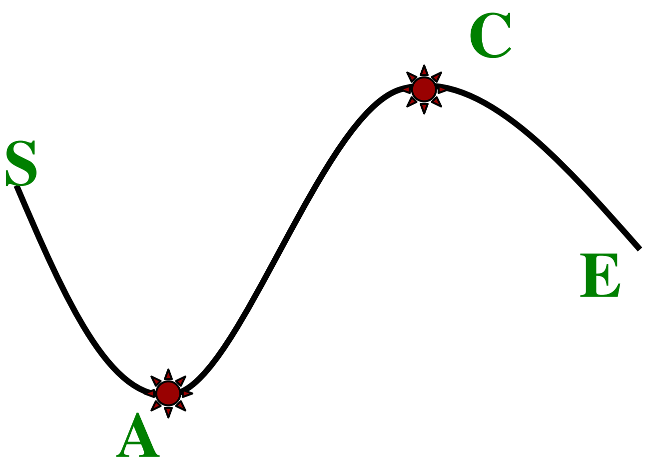
$$Angle\ \alpha(p) = \cos^{-1}\frac{\left\|p-p^{+}\right\|^{2}+\left\|p-p^{-}\right\|^{2}-\left\|p^{-}-p^{+}\right\|^{2}}{2\left\|p-p^{+}\right\|\left\|p-p^{+}\right\|}$$



$$d_{\min} \le \left|p-p^{+}\right| \le d_{\max} \qquad d_{\min} \le \left|p-p^{-}\right| \le d_{\max}$$

Key point labeling

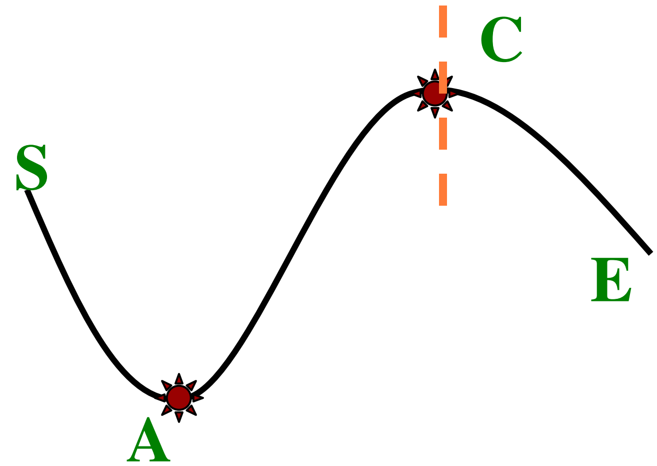
$$L(p) = \begin{cases} C\ (\text{Clockwise}),\ \text{if}\ \overrightarrow{k^{-}k} \otimes \overrightarrow{kk^{+}} > 0 \\ A\ (\text{Anticlockwise}), \quad \text{otherwise} \end{cases}$$



k→ Key points

Trajectory Segmentation

- Breaking between two segments is detected when A followed by C or vice versa

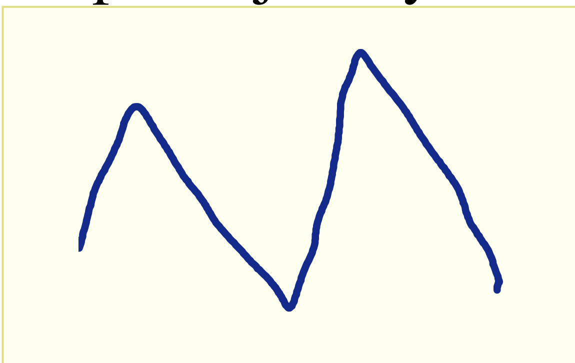


Polynomial Curve fitting

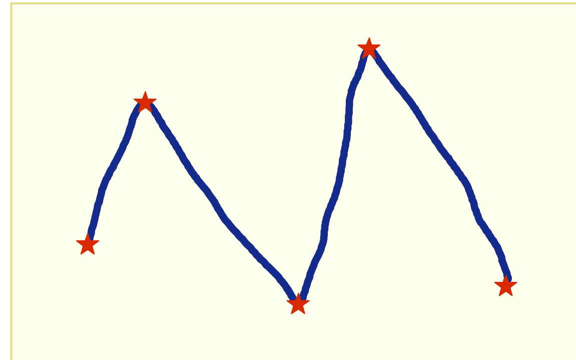
- Key points are used as Control points for Polynomial fitting
- If distance between any two control points is greater than a threshold, their middle point is chosen as a new control point
- A polynomial of second order is fitted through the control points for every sub trajectory

Trajectory representation

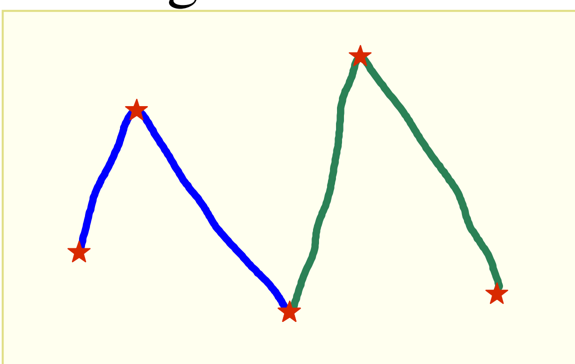
Input trajectory



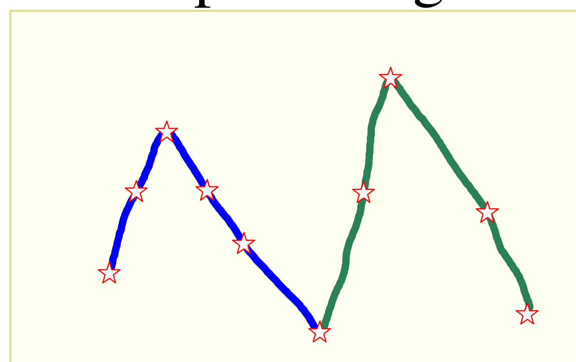
Key point detection



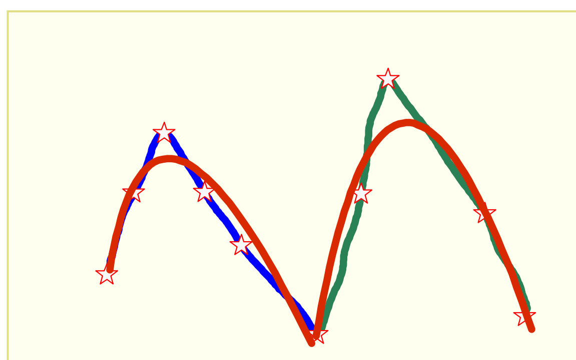
Segmentation



Control point Regeneration



Polynomial fitting



Trajectory matching

$$trajectory - \{(x(k), y(k))\}$$

$$velocity - \{(v^x(k), v^y(k))\}$$

Position difference

$$C_p = \frac{1}{N}\sum_{k=0}^{N-1}\left[\left(x_q(k) - x_m(k)\right)^2 + \left(y_q(k) - y_m(k)\right)^2\right]$$

Velocity difference

$$C_v = \frac{1}{N}\sum_{k=0}^{N-1}\left[\left(v_q^x(k) - v_m^x(k)\right)^2 + \left(v_q^y(k) - v_m^y(k)\right)^2\right]$$

Size difference

$$C_{sz} = \frac{2\left|size(q) - size(m)\right|}{size(q) + size(m)}$$

$$size(q) = \max(MaxX_q - MinX_q, MaxY_q - MinY_q)$$

$$C_t = wt_1 * C_p + wt_2 * C_v + wt_3 * C_{sz}$$

Video Search Engines (CBVR)

- VideoQ - Shape - Principal Components
  - Motion - Direct matching
- Netra-V- Shape – Fourier based descriptors
  - Motion – Affine motion Parameters
- JACOB - Color and Motion based
  - Motion magnitude and Direction histograms
- VIOLONE – Motion based – Chain code