# Affine Invariant Representation and Classification of Object Contours for Image and Video Retrieval

***Yannis Avrithis, Yiannis Xirouhakis and Stefanos Kollias***

**National Technical University of Athens**

**Department of Electrical and Computer Engineering**

# Problem Statement

- *Content-based retrieval* from image/video databases based on object shape (contour)
- Extraction of *video objects* based on color and motion segmentation and tracking
- *B-spline modeling* of object contours
- *Affine-invariant* curve representation using NFD, curve moments & novel curve normalization algorithm (AICN)
- Supervised classification of video objects into prototype object classes using *neural network*

# Applications

- Direct *content-based retrieval* based on object shape apart from other features (color, texture, motion etc.)
- *High level of abstraction* in the representation of video sequences using higher level classes as combinations of primary object classes
- *Generic object shape representation*, can be used with any curve matching algorithm
- Faster and more efficient *video queries*
- Multimedia database *management*

# Assumptions / Constraints

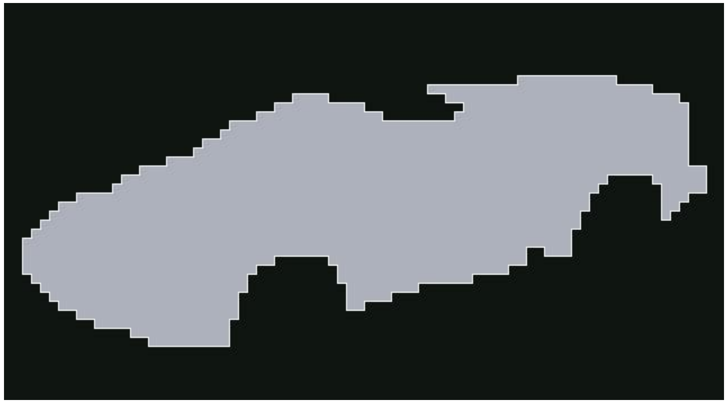
- High resolution images / video available
- *Main mobile objects* existing in foreground for good performance of motion segmentation algorithms
- Images of relatively *simple background* for good performance of color segmentation
- Relatively *planar objects* in foreground to ensure contour similarity for similar objects

# Video Object Extraction

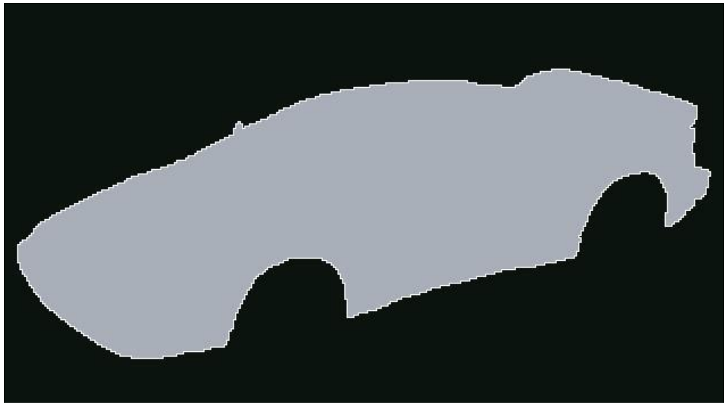
- *Shot cut* detection and partitioning
- *Unsupervised color and motion segmentation* and tracking using M-RSST algorithm, based on information available in MPEG video streams
- *Feature vectors* constructed using frame characteristics, such as number of segments, location, size and average color components & motion vectors
- Selection of the most *representative shots* and extraction of *key frames* for each shot

# Object Contour Extraction Results

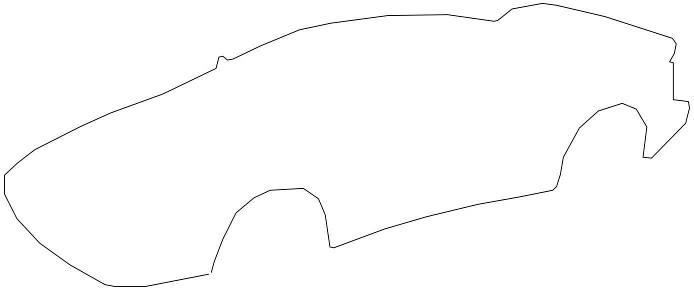
Original frame



First stage of segmentation



Final segmentation result



Object contour

# Curve Modeling using B-Splines (1)

- A dense set of $m$ data curve points $\mathbf{s}_j$ , $j = 0,..,m-1$ is given
- Input curve is modeled using closed *cubic B-splines* consisting of $n+1$ connected curve segments $\mathbf{r}_i$, $i = 0,1,..,n$
- Each segment is a linear combination of four *cubic polynomials* in the parameter $t \in [0,1]$:

$$\mathbf{r}_i(t) = \mathbf{C}_{i-1}Q_0(t) + \mathbf{C}_iQ_1(t) + \mathbf{C}_{i+1}Q_2(t) + \mathbf{C}_{i+2}Q_3(t)$$

where $Q_k(t) = a_{k0}t^3 + a_{k1}t^2 + a_{k2}t + a_{k3}$ , $k = 0,1,2,3$

# Curve Modeling using B-Splines (2)

- *Basis functions* $Q_k$ (t) are determined using
  - continuity constraints in position, slope and curvature
  - invariance property to coordinate transformations
- Modeled B-spline curve is given by

$$\mathbf{r}(t') = \sum_{k=0}^{n} \mathbf{r}_i (t' - i) = \sum_{k=0}^{n} \mathbf{C}_{i \bmod (n+1)} N_i(t')$$

where $0 \le t' \le n-2$ ,

and $N_i(t')$ denote the *blending functions*

$$N_i(t') = \begin{cases} Q_3(t' - i + 3) & i-3 \le t' < i-2 \\ Q_2(t' - i + 2) & i-2 \le t' < i-1 \\ Q_1(t' - i + 1) & i-1 \le t' < i \\ Q_0(t' - i) & i \le t' < i+1 \\ 0 & \textit{otherwise} \end{cases}$$

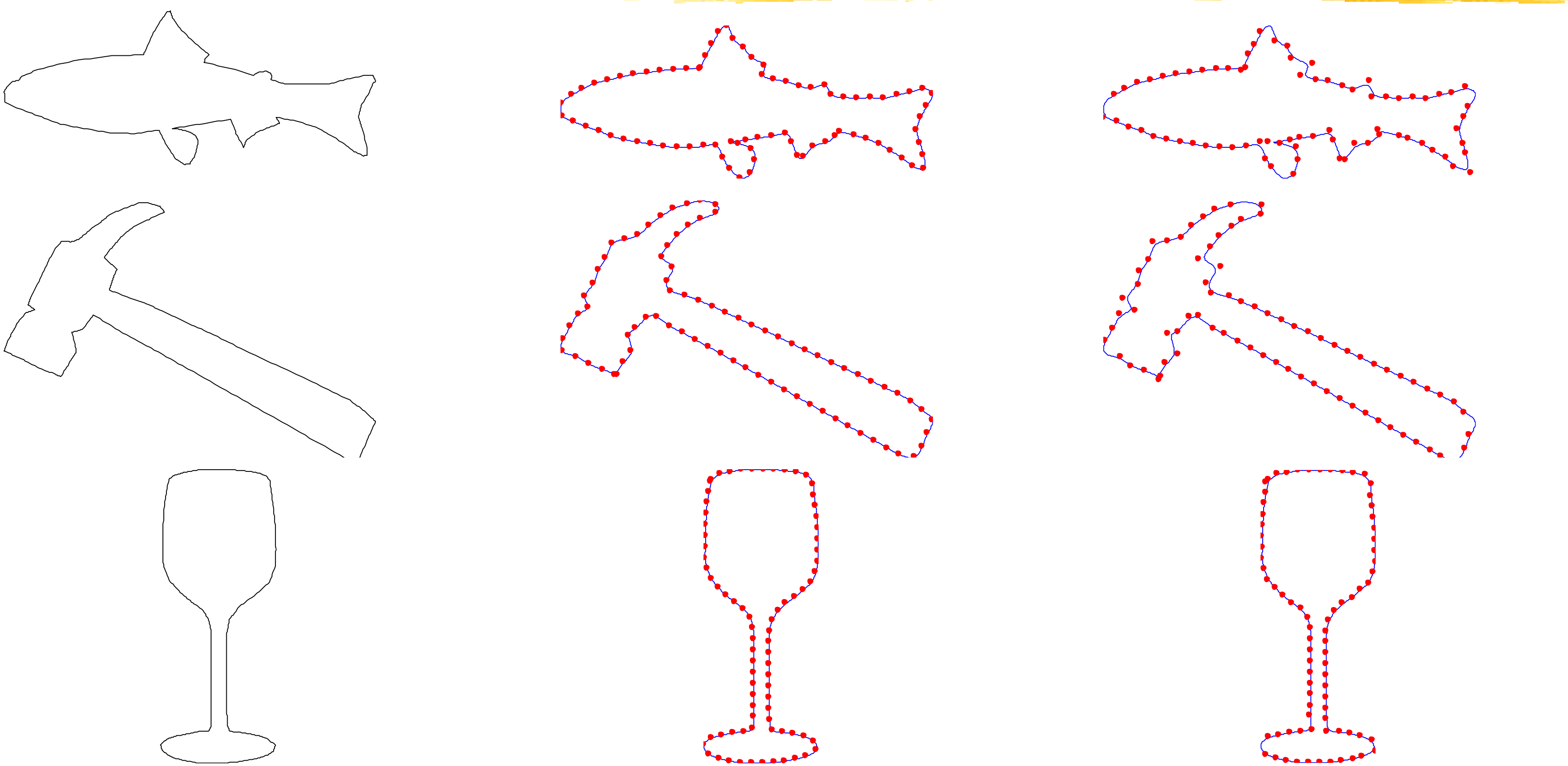
# Curve Modeling using B-Splines (3)

- *Control points* are determined so that the error between the observed data and the B-spline curve $d^2 = \sum_{j=1}^{m} \left\| \mathbf{s}_j - \mathbf{r}(t'_j) \right\|^2$ is minimized
- For appropriate parametric values of $t'$, MMSE solution is given in matrix form $\mathbf{C}_f = (\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T\mathbf{f}$, where **f** contains given data points
- Allocation of $t'$ values using *Chord Length* (CL) method, suffering from non-uniform noise / sampling, or *Inverse Chord Length* (ICL) method

# Curve Matching using Knot Points

- Control points cannot determine shape similarity, since different sets of control points may describe the same curve
- *Knot points* $\mathbf{p}_i$, $i$=0,1,...,$n$, derived using control points as $\mathbf{p}_f = \mathbf{A}\mathbf{C}_f$ , where A is a circulant matrix with [2/3,1/6,0,...,0,1/6] on its first row.
- Knot-points belong to the derived B-spline
- *Re-allocation* of knot points must be performed on each curve so that they are equal in number and that they correspond

# Knot Point Estimation Results



**Sample contours** | **B-splines with knot points** | **B-splines with control points**

# Normalized Fourier Descriptors (NFD)

- The sample data sequence $\mathbf{b}_k = \mathbf{s}_{xk} + j\,\mathbf{s}_{yk}$, $k=0,\ldots,m-1$, is formed and *discrete Fourier factors* are given by

$$F_i = \sum_{k=0}^{m-1} \mathbf{b}_k \cdot \exp\left(-\frac{j 2\pi \cdot i \cdot k}{m}\right), \quad i = 0,1,\ldots,m-1$$

- If $\mathbf{b}_{k'}$ is a sequence obtained from $b_k$ by scaling, translation, rotation and shift:

$$F_i' = a \cdot F_i \cdot \exp\left( j\,\frac{\vartheta - 2\pi \cdot i \cdot k_0}{m}\right) + \mathbf{b}_0 \cdot \delta(0)$$

- Normalized Fourier descriptors $\mathbf{v}_i = |F_i'| / |F_1'|$ are invariant to *translation, rotation & starting point*

# Curve Matching using Moments

- NFD provide a poor description of object contours: *Fine matching* is necessary
- Each spline is parametrized in terms of its arc lengths *s* as R(*s*)=[x(*s*), y(*s*)], and the (*p*,*q*)-*order moments* are estimated by
$$m(p,q)^{(j)} = \int_{s=0}^{S} x^p(s) \cdot y^q(s) \cdot w_j(x,y)\,ds$$
- Using appropriate kernels $w_j$ , *affine parameters* aligning two curves are estimated from their moments up to order two
- Curve matching is *time-consuming*; it can only be used to refine classification results

# Affine-Invariant Curve Normalization (1)

- A novel algorithm (AICN) is employed for removal of affine transformations without discarding shape information
- A series of linear transformations is applied to curves, with parameters estimated from 1st and 2nd order statistics of curve data
- Translation is removed by normalizing the gravity center of curve $\mathbf{s}_k$ , $k$=0,...,$m$ -1, to the axes origin, so that for the resulting data set:

$$\sum_{k=0}^{m-1} \mathbf{s}_{xk} = \sum_{k=0}^{m-1} \mathbf{s}_{yk} = 0$$

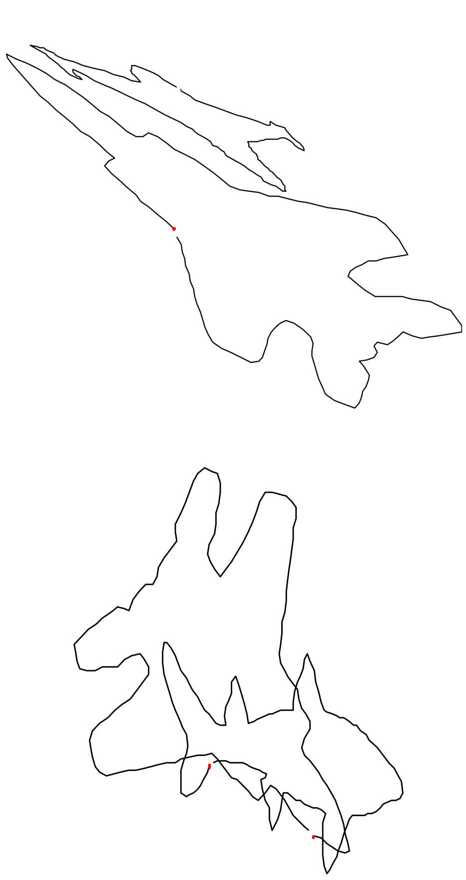
# Affine-Invariant Curve Normalization (2)

- Scale transformation is removed by 2 successive normalization steps (at 0° and 45°), so that

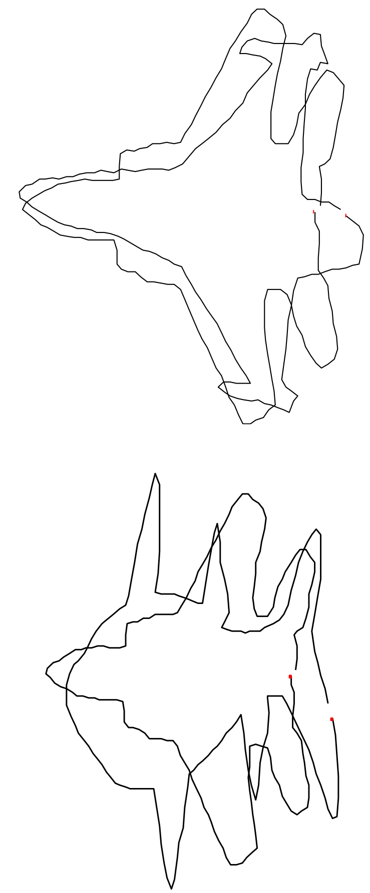
$$\sum_{k=0}^{m-1} \mathbf{s}_{xk}^2 = \sum_{k=0}^{m-1} \mathbf{s}_{yk}^2 = 1, \quad \sum_{k=0}^{m-1} \mathbf{s}_{xk} \mathbf{s}_{yk} = 0$$

- Starting point and rotation normalized so that first and last elements, $F_1$ and $F_{m-1}$, of the curve Fourier transform become real and positive
- AICN permits direct curve classification by any existing curve matching method, since it preserves all curve shape information

# Curve Normalization Results



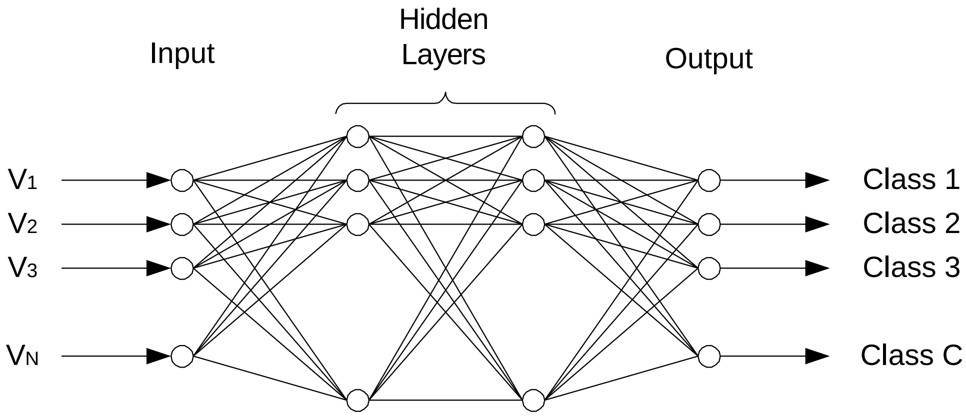
**Initial object contours**



**Normalized curves**

# Neural Network Classification (1)

- Definition of primary *object classes* (airplanes, cars, vases etc.) using groups of curve prototypes, organized in object class database
- NN classifier maps *input pattern* $\mathbf{v}=[v_1,v_2,\ldots,v_N]^T$ (NFD or normalized curve) to binary *output pattern* $\mathbf{d}=[d_1,d_2,\ldots,d_C]^T$:



# Neural Network Classification (2)

- In *training stage*, inputs $\mathbf{v}^{(p)}$, $p=1,\ldots,M$, corresponding to $M$ curve prototypes, are fed into the NN. *Desired outputs* $\mathbf{d}^{(p)}$, $p=1,\ldots,M$ are determined by setting one component of $\mathbf{d}^{(p)}$ equal to 1 and all others to 0
- *Levenberg-Marquardt* method used for training
- In *allocation* stage, $B$-spline representation $\mathbf{v}=[v_1,v_2,\ldots,v_N]^T$ of test curve is used as input to the NN. The *maximum* network output determines the corresponding object class

# NN Classification Results

| Object Class | Classification Results | |
|---|---|---|
| | **NFD** | **AICN** |
| Cars | 89.2% | 98.6% |
| Airplanes | 83.6% | 99.2% |
| Glasses | 76.1% | 94.9% |
| Spoons | 90.4% | 96.3% |
| Fish | 84.7% | 97.6% |
| **Total** | **84.8%** | **97.3%** |

# Conclusions - Further Work

- Direct *content-based retrieval* from video databases based on object shape apart from other features (color, texture, motion etc.)
- *Affine-invariant representation* of object contours can be used with any curve matching method
- Supervised classification of video objects into prototype object classes using *neural network*
- *High level of abstraction* in the representation of video sequences using higher level classes as combinations of primary object classes