Image Retrieval and Classification Using Affine Invariant B-Spline Representation and Neural Networks

Yiannis Xirouhakis, Yannis Avrithis and Stefanos Kollias

National Technical University of Athens

Department of Electrical and Computer Engineering

Problem Statement

- *Content-based image retrieval* from video databases based on object shape (contour)
- Extraction of *key-frames* that provide sufficient information about video content
- Extraction of *video objects* based on color and motion segmentation and tracking
- Affine-invariant *B-spline representation* of object contours
- Supervised classification of video objects into prototype object classes using *neural network*

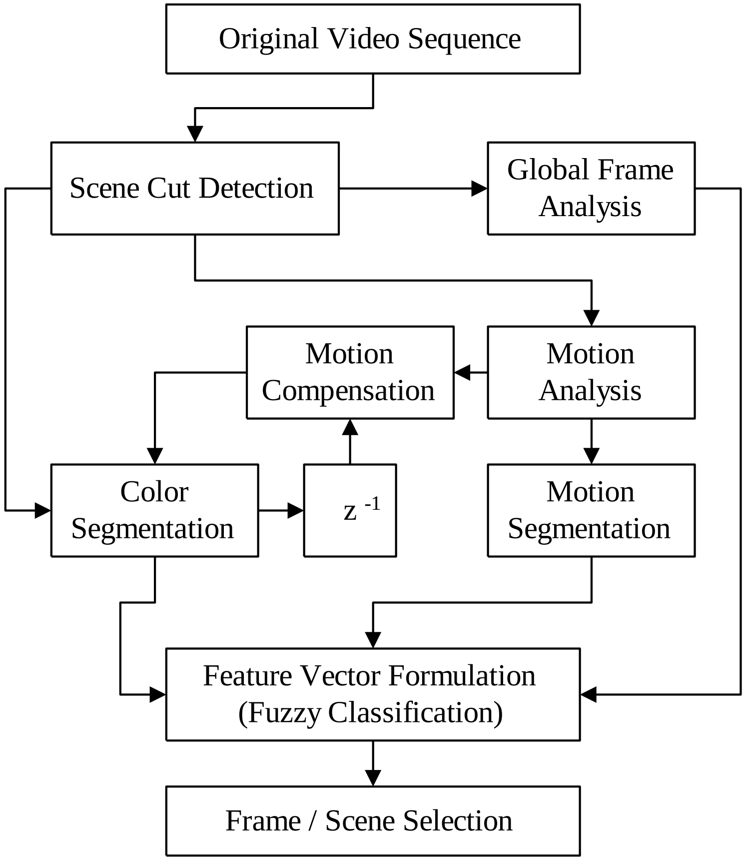
Applications

- 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
- Multimedia database *management*
- Reduction of *storage requirements* for search capabilities
- Faster and more efficient *video queries*

Assumptions / Constraints

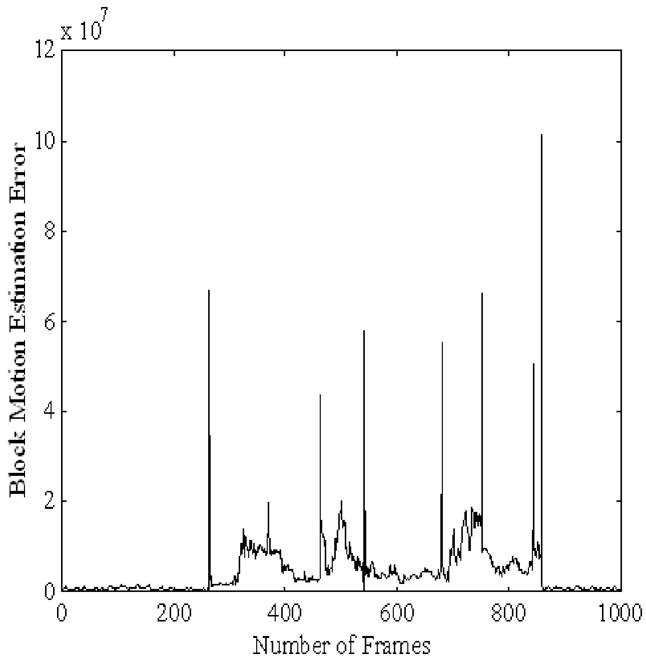
- High resolution images / video available
- Main mobile objects existing in foreground for good performance of motion segmentation algorithms
- Images of relatively simple background for good performance of color segmentation
- Relatively planar objects in foreground to ensure contour similarity for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

Video Processing



- Scene cut detection
- Feature extraction for each frame
- Formulation of scene feature vectors
- Selection of the most representative scenes
- Extraction of key frames for each scene

Scene Cut Detection

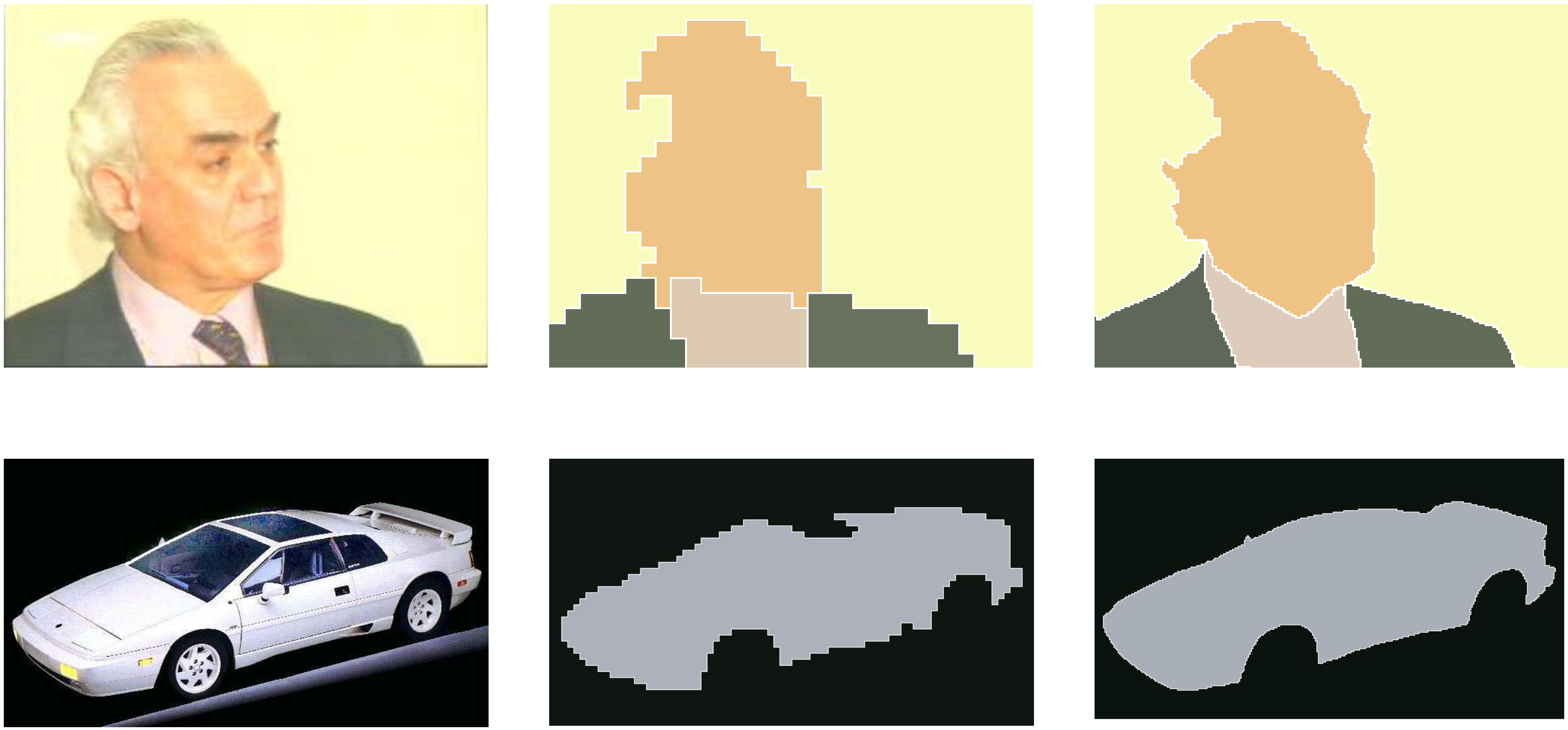


- Computation of the sum of the block motion estimation error
- Selection of frames for which sum exceeds a certain threshold
- Computations applied directly to MPEG - coded sequences

Color Segmentation

- Segmentation according to *spatial homogeneity*
- *Block resolution* (reduction of computational time, exploitation of MPEG information)
- *Hierarchical merging* of similar segments (w.r.t. color homogeneity & segment size)
- *Color features*: number of segments, location, size & mean color of each segment
- *Object tracking*: connected regions encouraged to remain connected in successive frames

Color Segmentation Results

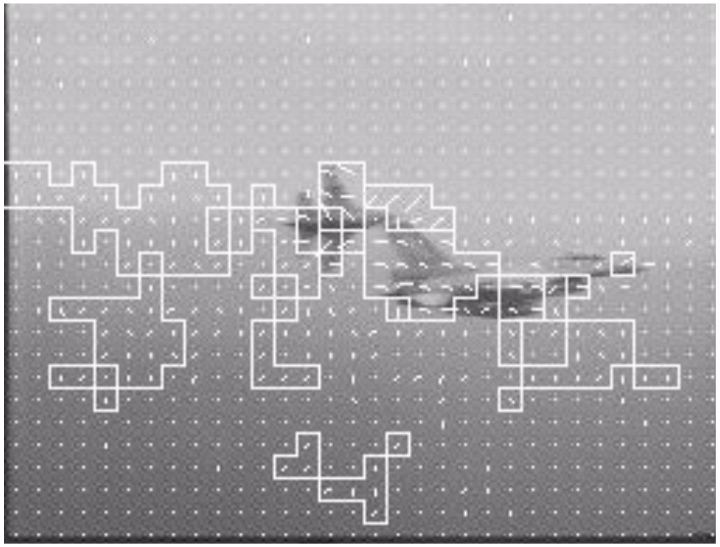


Two original frames | **First stage of segmentation** | **Final result (full resolution)**

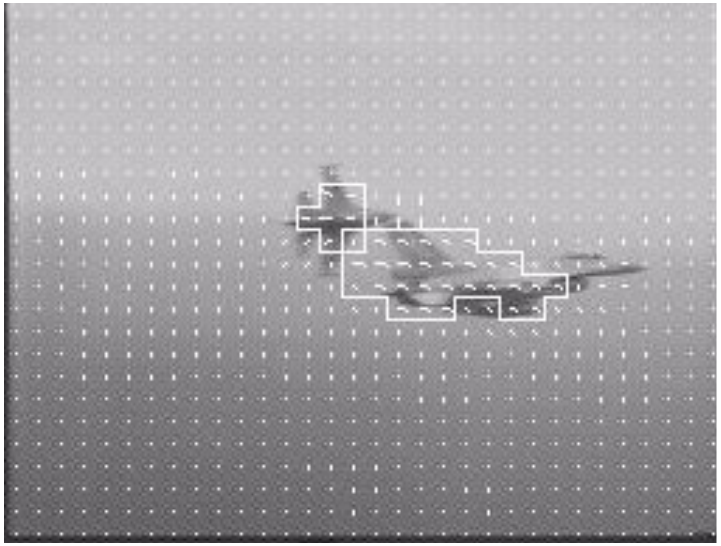
Motion Segmentation

- Segmentation according to *spatial homogeneity*
- *Block resolution* (reduction of computational time)
- Motion vectors derived from motion analysis, or directly from MPEG stream
- *Median filtering* of derived motion vectors: elimination of “noise”, preservation of “edges”
- *Motion features*: number of segments, location, size & mean motion vector of each segment

Motion Segmentation Results



Motion segmentation without smoothing



Motion segmentation with smoothing

Scene Selection Mechanism

- *Scene feature vector* constructed based on frame feature vectors over duration of scene
- *Clustering* of similar scene feature vectors $\mathbf{s}_i \in \Re^M$, $i=1,\ldots,N_S$ and selection of cluster representatives $\mathbf{c}_i$, $i=1,\ldots,K_S$
- Average *distortion* $D(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{K_S}) = \sum_{i=1}^{K_s} \sum_{\mathbf{s} \in Z_i} d(\mathbf{s}, \mathbf{c}_i)$ is minimized w.r.t. $\mathbf{c}_i$, $i=1,\ldots,K_S$
- *Generalized Lloyd* or *K-means* algorithm used as an optimization method

Key Frame Selection Mechanism

- Minimization of a *correlation criterion*: key frames should not be similar to each other
- *Correlation measure*

$$R(\mathbf{x}) = R(x_1,\ldots,x_{K_F}) = \left(\sum_{i=1}^{K_F-1} \sum_{j=i+1}^{K_F} (\rho_{x_i,x_j})^2 \right)^{1/2}$$

  of feature vectors $\mathbf{f}_i,\ i = x_1,\ldots,x_{K_F}$ is minimized w.r.t. index vector $\mathbf{x} = (x_1,\ldots,x_{K_F})$ corresponding to a set of selected frames
- Exhaustive search is unfeasible: minimization implemented by *logarithmic search algorithm*

Estimation of Curve Parameters

- Curve modeling using *cubic B-splines*
- Curve matching using *control* and *knot-points* for modeled curves
- Curve matching using *Fourier descriptors*
- Affine-invariant curve description and matching using *curve moments*

Curve Modeling using B-Splines (1)

- A dense set of *m* data curve points s_j, $j = 0,..,m-1$ is given
- Input curve is modeled using closed *cubic B-splines* consisting of $n+1$ connected curve segments 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
  - the 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*

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

Curve Modeling using B-Splines (3)

- *Control points* are determined, such 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 for the control points is given as

$\mathbf{C}_f = (\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T\mathbf{f}$ where $\mathbf{f}=[\mathbf{x},\mathbf{y}]$ and

$$\mathbf{P} = \begin{bmatrix} N_0(t'_1)+N_{n+1}(t'_1) & N_1(t'_1)+N_{n+2}(t'_1) & N_2(t'_1)+N_{n+3}(t'_1) & N_3(t'_1) & \cdots & N_n(t'_1) \\ N_0(t'_2)+N_{n+1}(t'_2) & N_1(t'_2)+N_{n+2}(t'_2) & N_2(t'_2)+N_{n+3}(t'_2) & N_3(t'_2) & \ldots & N_n(t'_2) \\ \vdots & & \vdots & & & \vdots \\ N_0(t'_m)+N_{n+1}(t'_m) & N_1(t'_m)+N_{n+2}(t'_m) & N_2(t'_m)+N_{n+3}(t'_m) & N_3(t'_m) & \cdots & N_n(t'_m) \end{bmatrix}$$

Curve Modeling using B-Splines (4)

- Allocation of t' values using the *Chord Length* (CL) method with $t'_1 = 0$, $t'_{max} = n - 2$ and

$$t'_j = t'_{j-1} + t'_{max} \cdot \left\| \mathbf{s}_j - \mathbf{s}_{j-1} \right\| \cdot \left(\sum_{l=2}^{m} \left\| \mathbf{s}_l - \mathbf{s}_{l-1} \right\| \right)^{-1}, \quad j = 2, \ldots, m$$

- Implies that the chord length is a very close approximation to the arc length, assuming constant speed of a particle onto the curve
- CL method suffers from non-uniform noise and non-uniform sampling. Alternatively, the inverse chord length method (ICL) can be used.

Curve Matching using Knot-Points (1)

- A set of M different curves (sets of samples) is modeled using M cubic B-splines
- Control points cannot determine shape similarity, since different sets of control points may describe the same curve
- For each curve we derive its *knot-points* p_i, $i=0,1,\ldots,n$, using its control points as $\mathbf{p}_f = \mathbf{AC}_f$ where A is the circulant matrix with [2/3,1/6,0,…,0,1/6] on its first row.
- Knot-points belong to the derived B-spline

Curve Matching using Knot-Points (2)

- *Re-allocation* of knot-points must be performed on each curve so that they are equal in number (l) and that they correspond
- The first knot-point is placed where the curve intersects the x-axis with its centroid on (0,0)
- The rest l-1 knot-points are placed equally spaced onto each curve
- The classifier based on the re-allocated knot-points is based on minimizing $d^2 = \sum_{i=1}^{l} \left\| \mathbf{p}_i^{(a)} - \mathbf{p}_i^{(b)} \right\|^2$ where *a*, *b* denote splines subject to comparison

Curve Matching using F.D. (1)

- At this point 2 major problems arise:
  - the comparison and classification of curves must be invariant to possible affine transformations
  - a rapid initial classification is demanded, to not compare a sample curve to all prototype curves
- These problems are addressed using *Fourier descriptors (F.D.)*, *curve moments* and *NN*
- For each sample s_k, $k=0,\ldots,m-1$, the sequence $b_k=s_{xk}+ j\, s_{yk}$ is obtained and discrete Fourier factors are given by

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

Curve Matching using F.D. (2)

- If $b_{k'}$ 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* and *starting point*
- Normalized Fourier descriptors are fed into a NN, so only the estimated knot-points are used, for reasonably small number of NN inputs

Curve Matching using Moments (1)

- Although Fourier descriptors possess desirable properties, they are poor description for the contour curve of an object.
- For this reason, they are used only as an 'initial description'.
- After assigning a class to each input sample curve, fine match is performed using *curve moments*
- Each spline is parametrized in terms of its arc lengths s as $R(s)=[x(s), y(s)]$

Curve Matching using Moments (2)

- 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* L, c aligning two curves, $\mathbf{r}(t')^{(a)} = \mathbf{L} \cdot \mathbf{r}(t')^{(b)} + \mathbf{c}$, are estimated from their moments up to order two, solving two second degree polynomials
- For each of *M* modeled B-splines, curve moments are computed and stored

Curve Matching using Moments (3)

- For any B-spline corresponding to an input sample curve, moments are computed and possible L, c are obtained for all *M* curves
- Input curve is subjected to the estimated affine transformation before comparison to the corresponding prototype
- *Knot-point classification* is then performed
- Curve moments, although superior to Fourier descriptors, is *time-consuming*; so it is used only to refine the results obtained from the NN.

Object Classification

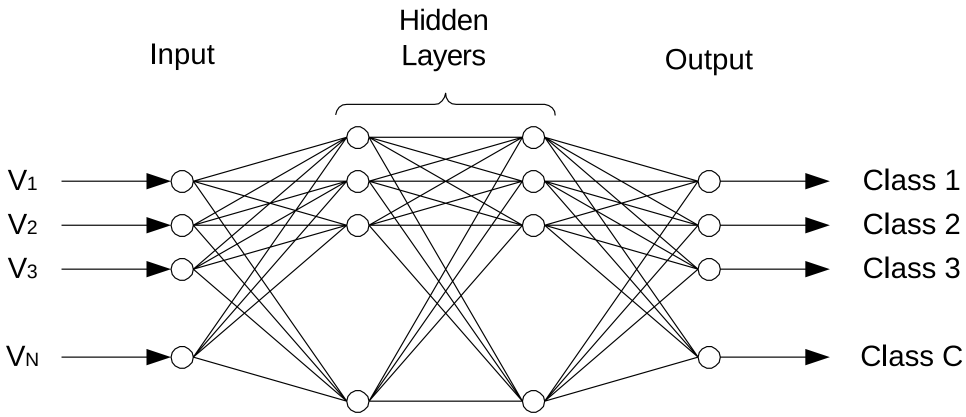
- Definition of primary *object classes* (airplanes, cars, vases etc.) using groups of curve prototypes, organized in object class database
- Each class contains several *prototypes* depicting different object instances or variations, different views or views in different level of detail
- Direct comparison of sample curves with all prototypes through curve matching extremely time consuming: *neural network (NN)* used at first stage of classification

Neural Network Training

- Normalized Fourier descriptors $\mathbf{v}=[v_1,v_2,\ldots,v_N]^T$ used as input to feedforward neural network
- NN attempts to map *input pattern* $\mathbf{v}$ to desired *output pattern* $\mathbf{d}=[d_1,d_2,\ldots,d_C]^T$
- In *training stage*, inputs $\mathbf{v}^{(p)}$, $p=1,\ldots,M$, corresponding to a set of 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

Neural Network Architecture

- Two *hidden layers*, with N input neurons, N_1 and N_2 neurons in the 1st and 2nd hidden layer, and C neurons in the output layer:

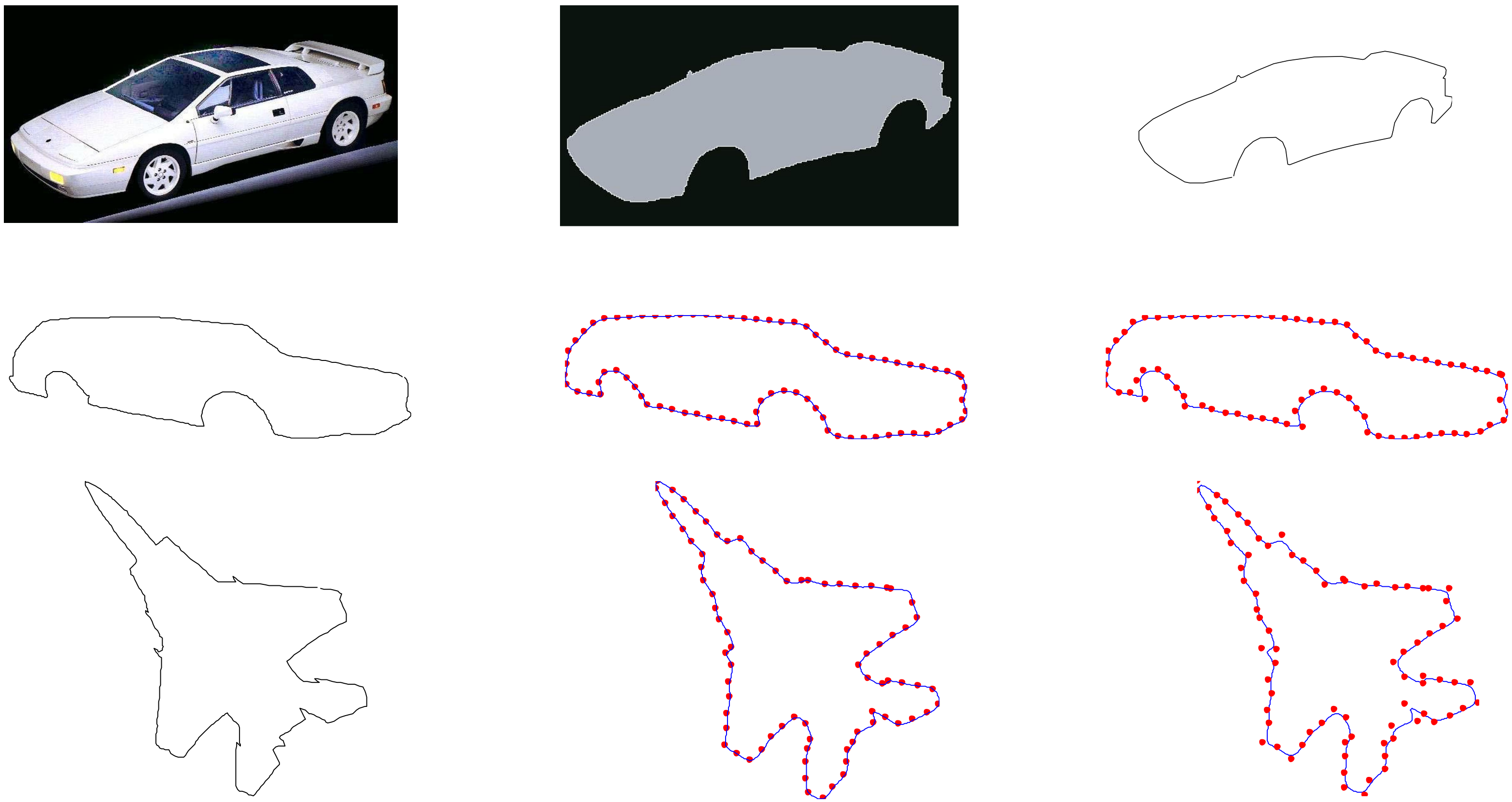


- *Levenberg-Marquardt* method used for training, attempting to minimize the sum-squared error between desired and actual output patterns

Neural Network Classification

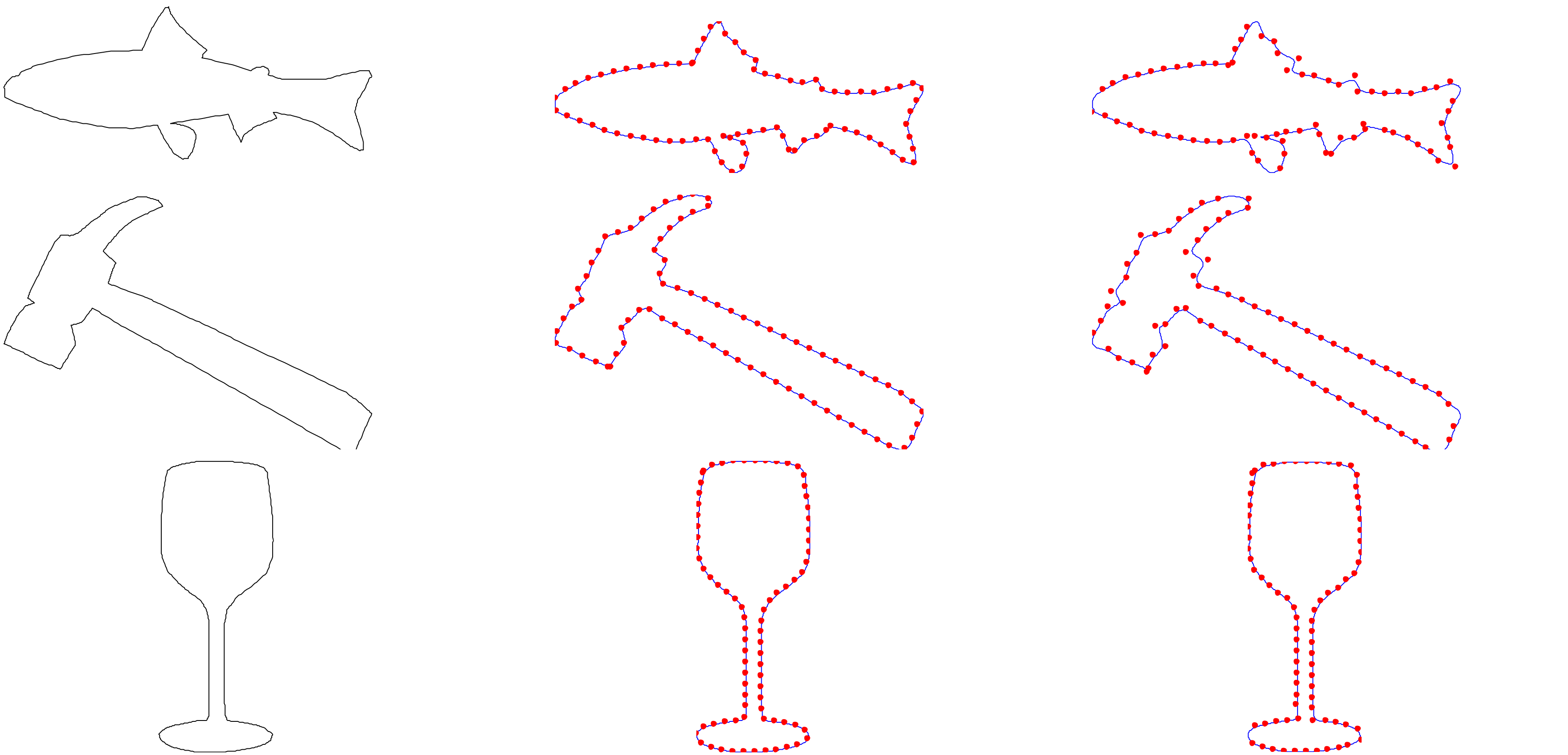
- In *allocation* stage, the *B*-spline representation $\mathbf{v}=[v_1,v_2,\ldots,v_N]^T$ of a test curve is used as input to the NN
- The input curve is typically classified to the object class that corresponds to the *maximum* network output
- In order to avoid misclassification, *R* classes are selected, corresponding to the network outputs with the maximum values. Final classification obtained through *curve matching*

Knot Point Estimation Results (1)



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

Knot Point Estimation Results (2)

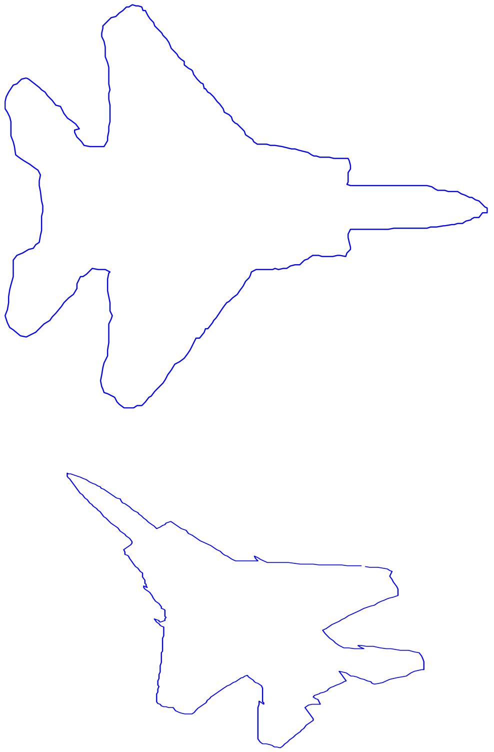


Sample contours

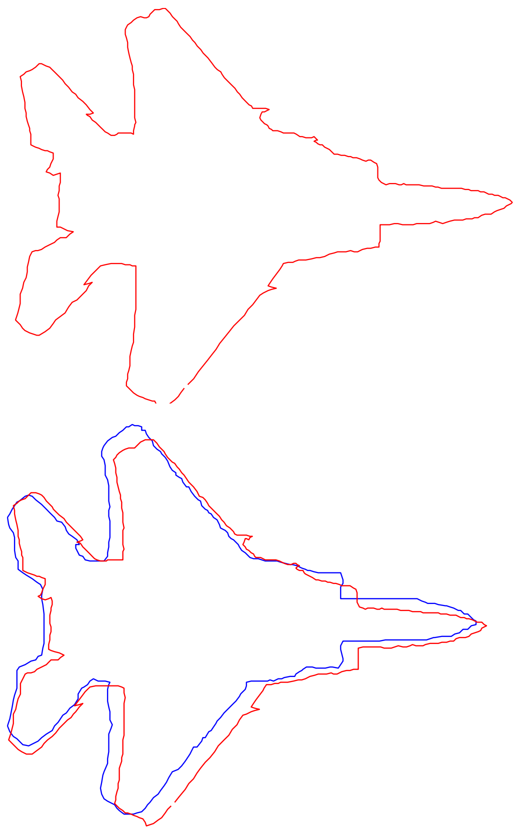
B-splines with knot points

B-splines with control points

Matching Results using Moments

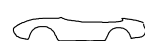
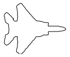
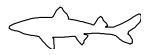
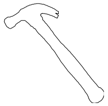
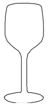


Sample contours

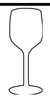


Curve Matching

NN Classification Results

		Sample Curves									
											
Object Class		**0.47**	**0.83**	0.00	0.00	0.04	0.00	0.00	0.01	0.00	0.01
		0.00	0.00	**1.00**	**0.72**	0.01	0.01	0.00	0.00	0.01	0.00
		0.05	0.03	**0.49**	**0.08**	**0.84**	**0.97**	0.00	0.01	0.00	**0.05**
		0.00	0.02	0.01	0.01	0.01	0.00	**0.95**	**0.68**	**0.04**	0.00
		**0.53**	**0.12**	0.00	0.01	**0.91**	**0.29**	**0.74**	**0.62**	**0.99**	**0.98**

NN and Curve Matching Results

Object Class	NN Classification (1/2 correct)	Curve Matching (1 correct)
	92 %	95 %
	98 %	100 %
	89 %	96 %
	95 %	98 %
	99 %	100 %
Total	**94.6 %**	**97.8 %**

Conclusions - Further Work

- Direct *content-based retrieval* from video databases based on object shape apart from other features (color, texture, motion etc.)
- Affine-invariant *B-spline representation* of object contours
- 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