Machine Vision and
Applications
© Springer-Verlag 2001

Affine-invariant curve normalization for object shape representation, classification, and retrieval

Yannis Avrithis, Yiannis Xirouhakis, Stefanos Kollias

Image, Video and Multimedia Systems Laboratory, Electrical and Computer Engineering Department, National Technical University of Athens, 9, Iroon Polytechniou Str., Zografou 157 73, Athens, Greece; e-mail: {iavr,jxiro}@image.ntua.gr, Tel.: +30-1-7722521, Fax: +30-1-7722492

Received: 29 April 2000 / Accepted: 21 February 2001

Abstract. A novel method for two-dimensional curve normalization with respect to affine transformations is presented in this paper, which allows an affine-invariant curve representation to be obtained without any actual loss of information on the original curve. It can be applied as a preprocessing step to any shape representation, classification, recognition, or retrieval technique, since it effectively decouples the problem of affine-invariant description from feature extraction and pattern matching. Curves estimated from object contours are first modeled by cubic B-splines and then normalized in several steps in order to eliminate translation, scaling, skew, starting point, rotation, and reflection transformations, based on a combination of curve features including moments and Fourier descriptors.

Key words: Curve normalization – Affine invariants – Shape analysis – Image and video retrieval

1 Introduction

The recent growth of interest in multimedia applications has led to an increasing demand for efficient storage, management, and browsing of multimedia databases. Browsing has been given considerable attention after the guidelines of the Moving Pictures Expert Group regarding the MPEG-4 and MPEG-7 standards (ISO 1997, 1998; Sikora 1997). Content-based query, indexing, and retrieval capabilities are of major importance in browsing digital image and video databases, due to the large amount of information involved (Avrithis et al. 1998; Doulamis et al. 1999; Xirouhakis et al. 1999b). Several prototype systems have been implemented to provide content-based image query and retrieval capabilities, including VIRAGE (Hamrapur et al. 1997), QBIC (Flickner et al. 1995), Photobook (Pentland et al. 1996), VisualSEEk (Smith et al. 1996), Netra (Ma and Manjunath 1997), MARS (Rui et al. 1997), VideoQ (Chang et al. 1998), Excalibur, CIIR, and C-BIRD. Some of these systems are already at the stage of commercial exploitation, while the current trend is to extend retrieval capabilities to video sequences separately from still images.

Content information in content-based retrieval systems is usually modeled in terms of low-level features such as color and texture composition (Avrithis et al. 1998), motion field and depth maps (Doulamis et al. 2000), as well as shape attributes (Xirouhakis et al. 1998). Higher-level attributes such as semantic objects can be obtained by appropriate fusion of low-level features, especially in the context of specific applications. As claimed in Persoon and Fu (1986), if the main information for description or classification of an object can be found from its boundary (contour) shape, it is natural to retain only the boundary for further analysis. Such situations may arise, for example, in the classification of silhouettes of airplanes or satellites, in character recognition, and in document processing (Jiang et al. 1999). However, the study of shape for the purpose of *general* object classification, recognition, or retrieval, either by itself or in combination with other object features, is an active field of current research (Swanson and Tewfik 1997; Yang and Cohen 1999).

There are two main reasons for this increased interest in shape analysis. First, object shape can provide a very powerful tool for visual image retrieval, by means of a *query-by-sketch* mechanism (Bimbo and Pala 1997), where user-sketched templates are used for similarity matching over object shapes in an image or video database. The process of matching is performed between prototype templates and the rough sketch of the desired object provided by the user. Second, content-based functionalities will be embedded in new multimedia coding standards (ISO 1998). Towards this goal, *second-generation* coding techniques have been proposed (Torres and Kunt 1996), where video coding is based on segmentation and allows content-based object manipulation (Salembier et al. 1997). Thus, shape information is included in video object planes in the form of binary image sequences, and can be used for the prediction of image partitions with applications to partition interpolation or extrapolation (Marques et al. 1998).

Several methods have been proposed in the literature for shape analysis, modeling, and representation, ranging from chain coding (Freeman 1970) to polygonal approximation (Pavlidis and Ali 1975), medial axis transform (skeleton)

Correspondence to: Y. Avrithis

(Blum 1967), Fourier descriptors (Persoon and Fu 1986), curve moments (Hu 1962), B-splines (Cohen et al. 1995), curvature scale spaces (Mokhtarian 1995), interest points (Wang et al. 1998), sinusoidal transform (Pratt 1996), and Legendre descriptors and Zernike moments (Khotanzad and Hong 1990). Basically, most approaches exploit geometric features of curves, either global (e.g., moments, length, principal axes, elongation, or compactness) or local (e.g., interest points, curvature measures, or implicit polynomials) in order to achieve shape matching, recognition, or classification. Whatever the application, all shape analysis methods share a common problem: object shapes can change drastically as the point of view changes due to perspective transformation. Most studies have approximated the viewpoint change by an *affine transformation*, which is a pretty good approximation when the object is far from the camera, since the slight distortion that may result from the more general projection can be regarded as part of a deformation. In order to avoid storing or matching several "prototype" shapes corresponding to different affine transformations (e.g., different rotation, translation, or scaling) one has to define *affine invariants*, i.e., properties that remain constant under arbitrary affine transformations.

One property that most affine-invariant techniques in the literature have in common is that invariance is "embedded" in the process of matching, recognition, or similarity measure estimation. For example, a similarity metric invariant to rotation, translation, and scaling based on turning functions for comparing polygonal shapes has been proposed in Arkin et al. (1991), and a similarity distance based on modified Fourier descriptors was introduced in Rui et al. (1998). For the same purpose, moment invariants are employed in Balslev (1998). Object recognition using normalized Fourier descriptors and neural networks has been presented in Wang and Cohen (1994), while genetic algorithms for affine-invariant shape recognition have been proposed in Tsang (1997). Techniques based on local curve features include local grayscale invariants based on automatically detected points of interest for image retrieval (Schmid and Mohr 1997), affine invariants based on convex hulls for image registration (Yang and Cohen 1999), and local deformation invariants for curve recognition based on implicit polynomials (Rivlin and Weiss 1995). Another approach is to match two given curves by optimally evaluating the affine parameters that maximize their similarity measure. This optimization is based, for example, on curve moments (Huang and Cohen 1996) or Fourier descriptors (Persoon and Fu 1986).

The main disadvantage of the first approach – embedding invariance in the matching or recognition process – is that in most cases some information about the original curve is lost. Meanwhile, the second approach – evaluating the affine parameters between two instances of an object – requires a priori knowledge of both instances, thus it can only be used for matching a particular pair of curves and not, for example, for recognition using a neural network or some other means of classification. Moreover, it usually involves a very high computational cost. To this end, the method of *normalization* has been recently introduced as an alternative to dealing with invariance. An image or curve can be normalized to a "standard" position, which is defined in such a way that all affine transformations of the same object are also normalized to the same position. Apart from the affine transformation parameters, to which the normalization is invariant, no other information is discarded; the normalization process consists in fact of an affine (linear) transformation and the shape of the original curve remains unchanged. A generalized normalization process for determining invariants is given in Rothe et al. (1996), image normalization is tackled in Shen and Ip (1997), while normalization of affinely distorted shapes is discussed in Taubin and Cooper (1992).

At the same time, a number of approaches have been proposed in the literature for curve matching under arbitrary deformations, which are based on deformable templates (Bimbo and Pala 1997). The deformable templates are obtained by imposing parametric transforms to the prototype curve, while the template curve variability is achieved in a probabilistic manner (Jain et al. 1996). Active contour models (snakes) are also appropriate in this sense (Lai and Chin 1995). Although deformable templates deal successfully with both image noise and local contour deformations (due to object local dissimilarities or even occlusion), local deformations can often be mixed with global changes due to rigid motion or affine transformations. In this sense, their performance is limited. However, improved results can be obtained when normalization is performed before employing deformable templates or active contour models, as pointed out in Ip and Shen (1998).

In the context of this paper, a novel method for two-dimensional (2-D) curve normalization with respect to affine transformations is presented, making it possible to obtain an affine-invariant curve representation without any actual loss of information on the original curve. In the case of closed contours, the representation is also invariant to the starting point. In particular, a 2-D closed curve representing the contour shape of an object is first modeled by a cubic B-spline so that the shape is simplified and segmentation noise is reduced, and uniform curve sampling in terms of arc length is obtained by estimating the B-spline knot points. Then the sampled curve is normalized in several steps in order to eliminate translation, scaling, skew, starting point, rotation, and reflection transformations. Normalization is based on a combination of curve features including moments and Fourier descriptors. All such features are globally estimated from all curve samples and no local information is used. The computational complexity involved is particularly low, so that the method can be easily integrated in a real-time system for image retrieval or video coding. Some of the main ideas of the approach followed in this paper have been introduced in Avrithis et al. (2000).

It is proved that each normalized curve *uniquely* corresponds to a set of curves that are related through an arbitrary affine transformation. Moreover, the normalized representation, together with the estimated affine parameters that relate the original curve with the normalized one, *completely* describe the original curve, since the latter can then be reconstructed exactly. Based on the above completeness and uniqueness properties, the curve is decomposed into global, affine-transformation-related position, and local shape information. Consequently, the proposed method can be applied as a preprocessing step to any shape representation, classification, recognition, or retrieval technique, since it efficiently decouples the problem of affine-invariant description from feature extraction and pattern matching. Several well-known curve similarity measures are employed to demonstrate the ability of the proposed representation to maintain all curve information except for arbitrary affine transformations, to which it is invariant. In all cases, it is experimentally shown to be considerably robust to shape deformations and noise.

The paper is organized as follows. Section 2 introduces the details of the environment where the proposed method can be applied, the assumptions made, and the known limitations. Section 3 presents the procedure of B-spline curve modeling and the extraction of uniform curve samples by means of knot points. Section 4 describes the curve orthogonalization procedure employed to eliminate translation, scaling, and skew transformations, while the remaining normalization with respect to starting point, rotation, and reflection is provided in Sect. 5. Finally, experiments on several real-life and simulated images and video sequences are presented in Sect. 6 to evaluate the performance, efficiency, and robustness of the proposed curve normalization, and conclusions are drawn in Sect. 7.

2 Problem statement

In the following, it is assumed that the contour shape of an object is available and represented by a set of ordered points forming a 2-D, planar, and closed curve. This set of sampled points is obtained from image data by means of manual or automatic segmentation. In practice, any segmentation algorithm can be applied, based for example on color or motion homogeneity, edge detection, or morphological tools (Salembier and Pardas 1994). The M-RSST algorithm (Avrithis et al. 1999) – a multiresolution implementation of the recursive shortest spanning tree algorithm – was used in our experiments for color segmentation, as described in Sect. 6. In the case of video sequences included in a video database, each sequence is first partitioned into video shots corresponding to a continuous action of a single camera operation, and segmentation is applied to video frames. In this case, segmentation is enhanced by exploiting motion information. In particular, 2-D parametric motion models were utilized for motion segmentation (Tekalp 1995; Odobez and Bouthemy 1997) and main mobile object detection techniques (Xirouhakis et al. 1999a). Although the starting point normalization presented in Sect. 5 is applicable to closed curves only, the remaining normalization steps remain valid for open curves too.

It is further assumed that curves correspond to nonoverlapping object boundaries that are completely known. This may lead to problems related to object occlusion, which can only be overcome by means of local invariants, whereas global features are only employed in the proposed method. Unfortunately, no method is known in the literature for normalization that can retain all curve information and at the same time deal successfully with occlusion problems, although considerable efforts have been made – see for example (Huang and Cohen 1996). In cases where the occlusion is “small” enough, however, it can be treated as a small shape deformation and tackled in the subsequent process of curve matching.

Each input curve is supposed to be subjected to two kinds of transformations: parameter and coordinate transformations. Parameter transformations are due to the fact that curves are obtained through image segmentation, hence discretization of continuous objects is involved, leading to segmentation noise and nonuniform sampling in terms of arc length. Also, since object contours correspond to closed curves, a point moving along the contour generates curve coordinates that are actually periodic functions of the arc length, and so an arbitrary starting point may be selected for the description of a single period. Moreover, the curve orientation might be either clockwise or counterclockwise. On the other hand, coordinate transformations are due to the fact that images are produced by projection of three-dimensional (3-D) objects onto a 2-D plane, leading to nonlinear perspective transformation. Assuming that an object is far enough from the camera, this could be approximated by a linear affine transformation. The problem is to normalize a curve and extract a representation that is invariant to both parameter and coordinate transformations, and yet maintains all the remaining curve information.

Parameter transformations due to nonuniform sampling and segmentation noise are tackled by means of a B-spline curve model, as described in Sect. 3. Some information is actually lost during this procedure, since this is necessary for noise removal and curve simplification, but the remaining normalization steps are completely reversible. Coordinate (affine) transformations are decomposed into translation, skewing, scaling, rotation, and reflection. A curve orthogonalization procedure is proposed for the elimination of translation, skew, and scaling transformations based on curve moments. Then, a normalization procedure based on Fourier descriptors is followed for elimination of starting point, rotation, reflection, and orientation.

3 B-spline representation

3.1 Curve modeling

B-splines have been widely employed for shape analysis and modeling, since they possess a number of important properties such as smoothness and continuity, built-in boundedness, local controllability, and shape invariance under affine transformation (Cohen et al. 1995). In this work, B-splines have been employed in order to obtain a smooth and continuous representation of curves, available as dense sets of data points, which in turn are generally obtained using nonuniform sampling. In our case, such data sets are provided by a motion and/or color segmentation scheme. In the rest of this section we cover some of the common background in fitting B-splines to data sets; for more details, see for example Cohen et al. (1995).

Cubic B-splines are composite curves consisting of a large generally number of curve segments with C^2 continuity on the connection points. Generally, a kth order B-spline is C^{k-1} continuous. We will hereon consider the case of closed cubic B-splines. Assuming that a closed cubic B-spline consists of $n+1$ connected curve segments $\mathbf{r}_i$, $i = 0, \ldots, n$, with $\mathbf{r}_i(u) = (x_i(u), y_i(u))$, each of these segments is a linear combination of four cubic polynomials $Q_k(u)$,

$k = 0, 1, 2, 3$ (commonly known as basis functions) in the parameter $u \in [0, 1]$:

$$\mathbf{r}_i(u) = \mathbf{C}_{i-1}\, Q_0(u) + \mathbf{C}_i\, Q_1(u) + \mathbf{C}_{i+1}\, Q_2(u) + \mathbf{C}_{i+2}\, Q_3(u), \qquad i = 0, \ldots, n \tag{1}$$

It can be seen that the whole B-spline $\mathbf{r}$, consisting of $n+1$ connected curve segments $\mathbf{r}_i$, is characterized by $n+1$ parameters, namely the control points $\mathbf{C}_i$. A parameterization of the whole curve is essential to the description of the B-spline, considering that the variable $u' \in [0, n+1]$. Then, for the ith segment, $u' = u + i$, where u is defined in [0,1]. The B-spline curve is then given on the basis of the curve segments as:

$$\mathbf{r}(u') \equiv \sum_{i=0}^{n} \mathbf{r}_i(u) = \sum_{i=0}^{n} \mathbf{r}_i(u' - i) \tag{2}$$

where $\mathbf{r}_i(u' - i)$ is nonzero for u in [0,1] or equally u' in $[i, i+1]$. Using (1), (2) can be written in a more convenient form as:

$$\mathbf{r}(u') = \sum_{i=0}^{n+3} \mathbf{C}_i N_i(u') \tag{3}$$

where $\mathbf{C}_i$ are defined for $i = 0, \ldots, n$, and $\mathbf{C}_{-1} = \mathbf{C}_n$, $\mathbf{C}_{n+1} = \mathbf{C}_0$, $\mathbf{C}_{n+2} = \mathbf{C}_1$, and $\mathbf{C}_{n+3} = \mathbf{C}_2$. By $N_i(u')$ we denote the so-called blending functions, which are simple functions of $Q_k(u)$ (Watt and Watt 1992). Along with the control points, the knot points are also defined as the connection points $\mathbf{p}_i$ between curve segments. Generally, $\mathbf{p}_i = \mathbf{r}_i(0) = \mathbf{r}_{i-1}(1)$. Given the control points, the knot points can be uniquely determined, assuming uniform placement of the knots, since:

$$\mathbf{p}_i = \frac{1}{6}\mathbf{C}_{i-1} + \frac{2}{3}\mathbf{C}_i + \frac{1}{6}\mathbf{C}_{i+1} \tag{4}$$

for $i = 0, \ldots, n$. In view of the above theoretical review, one can deduce that different pairs of control and knot points may define the same B-spine curve.

Once we are given a dense set of m data curve points $\mathbf{s}_j$, $j = 0, \ldots, m-1$, the control points $\mathbf{C}_i$ must be determined in order to fit an appropriate B-spline. The approach followed in this work tries to find an approximate B-spline such that the error between the observed data and their corresponding B-spline curve is minimized. In this sense, the metric

$$d^2 = \sum_{j=0}^{m-1} \left\| \mathbf{s}_j - \mathbf{r}(u'_j) \right\|^2 \tag{5}$$

where u'_j, $j = 0, \ldots, m-1$, are appropriate parametric values of u', should be minimized. If appropriate parametric values of u' were allocated on the curve, then the minimum mean-squared error solution for the control points would be given in matrix form as

$$\mathbf{C}_f = (\mathbf{P}^{\mathrm{T}}\mathbf{P})^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{f} \tag{6}$$

where $\mathbf{f}$ and $\mathbf{C}_f$ are of size $m \times 2$ and $(n+1) \times 2$, respectively, and contain the given data points $\mathbf{s}_j$ and the control points $\mathbf{C}_i$, respectively. The $m \times (n+1)$ matrix $\mathbf{P}$ contains appropriate values for the blending functions, estimated on the points $\mathbf{r}(u'_j)$ (Watt and Watt 1992). For the allocation of parametric values of u', the chord length (CL) method is employed. Specifically, for $u'_1 = 0$ and $u'_{\max} = n - 2$, u'_j associated with the sample point $\mathbf{s}_j$ is estimated by

$$u'_j = u'_{j-1} + u'_{\max} \cdot \frac{\|\mathbf{s}_j - \mathbf{s}_{j-1}\|}{\sum_{l=2}^{m} \|\mathbf{s}_l - \mathbf{s}_{l-1}\|} \tag{7}$$

where $j = 0, \ldots, m-1$.

The CL method is based on the fact that the chord length between any two points is a very close approximation to the arc length of the curve, and it assumes of constant speed of a particle onto the curve. It is robust to uniformly distributed noise, but suffers from nonuniform noise and nonuniform sampling. Alternatively, the inverse chord length method (ICL) could be used for robust results, as reported in Huang and Cohen (1996).

3.2 Knot point reallocation

Assume that a set of different curves (i.e., sets of sample points) is available in a database. After having modeled these sets of points with closed cubic B-splines, it can be seen that their control points cannot decide shape similarity between the curves, since generally different sets of control points may describe the same curve. For this reason, it is comfortable to derive for each curve the knot points $\mathbf{p}_i$, $i = 0, 1, \ldots, n$, using the estimated control points. For closed cubic B-splines, this is achieved from (4), or in matrix form:

$$\mathbf{p}_f = \mathbf{A}\mathbf{C}_f \tag{8}$$

where $\mathbf{p}_f$ is the $(n+1) \times 2$ matrix containing the knot points and $\mathbf{A}$ is the $(n+1) \times (n+1)$ circulant matrix:

$$\mathbf{A} = \begin{bmatrix} 2/3 & 1/6 & 0 & 0 \ldots & 0 & 1/6 \\ 1/6 & 2/3 & 1/6 & 0 \ldots & 0 & 0 \\ \vdots & & \vdots & & \vdots & \\ 1/6 & 0 & 0 & 0 \ldots & 1/6 & 2/3 \end{bmatrix} \tag{9}$$

It must be pointed out here that the knot points belong to the derived B-spline. However, it can be seen that for any two curves, it is not certain that their estimated knot points correspond, even if they are equal in number. For this reason, they must be reallocated on each curve (Cohen et al. 1995). In particular, we place l knot points equally spaced with respect to u' onto each curve, with the first knot at an arbitrary location on the spline. When the correct starting point is estimated during the normalization process, reallocation is again performed, as described in the sequel. The underlying reason for this method is that for any input sample curve in the system, the reallocated knot points should always correspond. Finally, a classifier based on the reallocated knot points could be based on minimizing a metric such as

$$d^2 = \sum_{i=1}^{l} \left\| \mathbf{p}_i^{(a)} - \mathbf{p}_i^{(b)} \right\|^2 \tag{10}$$

where a, b denote the ath and bth splines subjected to comparison. Other metrics can also be employed.

4 Curve orthogonalization

What has been achieved so far by the use of B-splines for the representation of a 2-D (planar) closed curve is simplification of the curve shape, reduction of segmentation noise, and obtaining uniform curve sampling in terms of the arc length. A *curve orthogonalization* procedure is presented in this section, as the first stage of normalization. This procedure effectively normalizes a curve with respect to possible translation, skewing, and scaling, and reduces an affine transformation to an orthogonal one, i.e., a transformation that involves only rotation or reflection.

4.1 Orthogonalization procedure

Let $\mathbf{s}_i = [x_i \; y_i]^{\mathrm{T}}$, $i = 0, 1, \ldots, N-1$, be N curve points obtained through B-spline modeling. A $2 \times N$ matrix notation $\mathbf{s} = [\mathbf{s}_0 \mathbf{s}_1 \ldots \mathbf{s}_{N-1}]$ will be used next to represent the curve, permitting convenient notation of transformations that involve multiplication by 2×2 matrices. In a similar fashion, the horizontal and vertical coordinates of the points will be represented by the $1 \times N$ vectors $\mathbf{x} = [x_0 \; x_1 \ldots x_{N-1}]$ and $\mathbf{y} = [y_0 \; y_1 \ldots y_{N-1}]$, respectively. For each curve $\mathbf{s}$, the (p,q)-order moments

$$m_{pq}(\mathbf{s}) = \frac{1}{N} \sum_{i=0}^{N-1} x_i^p y_i^q \tag{11}$$

of order up to two are used for the construction of the corresponding *normalized curve* $n_a(\mathbf{s})$. Without loss of generality, it is assumed from now on that $\mathbf{s}$ does not represent a line segment, so that $m_{20}(\mathbf{s}) \neq 0$ and $m_{02}(\mathbf{s}) \neq 0$. In such a case, normalization steps involving division by these quantities may be omitted.

The orthogonalization procedure comprises a set of linear operations (translation, scaling, and rotation) that do not depend on the selected starting point of the closed curve (or, in general, on the order of the points on the curve). For simplicity, in the subsequent analysis the addition or subtraction of a scalar from a vector (or from a row vector of a matrix) denotes addition or subtraction of all its elements.

1. The center of gravity of the curve is normalized so as to coincide with the origin:

$$\mathbf{x}_1 = \mathbf{x} - m_{10}(\mathbf{s}), \quad \mathbf{y}_1 = \mathbf{y} - m_{01}(\mathbf{s}) \tag{12}$$

   where $\mu_\chi = m_{10}(\mathbf{s})$, $\mu_y = m_{01}(\mathbf{s})$.
2. The curve is scaled horizontally and vertically so that its second-order moments become equal to one:

$$\mathbf{x}_2 = \sigma_x \mathbf{x}_1, \quad \mathbf{y}_2 = \sigma_y \mathbf{y}_1 \tag{13}$$

   where $\sigma_x = 1/\sqrt{m_{20}(\mathbf{s}_1)}$, $\sigma_y = 1/\sqrt{m_{02}(\mathbf{s}_1)}$.
3. The curve is rotated counterclockwise by $\theta_0 = \pi/4$:

$$\mathbf{s}_3 = \mathbf{R}_{\pi/4} \mathbf{s}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{x}_2 - \mathbf{y}_2 \\ \mathbf{x}_2 + \mathbf{y}_2 \end{bmatrix} \tag{14}$$

   where $\mathbf{R}_\theta$ is a 2×2 matrix corresponding to a counterclockwise rotation by θ radians.
4. Finally, the curve is scaled again, exactly as in step 2:

$$\mathbf{x}_4 = \tau_x \mathbf{x}_3, \quad \mathbf{y}_4 = \tau_y \mathbf{y}_3 \tag{15}$$

   where $\tau_x = 1/\sqrt{m_{20}(\mathbf{s}_3)}$, $\tau_y = 1/\sqrt{m_{02}(\mathbf{s}_3)}$.

The normalized curve $n_a(\mathbf{s}) \equiv \mathbf{s}_4$ can also be written as

$$\begin{aligned} n_a(\mathbf{s}) &= \mathbf{N}(\mathbf{s})(\mathbf{s} - \mu(\mathbf{s})) \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} \tau_x & 0 \\ 0 & \tau_y \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix} \left(\begin{bmatrix} \mathbf{x} - \mu_x \\ \mathbf{y} - \mu_y \end{bmatrix} \right) \end{aligned} \tag{16}$$

where $\mu(\mathbf{s}) = [m_{10}(\mathbf{s}) \; m_{01}(\mathbf{s})]^{\mathrm{T}}$ and $\mathbf{N}(\mathbf{s})$ denotes the 2×2 *normalization matrix* of $\mathbf{s}$. Although the dependence of parameters μ_χ, μ_y, σ_χ, σ_y, τ_χ, and τ_y on $\mathbf{s}$ is omitted for simplicity, the normalization matrix is still a function of $\mathbf{s}$. It can be seen that curve $\mathbf{s}_2$ obtained from step 2 above satisfies $m_{10}(\mathbf{s}_2) = m_{01}(\mathbf{s}_2) = 0$ and $m_{20}(\mathbf{s}_2) = m_{02}(\mathbf{s}_2) = 1$. The extra rotation and scaling of steps 3 and 4 are required so that $m_{11}(n_a(\mathbf{s})) = 0$. The following proposition summarizes the necessary and sufficient conditions for curve normalization (see Appendix A for the proof).

Proposition 1. *For every initial curve* $\mathbf{s}$*, the normalized curve* $n_a(\mathbf{s})$ *defined in (12)–(15) has the following properties:*

$$\begin{aligned} &m_{10}(n_a(\mathbf{s})) = m_{01}(n_a(\mathbf{s})) = m_{11}(n_a(\mathbf{s})) = 0\,, \\ &m_{20}(n_a(\mathbf{s})) = m_{02}(n_a(\mathbf{s})) = 1 \end{aligned} \tag{17}$$

Moreover, the above conditions can only be achieved if the rotation angle θ_0 *used in normalization step 3 is equal to* $k\pi/2 + \pi/4$, $k \in Z$.

Conditions (17) actually imply that the matrix representation of a normalized curve is orthogonal; that is, $n_a(\mathbf{s})(n_a(\mathbf{s}))^{\mathrm{T}} = \mathbf{I}_2$, where $\mathbf{I}_2$ is the 2×2 identity matrix. As a consequence, it will be seen below that this kind of normalization also reduces an affine transformation to an orthogonal one, thereby removing translation, scaling, and skew transformations. Moreover, the orthogonalization procedure is itself an affine transformation, as shown in (16); therefore no information on curve shape is lost, and so the original curve can always be recovered by applying the inverse transformation.

4.2 Invariance to translation, scaling, and skewing

Let us now consider two curves $\mathbf{s}$, $\mathbf{s}'$ related through an affine transformation:

$$\mathbf{s}' = \mathbf{A}\mathbf{s} + \mathbf{t} = \begin{bmatrix} \mathbf{x}' \\ \mathbf{y}' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{18}$$

where matrix $\mathbf{A}$ is assumed to be of full rank. Otherwise, in the degenerate case of $\det \mathbf{A} = 0$, any 2-D data set is mapped onto a line and normalization is impossible. Then $\mu(\mathbf{s}') = \mathbf{A}\mu(\mathbf{s}) + \mathbf{t}$, so that $\mathbf{s}_1' = \mathbf{s}' - \mu(\mathbf{s}') = \mathbf{A}(\mathbf{s} - \mu(\mathbf{s})) = \mathbf{A}\mathbf{s}_1$. Thus, after step 1 for translation normalization, curves $\mathbf{s}_1$ and $\mathbf{s}_1'$ are only related through the multiplication of a 2×2 matrix. In this case, the relation between the moments of the two curves is

$$m_{20}(\mathbf{s}_1') = a^2 m_{20}(\mathbf{s}_1) + b^2 m_{02}(\mathbf{s}_1) + 2ab\, m_{11}(\mathbf{s}_1) \tag{19a}$$

$$m_{02}(\mathbf{s}_1') = c^2 m_{20}(\mathbf{s}_1) + d^2 m_{02}(\mathbf{s}_1) + 2cd\, m_{11}(\mathbf{s}_1) \tag{19b}$$

$$m_{11}(\mathbf{s}_1') = ac\, m_{20}(\mathbf{s}_1) + bd\, m_{02}(\mathbf{s}_1) + (ad + bc)\, m_{11}(\mathbf{s}_1) \tag{19c}$$

It is then observed is that if $\mathbf{s}_1$ is normalized, i.e., satisfies (17), then the above equations reduce to

$$m_{20}(\mathbf{s}_1') = a^2 + b^2 \tag{20a}$$

$$m_{02}(\mathbf{s}'_1) = c^2 + d^2 \tag{20b}$$

$$m_{11}(\mathbf{s}'_1) = ac + bd \tag{20c}$$

This means, for example, that when a normalized curve is rotated or reflected – in which case **A** is orthogonal – it remains normalized. Inversely, if both curves are normalized, then **A** should be orthogonal. In this sense Proposition 2 establishes the relation of two curves in terms of their normalized counterparts (see Appendix B for the proof).

Proposition 2. *If two curves* **s**, **s**′ *are related through an affine transformation, there exists an orthogonal* 2×2 *matrix* **Q** *such that:*

$$n_a(\mathbf{s}') = \mathbf{Q}\, n_a(\mathbf{s}) \tag{21}$$

Moreover, the same relation holds between $n_a(\mathbf{s})$ *and a normalized curve* $\tilde{n}_a(\mathbf{s})$ *obtained by any possible normalization procedure based on (17).*

Thus, the proposed normalization procedure reduces affine transformations to orthogonal transformations that may contain only rotation and/or reflection (depending on whether $\det \mathbf{Q} = 1$ or $\det \mathbf{Q} = -1$), and therefore normalized curves are invariant to translation, scaling, and skew transformations. It is very important to note that normalization is performed *without knowledge* of the affine parameters **A** and **t**, and *without one-to-one matching* between curves **s** and **s**′; i.e., **s** is normalized without knowledge of **s**′, and vice versa. Note that transformation parameters $\{\mu_\chi,\ \mu_y,\ \sigma_\chi,\ \sigma_y,\ \tau_\chi,\ \tau_y\}$ are directly estimated from first- and second-order statistics of curve data. Furthermore, it can be seen that this set of parameters along with $n_a(\mathbf{s})$ contain all the information about the original curve **s**; although $n_a(\mathbf{s})$ satisfies the five constraints in (17), the parameters are six because τ_χ and τ_y are related by $1/\tau_x^2 + 1/\tau_y^2 = 1$. The *uniqueness* and *completeness* of $n_a(\mathbf{s})$ is thus verified, since it contains all information about **s** except for the transformation for which it is invariant. Since normalization is not based on matching, any classification mechanism could be used, including neural networks for instance. Finally, as demonstrated in the experiments, $n_a(\mathbf{s})$ is robust to noise (caused by segmentation or sampling) and to curve dissimilarities, since it is based on global properties such as curve moments.

5 Starting point and rotation normalization

At this point we have achieved the reduction of affine transformations to orthogonal ones; thus it would suffice to provide a transformation that is invariant to rotation and reflection. The overall normalization (orthogonalization and normalization with respect to rotation and reflection) would then be affine invariant. However, the latter normalization procedure comprises operations that do depend on the selected starting point of a closed curve and, in general, on the order of points on the curve. For example, rotation normalization depends on the starting point while reflection normalization depends on rotation. For this reason, starting point normalization is presented first – rotation and reflection normalization then follows.

In the subsequent discussion, normalization is based on the discrete Fourier transform instead of curve moments. For this purpose, complex vector notation is used for curve representation instead of matrices. The horizontal and vertical coordinates of the points on the curve are still represented by the $N \times 1$ vectors $\mathbf{x} = [x_0\ x_1 \dots x_{N-1}]^{\mathrm{T}}$ and $\mathbf{y} = [y_0 y_1 \dots y_{N-1}]^{\mathrm{T}}$, respectively; however, a curve is represented by the complex $N \times 1$ vector $\mathbf{z} = \mathbf{x} + j\mathbf{y} = [z_0\, z_1 \dots z_{N-1}]^{\mathrm{T}}$, where $z_i = x_i + jy_i$, $i = 0, 1, \dots, N-1$, denotes a single curve point.

5.1 Starting-point normalization

The normalization procedure with respect to the starting point is based on the discrete Fourier transform of the complex vector **z** representing a curve:

$$\mathbf{u} = \Im(\mathbf{z}) = \mathbf{W}\,\mathbf{z} = \begin{bmatrix} w^0 & w^0 & \cdots & w^0 \\ w^0 & w^1 & \cdots & w^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ w^0 & w^{N-1} & \cdots & w^{(N-1)^2} \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{N-1} \end{bmatrix} \tag{22}$$

where $w = e^{\mathrm{j}2\pi/N}$, so that $w^{\ell N} = 1$, $\ell \in Z$. Equivalently,

$$u_k = \sum_{i=0}^{N-1} z_i w^{-ki}, \quad k = 0, 1, \dots, N-1 \tag{23}$$

For each element of the Fourier transform we will employ its *primary argument*, or *phase*, defined as $a_k = \arg u_k = \theta \in [0, 2\pi)$: $u_k = r\, e^{\mathrm{j}\theta}$ where $r \in \Re^+$. The corresponding *phase vector* is then $\mathbf{a} = \arg \mathbf{u} = [a_0\ a_1 \cdots a_{N-1}]$. Consider now a second curve $\mathbf{z}' = [z'_0\ \ z'_1 \cdots z'_{N-1}]$ that is the same as **z** except for its starting point; i.e., it is circularly shifted with respect to **z** by m samples, where $m \in \{0, 1, \dots, N-1\}$:

$$\mathbf{z}' = S_m(\mathbf{z}): \quad z'_i = z_{(i+m) \bmod N}, \quad i = 0, 1, \dots, N-1 \tag{24}$$

Then, if we similarly define $\mathbf{u}' = \Im(\mathbf{z}')$ and $\mathbf{a}' = \arg \mathbf{u}'$, it can be shown that $u'_k = w^{km} u_k$, $k = 0, 1, \dots, N-1$, or

$$a'_k = (a_k + 2\pi k m / N) \bmod 2\pi, \quad k = 0, 1, \dots, N-1 \tag{25}$$

Hence the circular shift between **z** and **z**′ is reduced to a very simple relation between **a** and **a**′. If both curves **z** and **z**′ were known, matching between them would be feasible by estimating m from (25) and then applying a circular shift on **z**′. This, however, is not possible for a normalization scheme, since at any time only one curve is known. Even when both curves are known, m can be estimated only if they correspond to exactly the same curve (apart from the starting-point difference). In any other case, Fourier phases are so noisy that (25) can give no useful information. For this reason we estimate a *standard* circular shift for each curve, based on the difference between the first and last Fourier phases:

$$p(\mathbf{z}) = \left\lfloor \frac{N}{4\pi}(a_1 - a_{N-1}) \right\rfloor \bmod N/2 \tag{26}$$

and then apply the opposite shift in order to normalize the curve:

$$n_p(\mathbf{z}) = S_{-p(\mathbf{z})}(\mathbf{z}) \tag{27}$$

It is evident from (25) that a difference between two successive elements of $\mathbf{a}'$ (e.g., $a'_2 - a'_1$) is related to the respective difference $a_2 - a_1$ by a simple addition of $2\pi m/N$, modulo 2π. Thus, if $a_2 - a_1 = 0$, it is straightforward to obtain an estimate of m by dividing this difference by $(2\pi/N)$ and taking modulo N. For the selection of the first and last phases, however (which is explained below), elements a_1 and a_{N-1} are not successive – they differ by two samples since the Fourier coefficients are periodic with period N. An extra division by 2 is necessary in this case. This explains the above definition of $p(\mathbf{z})$, where the integer part is necessary for subsequently applying a circular shift in (27).

It can be shown that if N is even, the above normalization is invariant to the starting point, except for an uncertainty in the standard circular shift, which may cause an extra shift of $N/2$. If N is not even, an additional shift of ±1 sample between $n_p(\mathbf{z})$ and $n_p(\mathbf{z}')$ is obtained (see Appendix C for the proof).

Proposition 3. *If two curves* $\mathbf{z}$, $\mathbf{z}'$ *are related through a circular shift transformation, then*

$$p(\mathbf{z}') = (p(\mathbf{z}) + m) \mod N/2 \tag{28a}$$

$$p(n_p(\mathbf{z}')) = p(n_p(\mathbf{z})) = 0 \tag{28b}$$

$$n_p(\mathbf{z}') = \begin{cases} n_p(\mathbf{z}), & 0 \le p(\mathbf{z}) + m < N/2 \\ S_{N/2}(n_p(\mathbf{z})), & N/2 \le p(\mathbf{z}) + m < N \end{cases} \tag{28c}$$

It can be observed that the additive relation (28a) leads to starting-point normalization since we force – through the circular shift of (27) – the normalized curves to satisfy (28b), exactly as curves are orthogonalized when they are forced to satisfy (17). It is also observed that (28b) is equivalent to forcing Fourier coefficients u_1 and u_{N-1} to have equal phases. It will be seen below that if these phases are also set to zero, so that u_1 and u_{N-1} become real and positive, then rotation normalization can also be achieved. However, since (28a) holds modulo $N/2$, and not modulo N, exact starting-point normalization can only be achieved when $0 \le p(\mathbf{z}) + m < N/2$. In the opposite case, normalization results in an additional shift of $N/2$ in (28c). This uncertainty is caused by the selection of coefficients u_1 and u_{N-1}, and can only be removed after rotation normalization, as described below.

Although the selection of two successive coefficients, such as a_1 and a_2, would lead to complete starting-point invariance, the proposed selection (22) has two significant advantages: it can detect *reflection curve symmetries* and, as demonstrated in the experiments, it is also robust to noise and to curve dissimilarities. For example, selection of a_1 and a_2 for normalization leads to completely different – actually random – starting points for two curves that differ only slightly. The proposed selection of a_1 and a_{N-1} has also been employed in Persoon and Fu (1998) for the detection of the starting point of curves describing line patterns; e.g., character boundaries. The use of other Fourier coefficients is also possible (Marques et al. 1998). Automatic detection of appropriate coefficients is usually based on their magnitude separately from their phase and can deal more efficiently with cases of rotational symmetries (Shen and Ip 1997). The proposed method may fail in such cases, although the corresponding experiments were not extensive. However, such cases are rare in real-world applications, and the performance of the proposed normalization is satisfactory for natural images.

5.2 Rotation and reflection normalization

Assume that two original curves have been first orthogonalized and then normalized with respect to their starting point, and that the normalized curves are denoted by $\mathbf{s}$, $\mathbf{s}'$. According to Proposition 2, the normalized curves will satisfy $\mathbf{s} = \mathbf{Q}\mathbf{s}'$, where $\mathbf{Q}$ is an orthogonal 2×2 matrix. We can then uniquely decompose $\mathbf{Q}$ as

$$\mathbf{Q} = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \tag{29}$$

where $\theta \in [0, \pi)$, $s_x = \pm 1$, $s_y = \pm 1$, so that there is one-to-one relation between these parameters and the elements of $\mathbf{Q}$. In other words, $\mathbf{Q}$ uniquely corresponds to a rotation between 0 and π degrees, as well as a horizontal and a vertical reflection. For simplicity, we adopt again the complex vector notation $\mathbf{z}$ and $\mathbf{z}'$ for the two curves $\mathbf{s}$ and $\mathbf{s}'$, respectively, and rewrite their relation as

$$\mathbf{z}' = (s_x\mathbf{x} + \mathrm{j}s_y\mathbf{y})\, e^{\mathrm{j}\theta} \tag{30}$$

This brings us to the normalization procedure with respect to rotation and reflection, which comprises two steps of attempting to normalize rotation first and then reflection. The procedure is defined for curve $\mathbf{z}$, and can be similarly applied to $\mathbf{z}'$. As in the starting-point case, normalization is based on the first and last Fourier phases a_1 and a_{N-1} of $\mathbf{z}$, where $\mathbf{a} = \arg \Im(\mathbf{z}) = [a_0\ a_1 \cdots a_{N-1}]$:

1. The rotation of the initial curve $\mathbf{z}$ is normalized according to the average value of a_1 and a_{N-1}:

$$r(\mathbf{z}) = \left(\frac{1}{2}(a_1 + a_{N-1})\right) \mod \pi \tag{31a}$$

$$\mathbf{z}_1 = \mathbf{z}\, e^{-\mathrm{j}r(\mathbf{z})} \tag{31b}$$

2. The horizontal and vertical reflection is normalized according to the third-order moments of $\mathbf{z}_1$:

$$\begin{aligned} v(\mathbf{z}_1) &= v_x(\mathbf{z}_1) + \mathrm{j}v_y(\mathbf{z}_1) \\ &= \operatorname{sgn} m_{12}(\mathbf{z}_1) + \mathrm{j}\operatorname{sgn} m_{21}(\mathbf{z}_1) \end{aligned} \tag{32a}$$

$$n_r(\mathbf{z}) = \mathbf{z}_2 = v_x(\mathbf{z}_1)\mathbf{x}_1 + \mathrm{j}v_y(\mathbf{z}_1)\mathbf{y}_1 \tag{32b}$$

where sgn denotes the signum function. Note that it would be equally sufficient to define $r(\mathbf{z}) = a_1$. The above definition, however, leads to elimination of the starting-point effect seen in (25), so that the rotation normalization procedure presented here is invariant to the starting point, as explained below. Similarly with the curve orthogonalization procedure, the *uniqueness* and *completeness* of $n_r(\mathbf{z})$ can be proved (i.e., that $n_r(\mathbf{z})$ is invariant to rotation and reflection transformations), and that it retains all information about $\mathbf{s}$ except for the transformations for which it is invariant (see Appendix D the proof).

Proposition 4. *If two curves* $\mathbf{s}$, $\mathbf{s}'$ *are related through an orthogonal transformation, i.e.,* $\mathbf{s} = \mathbf{Q}\mathbf{s}'$ *where* $\mathbf{Q}$ *is an orthogonal* 2×2 *matrix decomposed into a rotation and a reflection matrix, then for their complex vector counterparts* $\mathbf{z}$, $\mathbf{z}'$ *the following hold:*

$$r(\mathbf{z}') = (\lambda r(\mathbf{z}) + \theta) \mod \pi \tag{33a}$$

$$n_r(\mathbf{z}') = n_r(\mathbf{z}) \tag{33b}$$

$$\begin{aligned} &r(n_r(\mathbf{z}')) = r(n_r(\mathbf{z})) = 0, \\ &v_x(n_r(\mathbf{z}')) = v_y(n_r(\mathbf{z}')) = v_x(n_r(\mathbf{z})) = v_y(n_r(\mathbf{z})) = 1 \end{aligned} \tag{33c}$$

where $\lambda = s_x s_y = \pm 1$.

Note that as in (17) and (28b), rotation and reflection transformations are removed so that normalized curves satisfy (33c). Furthermore, as in curve orthogonalization, the set of parameters $\{r(\mathbf{z}),\ v_x(\mathbf{z}),\ v_y(\mathbf{z})\}$ together with $n_r(\mathbf{z})$ contain all information about the original curve $\mathbf{z}$. It can also be deduced from the proof of Proposition 4 that the normalization procedure presented above is independent of the starting point; that is, $n_r(S_m(\mathbf{z})) = S_m(n_r(\mathbf{z}))$ for any $m \in \{0, 1, \ldots, N-1\}$, exactly as $n_a(S_m(\mathbf{s})) = S_m(n_a(\mathbf{s}))$. Thus rotation and reflection normalization can even be applied *before* starting-point normalization. In this case, it is very important to note that combining the results of Propositions 2 and 4 – curve orthogonalization followed by rotation and reflection normalization, i.e., $n_r(n_a(\mathbf{z}))$ – is *invariant to any affine transformation*. Finally, the robustness of $n_r(\mathbf{z})$ to curve deformations and noise is demonstrated in the experiments. Note also that similar efforts have been made for rotation normalization based on higher-order moments (Taubin and Cooper 1992).

Two final normalization steps are required after rotation and reflection normalization. First, recall that starting-point normalization leaves us with an ambiguity of $N/2$. It can be seen that after rotation normalization, the detected starting point (x_0, y_0) lies very close to the x-axis ($y_0 \cong 0$) and far from the y-axis, on either side. The starting-point ambiguity is thus resolved by applying an additional circular shift of $N/2$ samples if $x_0 < 0$; otherwise the normalized curve is left intact. Second, the curve orientation is normalized to being counterclockwise. At this point the entire normalization procedure is complete.

6 Experimental results

In this section, the performance of the proposed algorithm is illustrated and appropriate algorithmic guidelines are included. Initially, indicative results are provided on color and motion segmentation to obtain object contours. Normalization results are then given for all proposed normalization steps; i.e., translation, skewing, scaling, rotation, and starting-point normalization. The algorithm's efficiency is then discussed employing affine transformations of: (a) the same object, (b) the same object under nonuniform sampling, (c) similar objects, and (d) substantially different objects. At the same time, the closed cubic B-splines framework is implemented in order to counteract matching difficulties due to sampling and noise problems.

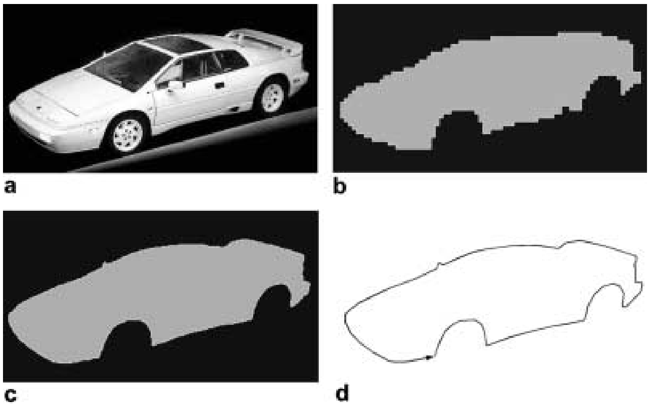


Fig. 1a–d. Indicative color segmentation results for an image of a car. **a** Original image. **b** Segmented image (first stage). **c** Segmented image (final result). **d** Obtained object contour

As mentioned in Sect. 2, color and motion segmentation results substantially affect the performance of the normalization schemes. The latter is generally true for any contour modeling and curve-matching approach based on natural data. In the present study existing color and motion segmentation algorithms in the literature were tested in order to obtain the best possible object contours from static images and image sequences. In this context, the M-RSST approach yielded adequate results for color objects in high resolution images, while motion segmentation was performed on the basis of 2-D parametric motion models. Some indicative color segmentation results for a still image of a car are illustrated in Fig. 1. Figure 1a depicts the original image, while the results of the first and final steps of the segmentation scheme (M-RSST) are depicted in Fig. 1b and c, respectively. The obtained contour in terms of sample points is depicted in Fig. 1d.

The derived object contours are next fed to the proposed orthogonalization scheme. It must be pointed out that the proposed algorithm is employed straightforwardly, in the sense that no human interaction is needed with respect to the particular contour and any parameters are initialized once, when the system is set on-line. Results of the algorithm's performance on the basis of sample points on the obtained contour are illustrated in Fig. 2. A sample contour consisting of 100 sample points, obtained from color segmentation on a still image containing a fish, is depicted on the upper right-hand side of Fig. 2a. Two other sample curves consisting of 100 points are included in the same figure, which were obtained through direct arbitrary affine transformations of the original sample points. Following the above-described methodology, the sample points of all three curves were subjected to translation, skewing, scaling, rotation, and starting-point normalization. In this sense, the resulting sample sets are depicted in Fig. 2b–d after translation, skewing/scaling, and rotation/starting-point normalization, respectively. In all cases, the starting point and orientation of the curves are denoted by an arrow. It can be seen that the final sample curves match perfectly when the initial curves are affine transformations of the same sample curve, verifying the results of Sects. 4 and 5. The computational cost of the normalization procedure is negligible: for curves consisting of 100 sample

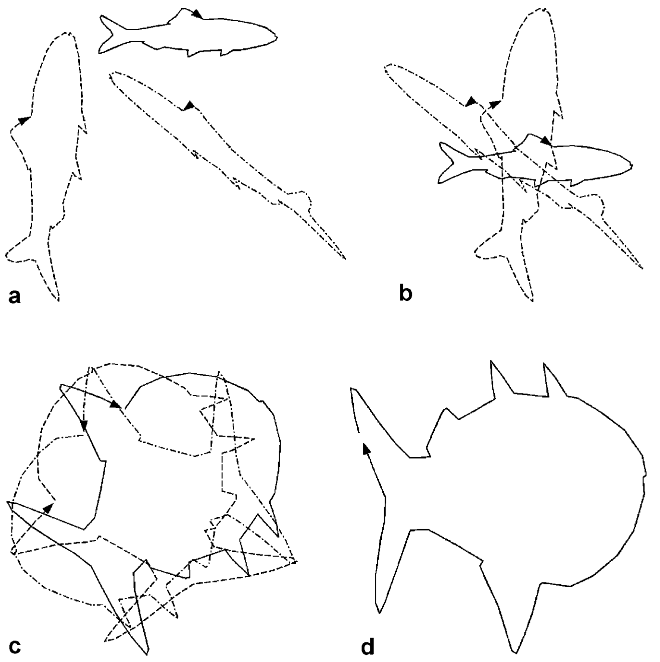


Fig. 2a–d. Intermediate and final results of the proposed orthogonalization scheme on the samples of a fish contour (no resampling scheme was employed). **a** Original contour and a pair of arbitrary affine transformations. **b** Curves after translation normalization. **c** Curves after scaling normalization. **d** Final curves

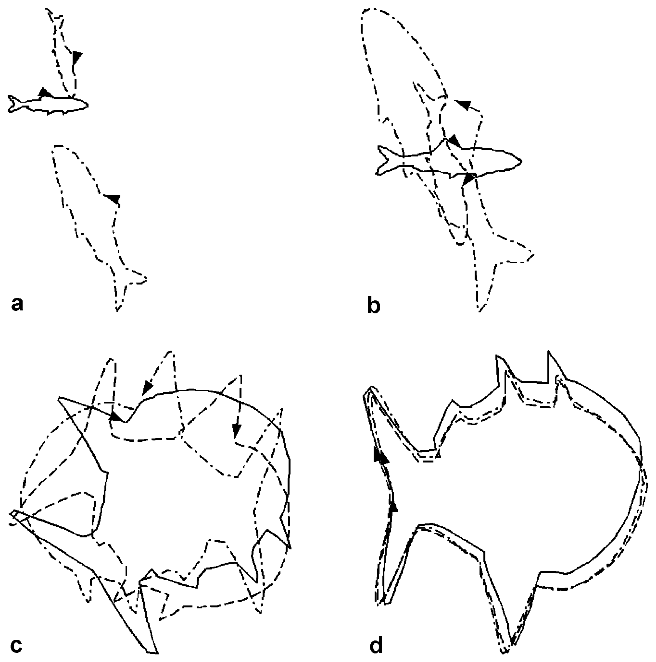


Fig. 3a–d. Results of the proposed orthogonalization scheme on the samples of a fish contour after nonuniform resampling on input curves. **a** Original contour and arbitrary affine transformations. **b,c,d** Mismatching of curves after translation, scaling, and rotation normalization, respectively

points, the average execution time is of the order of 2 ms using a nonoptimized Matlab implementation on a 266 MHz Pentium II PC. The execution time increases linearly with sample size, as only two of the FFT coefficients are required in the computations.

The resulting (orthogonalized) sample sets are identical only when derived from affine transformations of one particular sample curve. In this context, Fig. 3 demonstrates mismatching of orthogonalized curves derived from a particular initial contour of a fish under different (and nonuniform) sampling. In Fig. 3a, the initial sample curve is transformed using a pair of arbitrary affine transformations and the resulting curves are then resampled using nonuniform sampling. The results obtained from the normalization steps are subsequently depicted in Fig. 3b–d. It can be seen that the results are not far from the desired ones, even for significant differences in the sampling process. However, the observed dissimilarities can be annoying when the particular application requires high accuracy in the curve matching.

The obtained results can be significantly improved by performing a resampling scheme along with the proposed normalization schemes. Cubic B-splines have been utilized for this purpose, as knot point fitting and reallocation appeared to yield adequate results in terms of uniform sampling (see Sect. 3). Figure 4 illustrates the improvement in matching between orthogonalized curves by performing B-spline modeling and knot point reallocation twice; namely, before translation normalization and before starting-point/rotation rejection. A final resampling step is taken on the basis of the final curves, so as their knot points correspond and the matching technique is enhanced. The plots illustrated in Fig. 4 are directly comparable with that of Fig. 3. When referring to orthogonalized curves we will hereon assume that B-spline modeling and knot point reallocation is performed twice, along the lines of the aforementioned methodology.

Maybe the most important property of the proposed approach is its ability to align curves that appear to be spatially “similar”. The latter was tested over several object contours belonging to distinct object classes, such as fish, airplanes, cars, glasses, and hammers. Figure 5 illustrates orthogonalization results for three distinct airplanes obtained through color segmentation, and consequently under substantially different sampling. The proposed approach, along with the closed cubic B-spline framework, successfully aligned spatially relative object contours. The orthogonalization results in Fig. 5 should be directly compared to those of Fig. 6, where three sample curves belonging to distinct object classes are employed. It can be seen that, compared to those of Fig. 6d, the final curves of Fig. 5d yield a better resemblance on the basis of almost any matching scheme.

The proposed approach yields adequate results even in the presence of significant amount of noise in the sample curves. In order to model noise in object contours due to color/motion segmentation pitfalls, random noise with relatively large variance values was artificially induced into the available object contours. The algorithm’s performance proved to be adequate even in this case, as long as the object contour was not severely deformed. Indicative experimental results are provided in Fig. 7. In Fig. 7a, an object contour corresponding to an airplane along with two noisy counterparts are depicted, one of which is contaminated with twice the amount of noise (in terms of standard deviation) as the other. The respective normalization results are given in Fig. 7b. It must be pointed out here that arti-

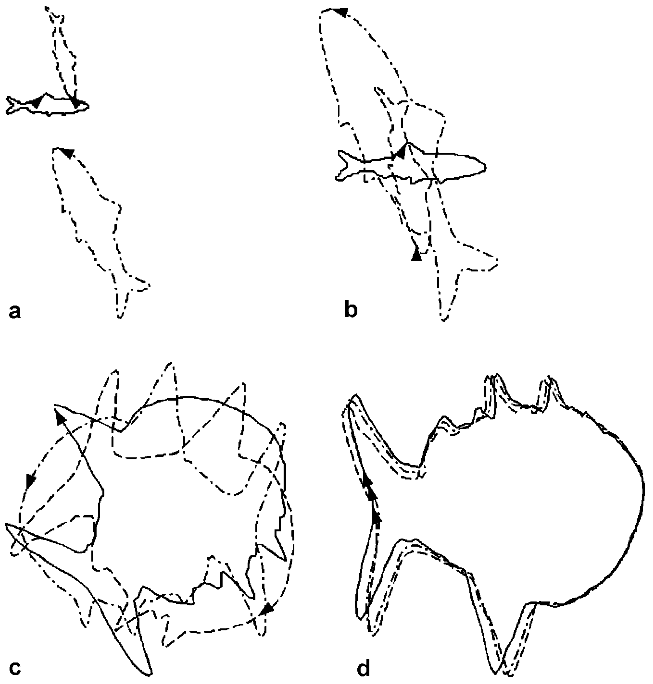


Fig. 4a–d. Improvement of orthogonalization results employing B-spline knot modeling and reallocation on the nonuniform sampled curves of Fig. 3. **a** Original contour and arbitrary affine transformations. **b,c,d** Curves after translation, scaling, and rotation normalization, respectively

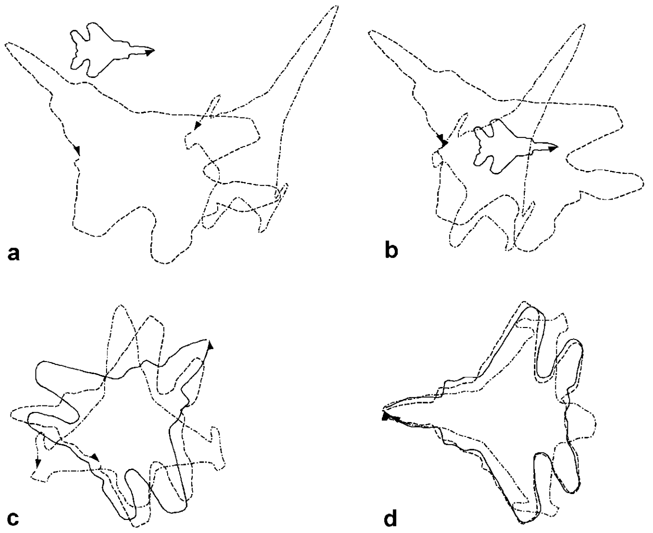


Fig. 5a–d. Immediate alignment of orthogonalization results for semantically similar (spatially similar) object contours. **a** Original contours for three distinct airplanes. **b,c,d** Curves after translation, scaling, and rotation normalization, respectively

ficial noise was induced to the nonuniform samples of the initial curves prior to B-spline modeling. As can be seen in Fig. 7b, the “less” noisy instance yields superior matching results than the “more” noisy one. In Fig. 7c and d, relevant results are depicted employing three spatially similar objects (airplanes).

As stated in the Introduction, object classification based on contour information has been tackled by several authors, and thus it is not analyzed thoroughly in this paper. However, some indicative classification results have been included in

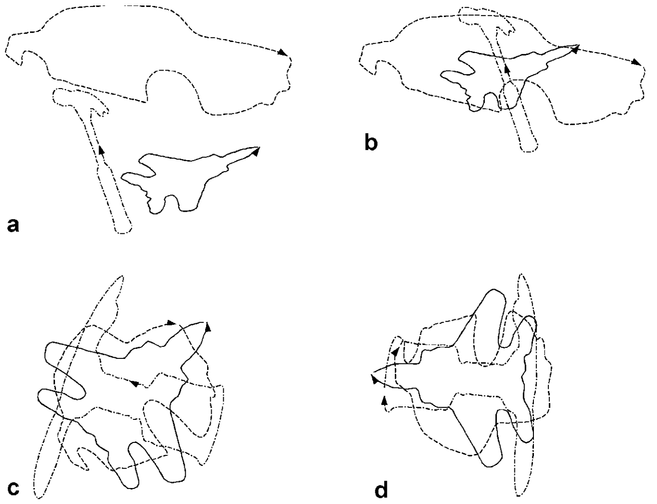


Fig. 6a–d. Alignment of orthogonalized contours for substantially different object contours. **a** Original contours of a hammer, car, and airplane. **b,c,d** Curves after translation, scaling, and rotation normalization, respectively

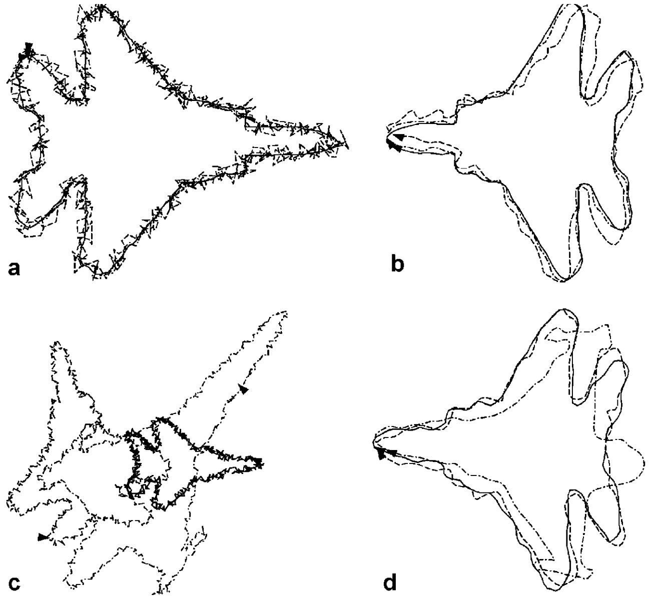


Fig. 7a–d. Orthogonalization results in the presence of random noise in the initial curve samples. **a,b** Initial and final curves for three instances of the same contour. **c,d** Initial and final curves for spatially (semantically) relative objects

Fig. 8. Both the prototype contour 1 and the example contour 2 have been subjected to the proposed normalization procedure. Their “normalized” counterparts were compared on the basis of three simple metrics, namely the Euclidean distance of: (a) the respective point sets, (b) the estimated Fourier descriptors, and (c) the modified Fourier descriptors proposed in Rui et al. (1998). As was both intuitively expected and experimentally verified, all three metrics fail for the initial data sets. On the contrary, they prove to be indicative of the normalized curve resemblance (see Fig. 8). Each set of distances was normalized so that 0 and 1 values denote absolute resemblance and no resemblance, respectively.

Finally, the proposed algorithm’s performance was successfully tested for content-based retrieval purposes based on

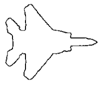
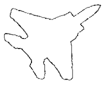
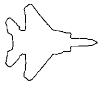
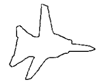
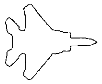
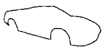
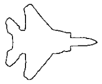
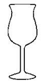
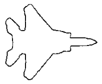
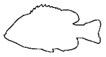
Contour 1	Contour 2	Points	FD	MFD
		0.01	0.02	0.01
		0.19	0.12	0.11
		0.75	0.41	0.57
		0.89	0.62	0.65
		0.76	0.25	0.32

Fig. 8. Indicative estimated contour distances for classification purposes (*FD*, Fourier Descriptors; *MFD*, Modified Fourier Descriptors)

contour similarity on a small database containing 50 still images of 5 visually distinct object classes: namely, airplanes, cars, fish, hammers, and glasses. To obtain best results, particular care was taken so that the main object contours were successfully extracted. In Fig. 9, relevant experimental results are illustrated for an input image containing a airplane. The input image is depicted in Fig. 9a, its extracted contour is shown in Fig. 9b, while the retrieved images are given in descending contour similarity in Fig. 9c. Similar results for an input image containing a car are given in Fig. 10. It must be pointed out for Fig. 9a that by utilizing only the object contour attribute, a fish could yield higher resemblance to an F-15 aircraft than a Stealth aircraft would. For such reasons, the proposed methodology could be employed along with other image attributes such as color and texture information in an integrated content-based retrieval system. It should be noted finally that the proposed technique concentrates only on curve normalization and does not treat the shape retrieval problem by itself; therefore, the results presented in this section can be characterized as indicative. Apart from the normalization scheme, a real-world shape retrieval system would also require a suitable shape-matching algorithm.

7 Conclusions – further work

Using the curve normalization procedure presented in this paper it is possible to obtain an affine-invariant curve representation without any actual loss of information on the original curve. The procedure can be applied as a preprocessing step to any shape representation, classification, recognition, or retrieval technique, since it effectively decouples the problem of affine-invariant description from feature extraction and pattern matching. This is verified by employing a number of well-known curve-matching methods in the context of content-based retrieval from image and video databases. In all cases, the proposed normalization is experimentally shown to be considerably robust to shape deformations and noise. Moreover, the technique is very efficient and can be integrated into any real-time system for content-based retrieval or even video coding, since its computational cost is low.

Enhancement and generalization of this normalization procedure is possible in several directions. First, it could be modified to handle sets of connected or unconnected, open or closed curves for the purpose of optical character recognition or indexing of technical line-drawing databases. Normalization of 2-D image data apart from object contours, or even 3-D models and multidimensional data is also desirable, while a very active field of research is normalization with respect to perspective transformation. Such subjects are currently under investigation.

Appendix A: Proof of Proposition 1

The restriction that $\mathbf{s}$ does not represent a straight-line segment is required so that under any translation, rotation or nonzero scaling transformation, $\mathbf{s}$ does not lie on the x or y axis, thus $m_{20}(\mathbf{s}) \neq 0$, $m_{02}(\mathbf{s}) \neq 0$ and all quantities involved in (12)–(15) can be defined. It can be seen that $m_{10}(\mathbf{s}_1) = m_{01}(\mathbf{s}_1) = 0$. This property is also retained for $n_a(\mathbf{s}) = \mathbf{s}_4$, since the remaining normalization steps only involve scaling and rotation. It is also evident that $m_{20}(\mathbf{s}_2) = m_{02}(\mathbf{s}_2) = m_{20}(\mathbf{s}_4) = m_{02}(\mathbf{s}_4) = 1$. It is thus observed that

$$\begin{aligned} m_{11}(\mathbf{s}_3) &= \mathbf{x}_3\mathbf{y}_3^{\mathrm{T}} = \frac{1}{2}(\mathbf{x}_2 - \mathbf{y}_2)(\mathbf{x}_2 + \mathbf{y}_2)^{\mathrm{T}} \\ &= \frac{1}{2}(m_{20}(\mathbf{s}_2) - m_{02}(\mathbf{s}_2)) = 0 \end{aligned} \qquad (1)$$

so that $m_{11}(\mathbf{s}_4) = m_{11}(\mathbf{s}_3)/(m_{20}(\mathbf{s}_3)m_{02}(\mathbf{s}_3))^{1/2} = 0$, and therefore $n_a(\mathbf{s})$ satisfies (17). If angle $\theta_0 = \pi/4$ is substituted by an arbitrary angle θ in (14), then

$$\mathbf{s}_3 = \mathbf{R}_\vartheta \mathbf{s}_2 = \begin{bmatrix} \mathbf{x}_2 \cos\vartheta - \mathbf{y}_2 \sin\vartheta \\ \mathbf{x}_2 \sin\vartheta + \mathbf{y}_2 \cos\vartheta \end{bmatrix} \qquad (2)$$

and (1) is rewritten as

$$\begin{aligned} m_{11}(\mathbf{s}_3) &= \mathbf{x}_3\mathbf{y}_3^{\mathrm{T}} \\ &= \sin\vartheta\cos\vartheta\left(\mathbf{x}_2\mathbf{x}_2^{\mathrm{T}} - \mathbf{y}_2\mathbf{y}_2^{\mathrm{T}}\right) \\ &\quad +(\cos^2\vartheta - \sin^2\vartheta)\left(\mathbf{x}_2\mathbf{y}_2^{\mathrm{T}}\right) \\ &= \cos(2\vartheta)\left(\mathbf{x}_2\mathbf{y}_2^{\mathrm{T}}\right) \end{aligned} \qquad (3)$$

so in order for $m_{11}(\mathbf{s}_3)$ to be always equal to zero for any initial curve, θ must satisfy $\cos(2\theta) = 0$ and therefore take values $k\pi/2 + \pi/4$, $k \in Z$.

Appendix B: Proof of Proposition 2

From (16), $n_a(\mathbf{s}) = \mathbf{N}(\mathbf{s})(\mathbf{s} - \mu(\mathbf{s})) = \mathbf{N}(\mathbf{s})\mathbf{s}_1$ and $n_a(\mathbf{s}') = \mathbf{N}(\mathbf{s}')\mathbf{s}_1' = \mathbf{N}(\mathbf{s}')\mathbf{A}\mathbf{s}_1$. Since $\mathbf{s}$ and $\mathbf{s}'$ are not line segments, $\det\mathbf{N}(\mathbf{s}) = \sigma_x\sigma_y\tau_x\tau_y/\sqrt{2} \neq 0$ (and similarly for $\mathbf{N}(\mathbf{s}')$), so the two normalization matrices are nonsingular and we can define

$$\mathbf{Q} = \mathbf{N}(\mathbf{s}')\mathbf{A}[\mathbf{N}(\mathbf{s})]^{-1} = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \qquad (4)$$

so that $n_a(\mathbf{s}') = \mathbf{Q}n_a(\mathbf{s})$. Now since both $n_a(\mathbf{s})$ and $n_a(\mathbf{s}')$ are normalized, we obtain from (19)

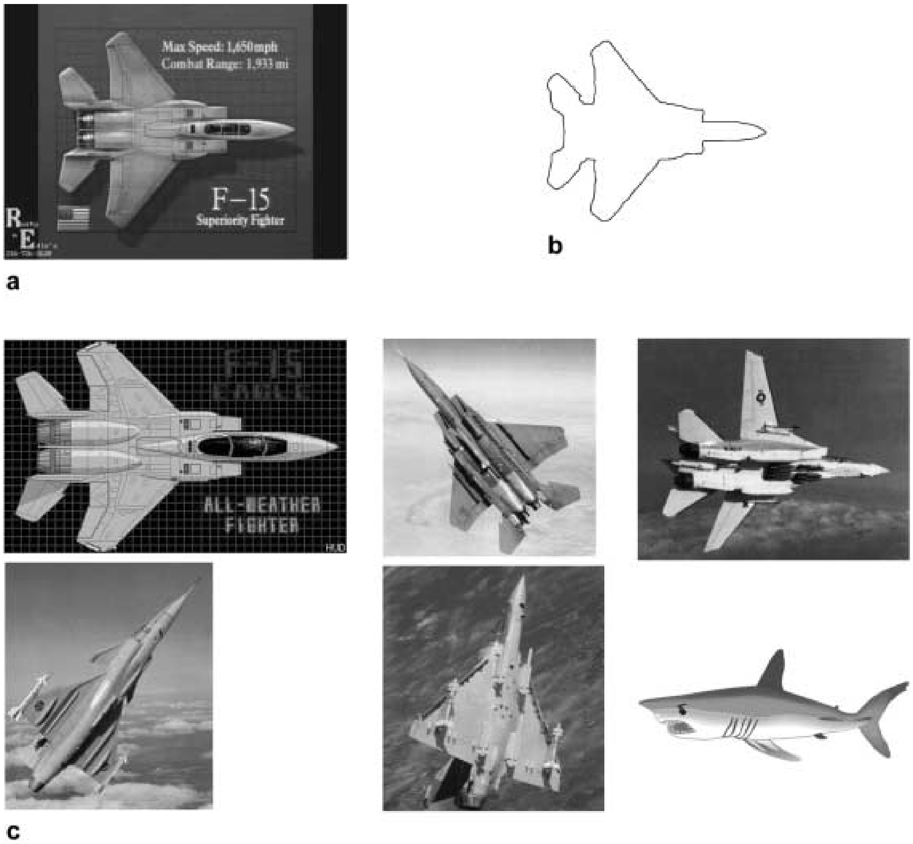


Fig. 9a–c. Image retrieval results (query-by-example) based on contour similarity for an input image containing an airplane. **a** Input image. **b** Extracted contour. **c** Retrieved images

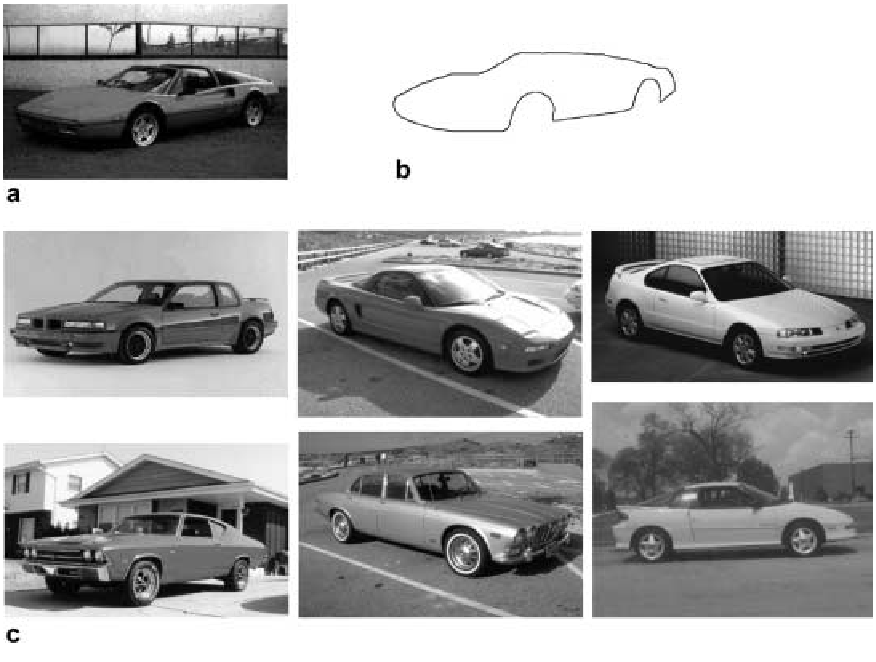


Fig. 10a–c. Image retrieval results for an input image containing a car. **a** Input image. **b** Extracted contour. **c** Retrieved images

$$m_{20}(n_a(\mathbf{s}')) = q_{11}^2 + q_{12}^2 = 1 \tag{38a}$$

$$m_{02}(n_a(\mathbf{s}')) = q_{21}^2 + q_{22}^2 = 1 \tag{38b}$$

$$m_{02}(n_a(\mathbf{s}')) = q_{11}q_{21} + q_{12}q_{22} = 0 \tag{38c}$$

Thus $\mathbf{Q}\mathbf{Q}^{\mathrm{T}} = \mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{I}_2$ and $\mathbf{Q}$ is an orthogonal matrix, meaning that $n_a(\mathbf{s})$ and $n_a(\mathbf{s}')$ differ only in a rotation transformation (if $\det \mathbf{Q} = 1$), plus a possible reflection transformation when $\det \mathbf{Q} = -1$.

Now let $\tilde{\mathbf{N}}(\mathbf{s})$ be another normalization scheme for $\mathbf{s}$ yielding a normalized curve $\tilde{n}_a(\mathbf{s}) = \tilde{\mathbf{N}}(\mathbf{s})\mathbf{s}_1$. In this case we can define $\tilde{\mathbf{Q}} = \tilde{\mathbf{N}}(\mathbf{s})[\mathbf{N}(\mathbf{s})]^{-1}$ so that $\tilde{n}_a(\mathbf{s}) = \tilde{\mathbf{Q}}\, n_a(\mathbf{s})$, and since both $n_a(\mathbf{s})$ and $\tilde{n}_a(\mathbf{s})$ are normalized, we can similarly conclude that $\tilde{\mathbf{Q}}$ is orthogonal. Therefore all possible normalization schemes based on (17) result in a curve that is related to the proposed $n_a(\mathbf{s})$ by a mere rotation (or reflection).

Appendix C: Proof of Proposition 3

If $\mathbf{z}' = S_m(\mathbf{z})$, then (25) holds and we can substitute $a'_1 = (a_1 + 2\pi m/N) \bmod 2\pi$ and $a'_{N-1} = (a_{N-1} - 2\pi m/N) \bmod 2\pi$ into

$$p(\mathbf{z}') = \left\lfloor \frac{N}{4\pi}(a'_1 - a'_{N-1}) \right\rfloor \bmod N/2 \tag{39}$$

Moreover, if N is even, $N/2$ is integer and we can interchange the integer part $\lfloor\cdot\rfloor$ and mod $N/2$ operators. This gives

$$p(\mathbf{z}') = \left(\left\lfloor \frac{N}{4\pi}(a_1 - a_{N-1}) \right\rfloor \bmod N/2 + m \right) \bmod N/2 \tag{40}$$

which leads to (28a). Now considering that $n_p(\mathbf{z})$ is obtained from $\mathbf{z}$ through a circular shift of $N - p(\mathbf{z})$, we can use (28a) for $\mathbf{z}$ and $n_p(\mathbf{z})$:

$$\begin{aligned} p(n_p(\mathbf{z})) &= p(S_{N-p(\mathbf{z})}(\mathbf{z})) \\ &= (p(\mathbf{z}) + (N - p(\mathbf{z}))) \bmod N/2 = 0 \end{aligned} \tag{41}$$

and a similar argument for $\mathbf{z}'$ and $n_p(\mathbf{z}')$ leads to (28b). Next, consider that $n_p(\mathbf{z}) = S_{-p(\mathbf{z})}(\mathbf{z})$ and $n_p(\mathbf{z}') = S_{-p(\mathbf{z}')}(\mathbf{z}') = S_{m-p(\mathbf{z}')}(\mathbf{z})$, so that $n_p(\mathbf{z}') = S_{p(\mathbf{z})+m-p(\mathbf{z}')}(n_p(\mathbf{z}))$. But then we can substitute

$$\begin{aligned} p(\mathbf{z}') &= (p(\mathbf{z}) + m) \bmod N/2 \\ &= \begin{cases} p(\mathbf{z}) + m, & 0 \le p(\mathbf{z}) + m < N/2 \\ p(\mathbf{z}) + m - N/2, & N/2 \le p(\mathbf{z}) + m < N \end{cases} \end{aligned} \tag{42}$$

into the previous relation, which leads to (28c). Finally, note that if N is not even, interchanging the integer part $\lfloor\cdot\rfloor$ and mod $N/2$ operators in (39) may lead to an error of ± 1 sample in the estimation of $p(\mathbf{z}')$ from $p(\mathbf{z})$, possibly causing an additional shift of ± 1 sample between $n_p(\mathbf{z})$ and $n_p(\mathbf{z}')$.

Appendix D: Proof of Proposition 4

If $\mathbf{s} = \mathbf{Q}\mathbf{s}'$ and $\mathbf{Q}$ is decomposed as in (29), then (30) holds and we can find a relation between the discrete Fourier transforms $\mathbf{u} = \Im(\mathbf{z})$ and $\mathbf{u}' = \Im(\mathbf{z}')$:

$$\begin{aligned} u'_k &= \sum_{i=0}^{N-1} z'_i w^{-ki} = s_x e^{\mathrm{j}\theta} \sum_{i=0}^{N-1} (x_i + \mathrm{j}\lambda y_i) w^{-ki} \\ &= \begin{cases} s_x e^{\mathrm{j}\theta} u_k, & \lambda = 1 \\ s_x e^{\mathrm{j}\theta} u^*_{N-k}, & \lambda = -1 \end{cases} \end{aligned} \tag{43}$$

for $k = 0, 1, \ldots, N-1$, where $\lambda = s_x s_y = \pm 1$. If we evaluate the phases $\mathbf{a} = \arg \mathbf{u}$ and $\mathbf{a}' = \arg \mathbf{u}'$, (43) leads to

$$\begin{aligned} a'_1 &= \begin{cases} a_1 + \theta + \arg s_x, & \lambda = 1 \\ -a_{N-1} + \theta + \arg s_x, & \lambda = -1 \end{cases} \\ a'_{N-1} &= \begin{cases} a_{N-1} + \theta + \arg s_x, & \lambda = 1 \\ -a_1 + \theta + \arg s_x, & \lambda = -1 \end{cases} \end{aligned} \tag{44}$$

Adding the previous two relations, we get $(a'_1 + a'_{N-1}) \bmod 2\pi = (\lambda(a_1 + a_{N-1}) + 2\theta) \bmod 2\pi$, since $s_x = \pm 1$ and thus $(2 \arg s_x) \bmod 2\pi = 0$. From the definition (31a) of $r(\mathbf{z})$, (33a) follows immediately.

Now evaluation of $\mathbf{z}'_1$ from (31b) yields $\mathbf{z}'_1 = \mathbf{z}' \, e^{-\mathrm{j}r(\mathbf{z}')} = s_x(\mathbf{x} + \mathrm{j}\lambda\mathbf{y})e^{\mathrm{j}(\theta - r(\mathbf{z}'))}$. If we define

$$l(\mathbf{z}) = \begin{cases} 1, & \lambda r(\mathbf{z}) + \theta \in [0, \pi) \\ -1, & \lambda r(\mathbf{z}) + \theta \in [\pi, 2\pi) \end{cases} \tag{45}$$

then from (33a) it follows that

$$\begin{aligned} \mathbf{z}'_1 &= s_x(\mathbf{x} + \mathrm{j}\lambda\mathbf{y})e^{\mathrm{j}(\theta - ((\lambda r(\mathbf{z}) + \theta) \bmod \pi))} \\ &= s_x l(\mathbf{z})(\mathbf{x} + \mathrm{j}\lambda\mathbf{y})e^{-\mathrm{j}\lambda r(\mathbf{z})} \\ &= \begin{cases} s_x l(\mathbf{z})\mathbf{z}_1, & \lambda = 1 \\ s_x l(\mathbf{z})\mathbf{z}^*_1, & \lambda = -1 \end{cases} \end{aligned} \tag{46}$$

so that $\mathbf{x}'_1 = s_x l(\mathbf{z})\mathbf{x}_1$ and $\mathbf{y}'_1 = \lambda s_x l(\mathbf{z})\mathbf{y}_1 = s_y l(\mathbf{z})\mathbf{y}_1$. Hence $\mathbf{x}'_1$ and $\mathbf{y}'_1$ differ from $\mathbf{x}_1$ and $\mathbf{y}_1$ only in their signs, and calculation of their third-order moments in (32a) gives $v_x(\mathbf{z}'_1) = s_x l(\mathbf{z}) v_x(\mathbf{z}_1)$ and $v_y(\mathbf{z}'{}_1) = s_y l(\mathbf{z}) v_y(\mathbf{z}_1)$. Subsequent normalization (32b) gives $\mathbf{x}'_2 = (s_x l(\mathbf{z}))^2 v_x(\mathbf{z}_1)\mathbf{x}_1 = \mathbf{x}_2$ and similarly $\mathbf{y}'_2 = \mathbf{y}_2$, and (33b) follows. Finally, (33c) can be directly verified using (32) and (33a).

Acknowledgement. This work was partially funded by the National Program "YPER" by the General Secretariat of Research and Development of Greece entitled "Efficient Content-Based Image and Video Query and Retrieval in Multimedia Systems".

References

Arkin EM, Chew LP, Huttenlocher DP, Kedem K, Mitchell JSB (1991) An efficiently computable metric for comparing polygonal shapes. IEEE Trans Pattern Anal Mach Intell 13: 209–216

Avrithis Y, Doulamis N, Doulamis A, Kollias S (1998) Efficient content representation in MPEG video databases. In: Proceedings of the IEEE workshop of Content-Based Access of Image and Video Libraries. Santa Barbara, Calif., 21 June, pp 91–95

Avrithis YS, Doulamis AD, Doulamis ND, Kollias SD (1999) A Stochastic framework for optimal key frame extraction from MPEG video databases. Comput Vis Image Understand 75: 3–24

Avrithis Y, Xirouhakis Y, Kollias S (2000) Affine-invariant curve normalization for shape-based retrieval. In: Proceedings of the International Conference on Pattern Recognition, Barcelona, Spain, 4–7 September, pp 1015–1018

Balslev I (1998) Noise tolerance of moment invariants in pattern recognition. Pattern Recog Lett 19: 1183–1189

Bimbo AD, Pala P (1997) Visual image retrieval by elastic matching of user sketches. IEEE Trans Pattern Anal Mach Intell 19: 121–132

Blum H (1967) A transformation for extracting new descriptors of shape. In: Wathen-Dum W (ed) Models for the perception of speech and visual form. MIT Press, Cambridge, Mass.

Chang S-F, Chen W, Meng HJ, Sundaram H, Zhong D (1998) A fully automated content-based video search engine supporting spatiotemporal queries. IEEE Trans Circuits Syst Video Technol 8: 602–615

Cohen FS, Huang Z, Yang Z (1995) Invariant matching and identification of curves using B-splines curve representation. IEEE Trans Image Process 4: 1–10

Doulamis AD, Avrithis YS, Doulamis ND, Kollias SD (1999) Interactive content-based retrieval in video databases using fuzzy classification and relevance feedback. In: Proceedings of the IEEE International Conference on Multimedia Computing and Systems, Florence, Italy, 7–11 June, pp 954–958

Doulamis ND, Doulamis AD, Avrithis YS, Ntalianis KS, Kollias SD (2000) Efficient summarization of stereoscopic video sequences. IEEE Trans Circuits Syst Video Technol 10: 501–517

Flickner M, Sawhney H, Niblack W, Ashley J, Huang Q, Dom B, Gorkani M, Hafner J, Lee D, Petkovic D, Steele D, Yanker P (1995) Query by image and video content: the QBIC system. IEEE Comput Mag 28: 23–32

Freeman H (1970) Boundary encoding and processing. In: Lipkin BS, Rosenfeld A (eds) Picture processing and psychopictorics. Academic, New York

Hamrapur A, Gupta A, Horowitz B, Shu CF, Fuller C, Bach J, Gorkani M, Jain R (1997) Virage video engine. In: SPIE Proceedings on Storage and Retrieval for Video and Image Databases V, San Jose, Calif., February, pp 188–197

Hu MK (1962) Visual pattern recognition by moment invariants. IRE Trans Inf Theory 8: 179–187

Huang Z, Cohen FS (1996) Affine-invariant B-spline moments for curve matching. IEEE Trans Image Process 5: 1473–1480

Ip HS, Shen D (1998) An affine-invariant active contour model (AI-snake) for model-based segmentation. Image Vis Comput 16: 135–146

ISO (1997) Text of ISO/IEC MPEG-7: context and objectives V5. ISO/IEC JTC1/SC29/WG11 N1920, Fribourg, October

ISO (1998) Text of ISO/IEC MPEG-4: video verification model V11.0. ISO/IEC JTC1/SC29/WG11 N2172, Tokyo, March

Jain AK, Zhong Y, Lakshmanan S (1996) Object matching using deformable templates. IEEE Trans Pattern Anal Mach Intell 18: 267–278

Jiang X, Bunke H, Widmer-Kljajo D (1999) Skew correction of document images by focused nearest-neighbor clustering. In: Proceedings of the Fifth International Conference on Document Analysis and Recognition, Bangalore, India, 20–22 September, pp 629–632

Khotanzad A, Hong YH (1990) Invariant image recognition by Zernike moments. IEEE Trans Pattern Anal Mach Intell 12: 489–497

Lai KK, Chin RT (1995) Deformable contours: modeling and extraction. IEEE Trans Pattern Anal Mach Intell 17: 1084–1090

Ma WY, Manjunath BS (1997) Netra: a toolbox for navigating large image databases. In: Proceedings of the International Conference on Image Processing, Santa Barbara, Calif., 26–29 October, pp 568–571

Marques F, Llorens B, Gasull A (1998) Prediction of image partitions using Fourier descriptors: application to segmentation-based coding schemes. IEEE Trans Pattern Anal Mach Intell 7: 529–542

Mokhtarian F (1995) Silhouette-based isolated object recognition through curvature scale space. IEEE Trans Pattern Anal Mach Intell 17: 539–544

Odobez JM, Bouthemy P (1997) Separation of moving regions from background in an image sequence acquired with a mobile camera. In: Li HH, Sun S, Derin H (eds) Video data compression for multimedia computing. Kluwer, Boston, pp 283–311

Pavlidis T, Ali F (1975) Computer recognition of handwritten numerals by polygonal approximation. IEEE Trans Syst Man Cybern 6: 610–614

Pentland A, Picard RW, Sclaroff S (1996) Photobook: content-based manipulation of image databases. Int J Comput Vis 18: 233–254

Persoon E, Fu KS (1986) Shape discrimination using Fourier descriptors. IEEE Trans Pattern Anal Mach Intell 8: 388–397

Pratt I (1996) Shape representation using Fourier coefficients of the sinusoidal transform. Technical Report Series UMCS-96-7-1. University of Manchester, U.K.

Rivlin E, Weiss I (1997) Deformation invariants in object recognition. Comput Vis Image Understand 65: 95–108

Rothe I, Susse H, Voss K (1996) The method of normalization to determine invariants. IEEE Trans Pattern Anal Mach Intell 18: 366–375

Rui Y, Huang TS, Mehrotra S (1997) Content-based image retrieval with relevance feedback in MARS. In: Proceedings of the International Conference on Image Processing, Santa Barbara, Calif., 26–29 October, pp 815–818

Rui Y, She A, Huang TS (1998) A modified Fourier descriptor for shape matching in MARS. In: Chang SK (ed) Image databases and multimedia search. (Series on software engineering and knowledge engineering, vol 8) World Scientific, Singapore, pp 165–180

Salembier P, Pardas M (1994) Hierarchical morphological segmentation for image sequence coding. IEEE Trans Pattern Anal Mach Intell 3: 639–651

Salembier P, Marques F, Pardas M, Morros R, Corset I, Jeannin S, Bouchard L, Meyer F, Marcotequi B (1997) Segmentation-based video coding system allowing the manipulation of objects. IEEE Trans Circuits Syst Video Technol 7: 60–73

Schmid C, Mohr R (1997) Local grayvalue invariants for image retrieval. IEEE Trans Pattern Anal Mach Intell 19: 530–535

Shen D, Ip H (1997) Generalized affine invariant image normalization. IEEE Trans Pattern Anal Mach Intell 19: 431–440

Sikora T (1997) The MPEG-4 video standard verification model. IEEE Trans Circuits Syst Video Technol 7: 19–31

Smith JR, Chang SF (1996) VisualSEEk: a fully automated content-based image query system. In: Proceedings of the ACM Multimedia Conference, Boston, Mass., 18–22 November, pp 87–98

Swanson MD, Tewfik AH (1997) Affine-invariant multiresolution image retrieval using B-splines. In: Proceedings of the International Conference on Image Processing, Santa Barbara, Calif., 26–29 October, pp 831–834

Taubin G, Cooper D (1992) Object recognition based on moment (or algebraic) invariants. In: Mundy JL, Zisserman A (eds) Geometric invariance in computer vision. MIT Press, Cambridge, Mass., pp 375–397

Tekalp M (1995) Digital video processing. Prentice Hall, Upper Saddle River, N.J.

Torres L, Kunt M (1996) Video coding: the second generation approach. Kluwer, Boston

Tsang PWM (1997) A genetic algorithm for affine invariant recognition of object shapes from broken boundaries. Pattern Recog Lett 18: 631–639

Wang J, Cohen FS (1994) Part II: 3-D object recognition and shape estimation from image contours using B-splines, shape invariant matching, and neural network. IEEE Trans Pattern Anal Mach Intell 16(1): 13–23

Wang J, Yang W-J, Acharya R (1998) Efficient access to and retrieval from a shape image database. In: Proceedings of the IEEE workshop of Content-Based Access of Image and Video Libraries. Santa Barbara, Calif., 21 June, pp 63–66

Watt A, Watt M (1992) Advanced animation and rendering techniques. ACM Press, New York

Xirouhakis Y, Avrithis Y, Kollias S (1998) Image retrieval and classification using affine invariant B-spline representation and neural networks. In: Proceedings of the IEE Colloquium on Neural Nets and Multimedia, London, 22 October, pp 4/1–4/4

Xirouhakis Y, Mathioudakis V, Delopoulos A (1999a) An efficient algorithm for mobile object localization in video sequences. In: Proceedings of the Visual, Modeling and Visualization Workshop, Erlangen, Germany, 17–19 November, pp 117–124

Xirouhakis Y, Tirakis A, Delopoulos A (1999b) An efficient graph representation for image retrieval based on color composition. In: Mastorakis N (ed) Advances in intelligent systems and computer science. World Scientific and Engineering Society, New York, pp 336–341

Yang Z, Cohen FS (1999) Image registration and object recognition using affine invariants and convex hulls. IEEE Trans Pattern Anal Mach Intell 8: 934–946

Yannis S. Avrithis was born in Athens, Greece, in 1970. He received the Diploma degree in Electrical and Computer Engineering from the National Technical University of Athens (NTUA) in 1993, the M.Sc. degree in Electrical and Electronic Engineering (Communications and Signal Processing) from the Imperial College of Science, Technology and Medicine, London, U.K., in 1994, and the Ph.D. degree in Electrical and Computer Engineering from NTUA in 2001. He is currently a senior researcher at the Image, Video and Multimedia Systems Laboratory of NTUA. His research interests include digital image and video processing, image segmentation, affine-invariant image representation, content-based indexing and retrieval, and video summarization. He has published 7 articles in international journals and books, and 17 in proceedings of international conferences.

Yiannis S. Xirouhakis was born in Athens, Greece, in 1975. He received his degree in Electrical and Computer Engineering from the National Technical University of Athens (NTUA) in 1997. Currently he is a Ph.D. candidate in the Image, Video and Multimedia Systems Laboratory of the Electrical and Computer Engineering Department of NTUA. His main area of research includes computer vision and video understanding; more precisely, 3-D motion and structure estimation, object recognition, and video indexing. As a student he received several national awards in Greece in mathematics and physics. He has published 25 articles in international journals and conferences in his area of research.

Stefanos D. Kollias was born in Athens, Greece, in 1956. He received the Diploma degree in Electrical Engineering from the National Technical University of Athens (NTUA) in 1979, the M.Sc degree in Communication Engineering from the University of Manchester (UMIST), U.K., in 1980, and the Ph.D degree in Signal Processing from the Computer Science Division of NTUA in 1984. In 1982 he received a ComSoc Scholarship from the IEEE Communications Society. Since 1986 he has been with the NTUA where he is currently a Professor. From 1987 to 1988, he was a Visiting Research Scientist in the Department of Electrical Engineering and the Center for Telecommunications Research, Columbia University, New York, U.S.A. His current research interests include image processing and analysis, neural networks, image and video coding, and multimedia systems. He is the author of more than 140 articles in the aforementioned area.