A Multiscale Representation Method for Nonrigid Shapes With a Single Closed Contour

Tomasz Adamek and Noel E. O'Connor, *Member, IEEE*

***Abstract*—In this paper, we discuss the criteria that should be satisfied by a descriptor for nonrigid shapes with a single closed contour. We then propose a shape representation method that fulfills these criteria. In the proposed approach, contour convexities and concavities at different scale levels are represented using a two-dimensional (2-D) matrix. The representation can be visualized as a 2-D surface, where "hills" and "valleys" represent contour convexities and concavities, respectively. The optimal matching of two shape representations is achieved using dynamic programming and a dissimilarity measure is defined based on this matching. The proposed algorithm is very efficient and invariant to several kinds of transformations including some articulations and modest occlusions. The retrieval performance of the approach is illustrated using the MPEG-7 shape database, which is one of the most complete shape databases currently available. Our experiments indicate that the proposed representation is well suited for object indexing and retrieval in large databases. Furthermore, the representation can be used as a starting point to obtain more compact descriptors.**

***Index Terms*—Dynamic programming, matching, shape descriptor, similarity metric.**

I. Introduction

THE recognition of objects in multimedia content is a challenging task and a complete solution requires the combination of multiple elementary visual and audio features via a knowledge-based inferencing process. Typically, visual features which provide complementary information, such as shape, color, and texture, are chosen. Shape is clearly an important cue for recognition since humans can often recognize characteristic objects solely on the basis of their shapes. This distinguishes shape from other elementary visual features, which usually do not reveal object identity. A user survey presented in [1] regarding cognition aspects of image retrieval indicates that users are more interested in retrieval by shape than by color and texture. From this, it can be deduced that shape descriptors for comparing two-dimensional (2-D) silhouettes in order to determine their similarity are an important and useful building block for applications requiring object recognition and object-based indexing/retrieval from multimedia databases.

Many shape analysis techniques have been proposed over the past three decades. An extensive survey of shape matching in computer vision can be found in [2] and [3]. Most shape matching methods attempt to quantify shape in ways aligned with human intuition. The central question to be addressed can be stated as: what constitutes similarity of shape? Common criteria used for shape representation for reliable shape matching and retrieval include: uniqueness, invariance to translation, scale, rotation and symmetric transformations, scalability, compactness, extraction and matching efficiency, and robustness to different kinds of distortions [4]. Shape distortions may be present as a result of a perspective transformation due to a change in viewing angle or may be a side effect of the segmentation or extraction process used. Some objects are flexible and thus the technique should be able to cope with nonrigid deformations (e.g., object shapes corresponding to people walking). In [5], it is concluded that it is not difficult to compare very similar shapes but that difficulties arise when we have to measure the degree of similarity of two shapes which are significantly different. All of these criteria can be summarized by stating that a good shape descriptor should enable the similarity between shapes to be quantified in a way aligned with human intuition. However, when evaluating the performance of different shape analysis techniques, many researchers [5], [6] have reached the conclusion that human judgements of shape similarity differ significantly between observers, thereby making the evaluation task very difficult.

Even in the case of 2-D silhouettes, there can be two notions of similarity. Objects having similar spatial distributions can have very different outline contours and vice versa. Therefore, posing a query using region-based or contour-based criteria can result in retrieving different objects [7].

A contour-based descriptor encapsulates the shape properties of the object's outline (silhouette). It should distinguish between shapes that have similar region-shape properties but different contour-shape properties. Such descriptors are usually very efficient in applications where high intraclass variability in the shape is expected, due to deformations in the object (rigid or nonrigid) or perspective deformations [7]. In this paper, we are primarily interested in contour-based similarity especially in the case where the similarity between the two curves is weak.

Since silhouettes do not have holes or internal markings, the associated boundaries are conveniently represented by a single closed curve which can be parameterized by arc length. Early work used Fourier descriptors [5] which are easy to implement and are based on the well-developed theory of Fourier analysis. The disadvantage of this approach is that, after the Fourier transform, local shape information is distributed to all coefficients and not localized in the frequency domain [2]. A representation that has proven to be relevant in human vision is the medial axis transform (MAT) originally proposed by Blum [8], [9]. This approach led Sebastian *et al.* [10] to attempt to capture the structure of the shape in the graph structure of the skeleton. In this

Manuscript received April 30, 2003; revised August 11, 2003. This work was supported by the Informatics Research Initiative of Enterprise, Ireland.

The authors are with the School of Electronic Engineering, Centre for Digital Video Processing, Dublin City University, Dublin 9, Ireland (e-mail: adamekt@eeng.dcu.ie; oconnorn@eeng.dcu.ie).

Digital Object Identifier 10.1109/TCSVT.2004.826776

1051-8215/04$20.00 © 2004 IEEE

approach, the edit distance between shock graphs is used as a metric of similarity between shapes. Sebastian *et al.* claim that their approach is robust to various transformations including articulation, deformation of parts, and occlusion. The disadvantage of this description method is that matching two graphs is very computationally expensive. Given the one-dimensional (1-D) nature of silhouette curves, a natural alternative was to use dynamic programming approaches which use the edit distance between curves for matching [11]–[14], [4]. These algorithms are usually fast and invariant to several kinds of transformations including some articulation and occlusion.

A comprehensive comparison of different shape descriptors for comparing silhouettes was carried out as part of the MPEG-7 standardization activity [15], [16]. As a result of this comparison, two leading approaches emerged: one by Latecki *et al.* [17], [18] based on the best possible correspondence of visual parts and a second approach developed by Mokhtarian *et al.* [19]–[21], [6] based on the curvature scale space (CSS) representation. Both descriptors base the computation of a similarity measure on the best possible correspondence between maximal convex/concave arcs contained in simplified versions of boundary contours. The simplified boundary contours are obtained by two different processes of curve evolution. Both descriptors are cognitively motivated, since convex/concave arcs play an important role in human visual perception.

The main motivation for this study was provided by a scale space technique for plane curves description proposed by Mokhtarian and Mackworth in [19]. In this method, the arc-length parameterized contour $C(s)$ is convolved with a Gaussian kernel of width σ. With increasing value of σ, the resulting contour becomes smoother, and the number of zero crossings of the curvature along the contour decreases until finally the contour is convex and the curvature is positive. For continuously increasing values of σ, the positions of the curvature zero-crossing points continuously move along the contour, until two such positions meet and annihilate. The CSS image of each boundary is computed and then the maxima of the CSS contours are used as a shape descriptor for each object. Matching of two objects is achieved by matching points of annihilation in the s, σ plane [19]–[21], [6]. The resulting shape descriptor is very compact and extensive tests [16] reveal that the method is quite robust with respect to noise, scale and orientation changes of objects. Because of these advantages, the CSS shape descriptor has been included as one of two shape descriptors in the ISO/IEC MPEG-7 standard. As shown in [22], however, it does not always give results aligned with human intuition. The main drawback of this description is its potentially high degree of ambiguity. The positions of zero-crossing point maxima for very deep and sharp concavities and for very long shallow concavities are almost identical. The convex parts of the curve are represented only implicitly by assuming that every concavity must be surrounded by two convexities. As such, it is impossible to use the CSS descriptor to distinguish between totally convex curves (i.e., circles, squares, and triangles).

Given the above observations, we conclude that a representation with enhanced discriminatory capabilities could be obtained by explicitly representing the position and depth of both convexities and concavities. To evaluate this idea we introduce a new shape description method termed multi-scale convexity concavity (MCC) representation, where for each contour point we store information about the amount of convexity/concavity at different scale levels.

The remainder of paper article is organized as follows. In Section II, the relationship between our method and other curve-matching methods is discussed. Section III describes the new representation in detail and the robustness of the proposed scheme is discussed. Section IV presents an optimal solution for matching two MCC representations and a definition of the dissimilarity measure used. The computational complexity of the proposed approach is discussed in Section V. Section VI presents the results obtained using our approach, including a comparison with the CSS approach in our own simulation of the "CE-Shape" MPEG-7 core experiment. Finally, conclusions are formulated and presented in Section VII.

II. Related Work

As far as we are aware, the use of a measure of convexity/concavity for every contour point at multiple scales as a means to matching 2-D curves is novel. The most closely related idea in previous work is that of Quddus *et al.* in their work on wavelet-based multilevel techniques for contour description and matching [22]. This technique uses simple features extracted at high curvature points. These points are detected at each level of the wavelet decomposition as wavelet transform modulus maxima (WTMM). During the matching process, the similarity score is computed at each level and the final similarity score is computed as the mean of the scores at each level.

The idea of using dynamic programming (DP) for matching whole contours or parts thereof is not new [11], [12]. Multiscale methods have been combined with dynamic programming in the past [13]. Recently, very powerful matching algorithms have been proposed by Gdalyahu and Weinshall [14] and by Petrakis *et al.* [4]. All of these algorithms are quite complicated. Usually some edit transformation which maps one curve to the other is defined, and merging, skipping, or stretching parts of the curves is penalized during the matching process.

The main advantage of all of these approaches is their robustness to occlusions. Some allow uniform matching of the open and closed curves [4]. Our matching approach is rather simple and originates directly from dynamic time warping (DTW) [23] as used in speech recognition. We assume that the contours are closed. Occlusions are not detected explicitly but penalized implicitly by the cost of matching them to the nonoccluded contour. There is no extra penalty for merging or skipping parts of the contours during the matching process. Rather, a global optimal match between two contours is found using a rich multiscale feature for each contour point.

III. Proposed Shape Representation

A. Definition

We propose a new rich multiscale shape representation, where for each contour point we store information about the

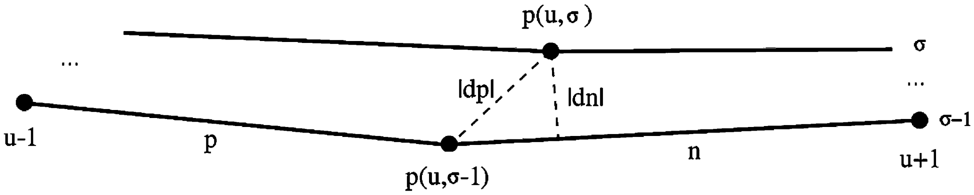

Fig. 1. Calculation of the contour displacement at two consecutive scale levels.

convexity/concavity at different scale levels. The representation can be stored in the form of a 2-D matrix where the columns correspond to contour points (contour parameter u) and the rows correspond to the different scale levels σ. The position (u, σ) in this matrix contains information about the degree of convexity or concavity for the uth contour point at scale level σ.

The simplified boundary contours at different scale levels are obtained via a curve evolution process very similar to that used in [19] to extract the CSS image. We assume that a size normalized contour C is represented by N contour points. It should be noted that we use the same number of contour points for each shape to be compared (resulting in an $N \times N$ distance table—see Section IV-B). Let the contour C be parameterized by arc length u: $C(u) = (x(u), y(u))$, where $u \in \langle 0, N \rangle$. The coordinate functions of C are convolved with a Gaussian kernel ϕ_σ of width $\sigma \in \{1, 2 \ldots \sigma_{\max}\}$ as follows:

$$x_\sigma(u) = \int x(u)\phi_\sigma(t-u)\,dt \tag{1}$$

$$\phi_\sigma(t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{t^2}{2\sigma^2}} \tag{2}$$

and similarly for $y(u)$. The resulting contour C_σ becomes smoother with increasing value of σ, until finally the contour is convex. Note that σ is constrained to assume integer values only.

We propose a very simple measure for the convexity and concavity of the curve. The measure is defined as the displacement of the contour between two consecutive scale levels. If we denote the contour point u at scale level σ as $p(u, \sigma)$, the displacement of the contour between two consecutive scale levels $d(u, \sigma)$ at point $p(u, \sigma)$ can be defined as the Euclidian distance between position of $p(u, \sigma)$ and $p(u, \sigma - 1)$ as follows:

$$d(u, \sigma) = k\sqrt{(x_\sigma(u) - x_{\sigma-1}(u))^2 + (y_\sigma(u) - y_{\sigma-1}(u))^2} \tag{3}$$

where

$$k = \begin{cases} 1, & \text{if } p(u, \sigma - 1) \text{ inside } C_\sigma \\ -1, & \text{otherwise.} \end{cases} \tag{4}$$

$d(u, \sigma)$ is the distance by which contour point u has moved between scale level $\sigma - 1$ and σ. The sharper the convexity or concavity, the larger the change in the position of its contour points during filtering. Equation (4) distinguishes between convex and concave parts of the contour via a change in sign. It is assumed that the distance $d(u, \sigma)$ will have positive (negative) values for u at convex (concave) parts of the contour. If $p(u, \sigma)$ lies inside the contour produced at level $\sigma - 1$ this indicates that the contour at $p(u, \sigma)$ is convex ($d(u, \sigma)$ has positive value). If $p(u, \sigma)$ lies outside the contour produced at level $\sigma - 1$ this indicates that the contour at $p(u, \sigma)$ is concave ($d(u, \sigma)$ has negative value).

For a small class of shapes, direct implementation of (3) can result in a loss of continuity for values of $d(u, \sigma)$ at the borders between convexities and concavities. At high scale levels, where the filtered contours are very smooth, some contour points move in a direction roughly parallel to the contour rather than in a direction perpendicular to the contour. In this case, the distance between successive positions of the contour point does not reflect the displacement of the contours between two scale levels—an example is shown in Fig. 1. This problem can be easily avoided by taking the displacement measure as the locally minimum distance between contour point $p(u, \sigma)$ and neighboring segments of this contour point at scale level $\sigma - 1$. Theoretically, the search for the minimum distance should continue until the first local minimum distance is found. In a practical implementation, it is often sufficient to take the displacement measure as the minimum of two distances $|d_p|$ and $|d_n|$, where $|d_p|$ and $|d_n|$ denote the Euclidian distances between $p(u, \sigma)$ and segments $\langle u - 1, u \rangle$ and $\langle u, u + 1 \rangle$ at scale level $\sigma - 1$.

An example of the shape representation is illustrated in the form of a 2-D surface in Fig. 2. Fig. 2(a) shows the original shape. Fig. 2(b) shows filtering the original contour with different values of σ. The arrows indicate the displacement direction for convex and concave contour points respectively. All concavities and convexities are present at the lowest scale levels. Moving to higher scale levels the level of detail decreases and only the most significant convexities and concavities are retained. Our observations indicate that the majority of contours become completely indistinguishable above the tenth scale level. As a result, in all our experiments we use shape representations with scale levels from 1 to 10.

Note that the curvature could be employed as an alternative convexity/concavity measure as widely used in many other shape analysis methods [19]. However, whilst our experiments show that the use of curvature results in a descriptor with strong discriminatory capabilities, sometimes small changes in the contour lead to large changes in the curvature and such curvature extremes can dominate the matching process.

B. Invariance and Robustness

A shape descriptor should be invariant to scaling, translation, rotation and should be robust in the presence of small geometrical distortions, occlusion and outliers. In this section, we evaluate the MCC representation according to these criteria.

1) Linear Transformations: Translation of the contour does not affect the MCC representation since the convexity/concavity

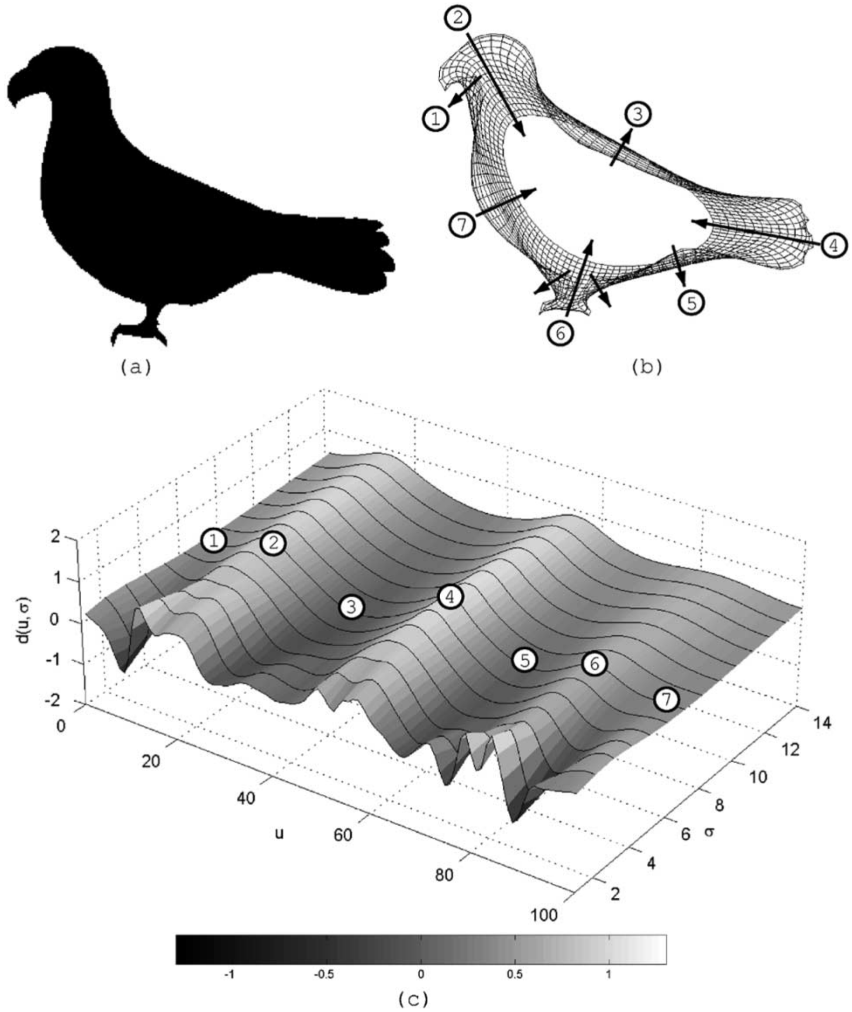

Fig. 2. Example of extracting the MCC shape representation: (a) original shape image; (b) filtered versions of the original contour at different scale levels; (c) final MCC representation for 100 contour points at 14 scale levels.

measure for a given contour point is calculated solely with the respect to its neighborhood points. A size normalization process is employed to ensure that the approach is invariant with respect to scaling. A rotation of the object usually causes a circular shift of its representation (along the u axis) which is taken into account during the matching process—see Section IV. This also addresses the possible effects of a change in the contour starting point used.

2) Nonlinear Transformations: In this section, we illustrate the influence of shape deformations on the proposed shape representation using a very simple shape as an example. Fig. 3 shows four different shapes representing digit one and their corresponding MCC representations. In fact, the shapes in Fig. 3(b), (c) and (d) are manually modified (deformed) versions of the shape in Fig. 3(a).

Fig. 3(a) illustrates that the corresponding shape has three convexities, labeled $\{1\}, \{2\}$, and $\{4\}$ and one deep concavity, labeled $\{5\}$. One straight segment, labeled $\{3\}$, is also illustrated (other shorter straight segments present in the shape are not labeled). In the MCC representation, the convexities correspond to three maxima. Similarly, the concavity is represented by a single deep minimum. Shallow plateaus correspond to straight segments between convex parts. Modifying the top of the digit, see Fig. 3(b), makes the concavity shallower. This results in a slightly shallower minimum in the MCC representation, especially at higher scale levels. The shape in Fig. 3(c) has a longer top part compared to the original version. In the MCC representation, this results in a larger separation between maxima/minima at all scale levels and the minimum becomes deeper, especially at higher scale levels. Introducing small outliers, as in Fig. 3(d), introduces new maxima/minima in the MCC representations at lower scale levels. Similarly, contour noise (not shown in Fig. 3) introduces additional convexities and concavities at lower scale levels, but higher scale levels usually remain unaffected due to the implicit filtering process. In all cases, there are small differences in the position of the maxima/minima along the contour—a fact which must be addressed during the shape matching process.

Further examples of the MCC shape representations are shown in Fig. 9. Of particular interest are the differences in the MCC representation between rigid shapes like *circle*, *square*, and *triangle*. In order to compare the MCC representation to the CSS representation, it should be noted that the CSS representation of these shapes would contain no maxima. The MCC representation of shapes, which could be considered elastic, like *pen*, and *horseshoe* is also illustrated in Fig. 9.

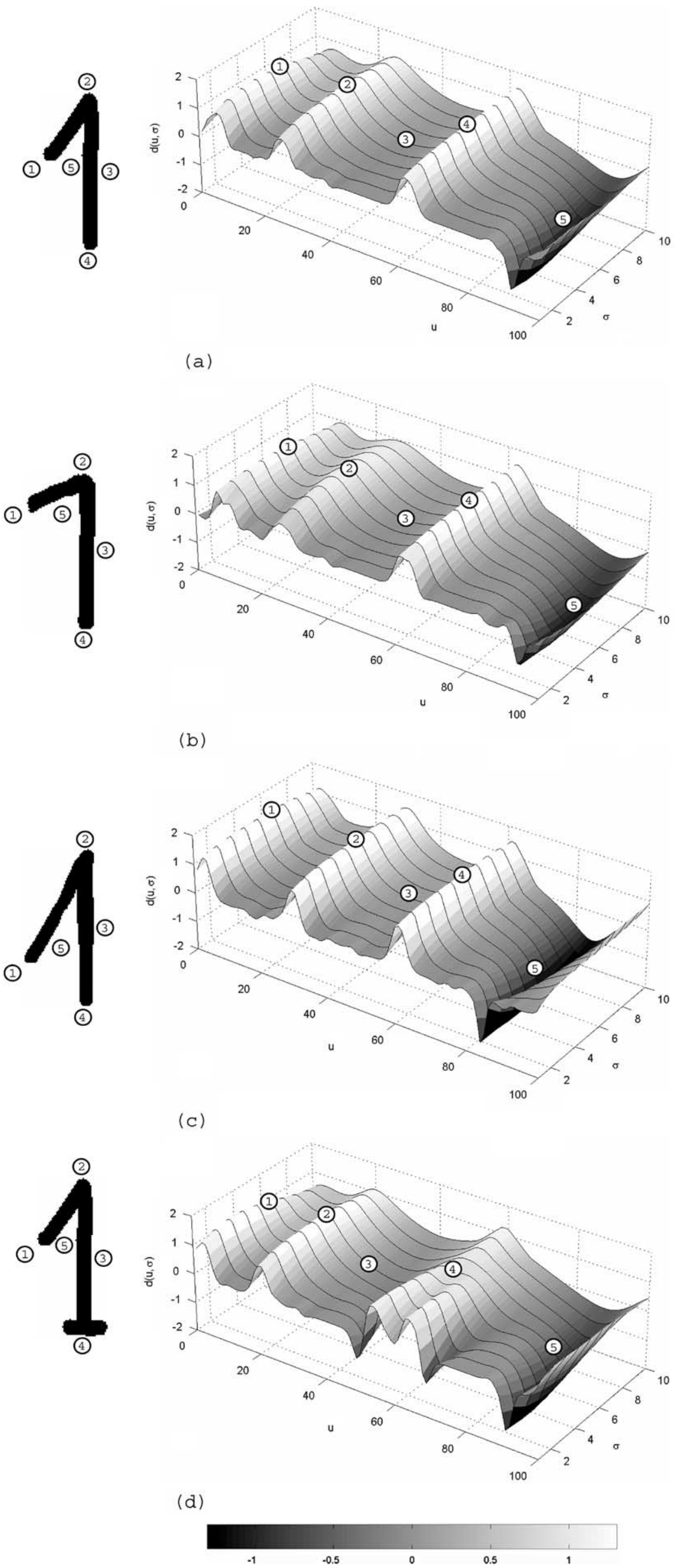

Fig. 3. Examples of shape deformations and their influence on the MCC representation: (a) original shape of digit one; (b) original shape with the top part moved up; (c) original shape with the top part longer; (d) original shape with outliers.

We can conclude from the above that the proposed description scheme is very robust to elastic deformations and that changes in the MCC representation could be used for quantifying these differences in a way which conforms with human intuition. While we have no theoretical justification of this, this assumption is evaluated experimentally in Section VI.

IV. Matching and Similarity Measure

In this section, a contour matching method and dissimilarity measure based on the proposed MCC representation are presented. It should be stressed that although the presented algorithm is quite efficient, much faster heuristic matching approaches could be developed in the future. The presented algorithm was developed to find the global optimal match between two MCC representations of closed contours and is described here to illustrate the full capabilities of the proposed descriptor. However, more advanced matching schema, including partial matching [11], [12], [14], [4], are also possible.

A. Distance Between Contour Points

When comparing two contours A and B, it is necessary to examine the distance between each contour point of both contours. If two contour points u_A and u_B are represented by their multi-scale features $d_A(u_A,\sigma)$ and $d_B(u_B,\sigma)$ respectively, for $\sigma = 1, 2, \ldots, K$, then the distance between the two contour points can be defined as:

$$d(u_A, u_B) = \frac{1}{K}\sum_{\sigma=1}^{K} |d_A(u_A,\sigma) - d_B(u_B,\sigma)| \tag{5}$$

It should be noted that according to the above definition, all scale levels are currently incorporated in the distance measure with the same importance (weight). It is possible that using different weights could result in improved overall retrieval accuracy, however this requires significant further experimentation and will be the basis of our future work in this area.

B. Optimal Solution for Contour Point Correspondence

As part of the matching process, the best correspondence between contour points must be determined. The matching process must determine the optimal circular shift between two representations and take into account small variations in the relative positions of feature vectors along the contour length caused by contour deformation. We assume that the contours are closed. Occlusions are not detected explicitly, but penalized implicitly by the cost of matching them to the nonoccluded contour. This assumption is motivated by the observation that local contour occlusions and outliers only locally affect the MCC representation, making global alignment still possible.

These simplifications allow us to use a *Dynamic Programming* technique similar to one used for speech recognition where it is referred to as *Dynamic Time Warping* (DTW) [23]. DTW has been widely used for speech recognition where several utterances of the same word are likely to have different durations, and utterances of the same word with the same duration will differ due to different parts of the words being spoken at different rates. As such, a time alignment is performed to obtain a global distance between two speech patterns. In our case, the task is to find an optimal global alignment along the contour.

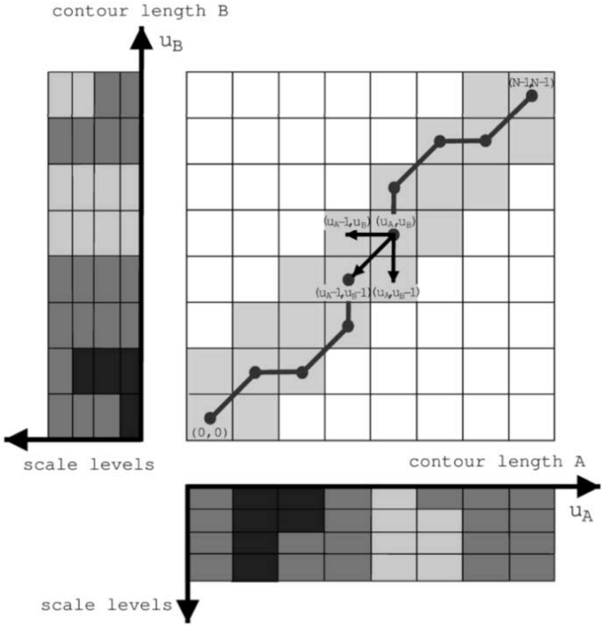

Fig. 4. Illustration of matching two MCC representations by dynamic programming.

We use an $N \times N$ distance table to conveniently examine the distances between corresponding contour points on both shapes. The columns represent contour points of one shape representation and the rows represent the contour points of the other. Each row/column entry in the table is the distance between two corresponding contour points calculated according to (5).

Finding the optimal match between the columns corresponds to finding the lowest cost diagonal path through the distance table—see the example in Fig. 4 where the contours' feature vectors are illustrated as grey levels along each axis. If one of the contours is rotated with respect to another or different starting points are chosen for the two contours, the optimal path will be shifted in the distance table. Thus, rotation invariance can be obtained by checking N possible circular shifts for the optimal diagonal path. If one of the contours is a mirrored version of the other, the optimal path will become perpendicular to the optimal path for a nonmirrored version. Invariance to mirror transformations can be obtained by checking for the optimal path by traversing through the columns or rows in both directions. The final dissimilarity is calculated based on the cumulative cost along the optimal path.

1) Properties of the Optimal Path: In a dynamic programming formulation, the sequential nature of the contours is maintained as the path is traced through the distance table. This imposes the condition that matching paths cannot move backward along the columns. The path may only proceed upwards and to the right in the table—see Fig. 4. A global distance measure for the path under investigation is obtained by summing the local distance measures.

We impose the additional condition that every column in the input MCC representations must be used in calculating the matching path. Furthermore a column from one contour can only be assigned to a maximum of two columns from the other contour. This ensures that long horizontal or vertical segments

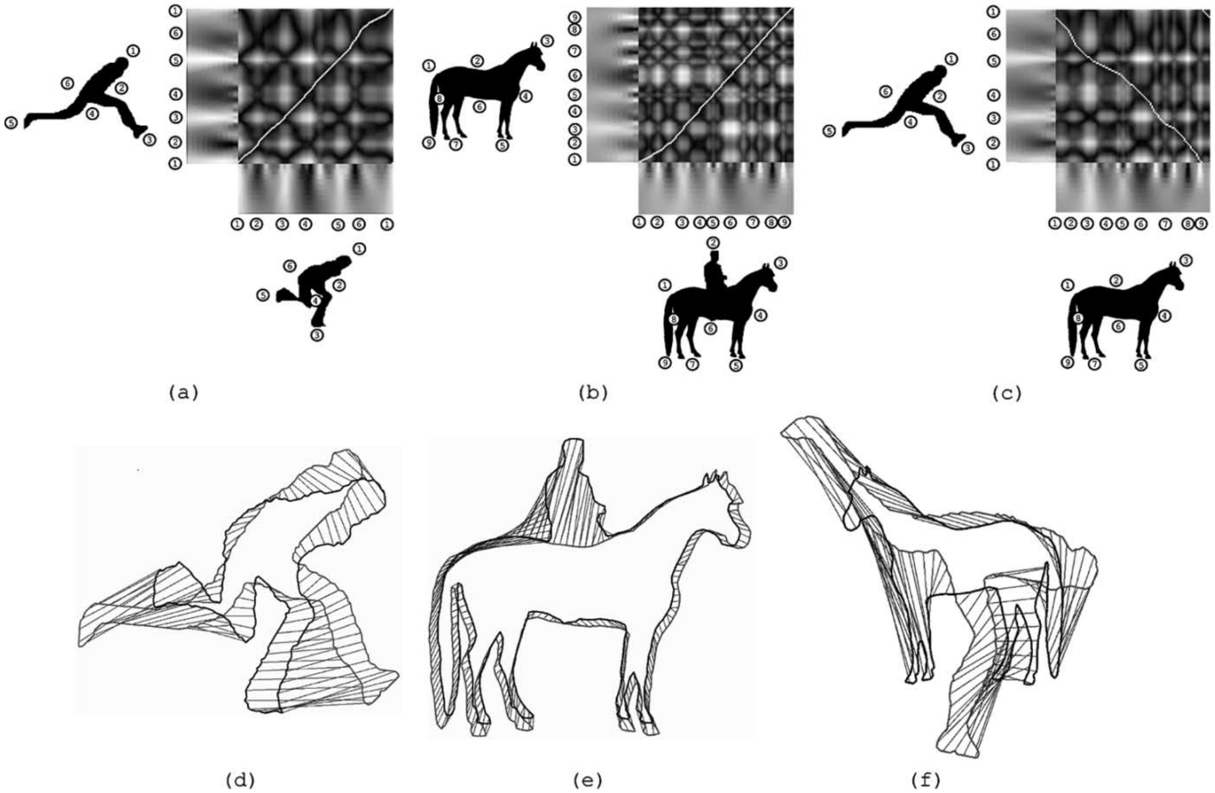

Fig. 5. Examples of typical matching: (a) distance table and the optimal path found for MCC representations of *stef09* and *stef13*, (b) distance table and the optimal path found for MCC representations of *horse14* and its modified version, (c) distance table and the optimal path found for MCC representations of *stef09* and *horse14*, (d) contour point correspondences for contours from (a), (e) contour point correspondences for contours from (b), (f) contour point correspondences for contours from (c).

in the path are avoided, which would correspond to a single contour point being matched to a large contour segment.

As it is common practise in DTW algorithms [23], we restrict the path to lie in the area close to the ideal straight diagonal line in order to speed up the matching process. The path is allowed to deviate from a straight diagonal path by no more than a chosen deviation threshold $\pm T_d$. In our experiments wechose $T_d = 0.04N$, which in most cases has no affect on the optimal path as a result of the restriction imposed on the path's geometry. This value is also a good trade off between speed and recognition efficiency.

2) Dynamic Programming Formulation: To describe the dynamic programming formulation, let us initially assume that the starting points in both MCC representations are known and shifted to position 0 along the contours. Let us also define $D(u_A, u_B)$ to be the total distance along the minimum distance path starting from cell (0, 0) to cell (u_A, u_B). The value of $D(u_A, u_B)$ at each point of the path may be evaluated as:

$$D(u_A, u_B) = d(u_A, u_B) + \min \begin{cases} D_p(u_A - 1, u_B) \\ D_p(u_A - 1, u_B - 1) \\ D_p(u_A, u_B - 1) \end{cases} \tag{6}$$

where $D_p(u_A-1, u_B)$, $D_p(u_A-1, u_B-1)$, and $D_p(u_A, u_B-1)$ denote the total distances for the left, bottom-left, and bottom predecessors respectively. The constraint that a column from one contour can only be assigned to a maximum of two columns from the other contour, means that the total distance for left and bottom predecessors is defined as:

$$\begin{aligned} &D_p(u_A - 1, u_B) \\ &\quad = \begin{cases} D(u_A - 1, u_B) & \text{if } \mathrm{pred}_V(u_A - 1, u_B) \neq u_B \\ \infty & \text{otherwise} \end{cases} \end{aligned} \tag{7}$$

$$\begin{aligned} &D_p(u_A, u_B - 1) \\ &\quad = \begin{cases} D(u_A, u_B - 1) & \text{if } \mathrm{pred}_H(u_A, u_B - 1) \neq u_A \\ \infty & \text{otherwise} \end{cases} \end{aligned} \tag{8}$$

where $\mathrm{pred}_V(u_A, u_B)$ and $\mathrm{pred}_H(u_A, u_B)$ denote vertical and horizontal positions of the lowest cost predecessor for cell (u_A, u_B). Note, that implementation of the last equation requires storing the position of the lowest cost predecessor for each cell of the DP table.

The DP matching algorithm starts at cell (0, 0) and proceeding upwards and to the right through the DP table, the costs of all possible allowable paths are updated according to Formula (6). The least cost path through the distance table, corresponding to the best matching between two MCC representations assuming a given pair of starting points, is the total cost at the top right cell $D(N - 1, N - 1)$.

3) Matching Algorithm: Since the starting points and rotation alignment for both contours are in fact unknown, it is therefore necessary to investigate all possible starting points. This means that the DP algorithm must be repeated N times with different contour points chosen as starting points. If the DP matching algorithm starts from a cell from the bottom row of distance table other than (0, 0), the optimal path may need to wrap around from the last column to the first column of the sequence of contour points. This may be efficiently implemented by repeating the columns of the distance table, resulting in an extended distance table of size $2N \times N$ [11]. The DP algorithm can then be run iteratively for $i = 0, 1, \ldots N$, each time determining the optimal path between cells $(i, 0)$ (bottom row) and $(i + N - 1, N - 1)$ (top row) in the extended distance table. Invariance to a mirror transformation can be obtained by flipping the columns of the extended distance table and repeating the search for the optimal path.

The entire algorithm for determining the least cost path through the distance table can be stated as follows:

```
INPUT: MCC representations of shapes $A$ and $B$;
OUTPUT: The least cost $D_{\min}$ of matching $A$ and $B$;
  1) SET $D_{\min} = \infty$
  2) FILL N × N Distance Table using (5);
  3) EXTEND Distance Table to 2N × N by repeating columns;
  4) FOR $i = 0, 1, \ldots N$ DO //Search for the least cost path
      • FIND the lowest cost path between cells $(i, 0)$ and
        $(i + N - 1, N - 1)$ proceeding upwards and to the right
        through Distance Table and summing local distances using
        (6);
        //Check if the cost of the new path is lower than $D_{\min}$
      • IF $D_{\min} > D(i + N - 1, N - 1)$
        THEN $D_{\min} = D(i + N - 1, N - 1)$;
    END FOR
  5) FLIP the columns of Distance Table and REPEAT 4;
```

The above algorithm finds the minimum cost $D_{\min}$ of traversing the distance table for all possible N starting points and taking into account mirror transformation.

C. Matching Examples

Matching examples are shown in Fig. 5. Fig. 5(a) shows the optimal path in the distance table between MCC representations of two shapes from the MPEG-7 dataset: *stef09* and *stef13*. It can be seen that despite significant nonrigid deformations between these shapes, the optimal path deviates only minimally from the straight diagonal line (part $\{6\}$), resulting in very intuitive contour point correspondence between the shapes—illustrated in Fig. 5(d). This example confirms that the MCC representation is very robust to nonrigid deformations. Fig. 5(b) shows an example of matching the shapes *horse14* from the MPEG-7 dataset and a modified version of this shape with a rider on the horse. From Fig. 5(b), which shows the optimal path found in the distance table for these shapes, it can be seen that deviation of the path for parts $\{1\}$ and $\{2\}$ of the contours, corresponds to matching the rider from one contour to the horse's back from the other contour—see Fig. 5(e). The remainder of the path is almost straight indicating that the remaining part of the MCC representation is almost unaffected even in the presence of significant local outliers. The cost of matching the rider to the back of the horse is used to quantify the dissimilarities between two shapes. An example of matching two very dissimilar shapes, corresponding to *stef09* and *horse14*, is illustrated in Fig. 5(c) with the contour point correspondence obtained illustrated in Fig. 5(f). Despite significant differences between both shapes, the matching algorithm is able to find an intuitive match. Note that the best match is found for a mirrored version of shape *horse14*. Of course, even such globally optimal matching will result in a large dissimilarity measure between these shapes.

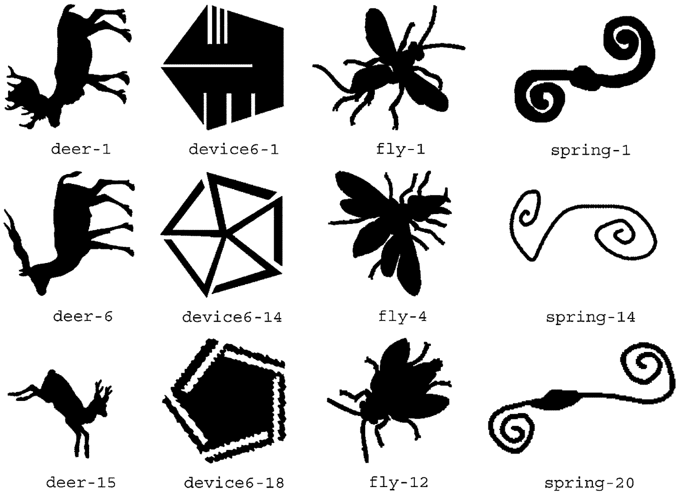

Fig. 6. Example shapes used in part B of "CE-Shape-1" MPEG-7 core experiment.

These examples illustrate the fact that the DP algorithm compensates for small deviations in the position of the convexities/concavities along the contour and does not merge or skip outliers present in one or both contours.

D. Final Dissimilarity Measure

The final dissimilarity measure between two contours is calculated based on the least cost path $D_{\min}$ found in the dynamic programming algorithm. This cost is normalized by the number of contour points N used in the MCC representation. Additionally, the measure is normalized according to the estimated complexity of both shapes. The introduction of the second normalization is motivated by our observations that humans are generally more sensitive to contour deformations when the complexity of the contour is lower. Thus, similarity between low complexity contours should be additionally penalized. The complexity of the contour is estimated as the average of the differences between maximum and minimum convexity/concavity measures over all scale levels:

$$C = \frac{1}{K} \sum_{\sigma=1}^{K} |\max_{u}\{d(u,\sigma)\} - \min_{u}\{d(u,\sigma)\}|. \tag{9}$$

Finally, we define the dissimilarity measure between two contours A and B as

$$D(A, B) = \frac{2 \cdot D_{\min}}{N \cdot (C_A + C_B)} \tag{10}$$

where C_A and C_B denote complexity of contour A and B respectively.

It should be noted that there is no extra penalty for stretching and merging of contour parts during the matching process. Rather, a global optimal match between two contours is found

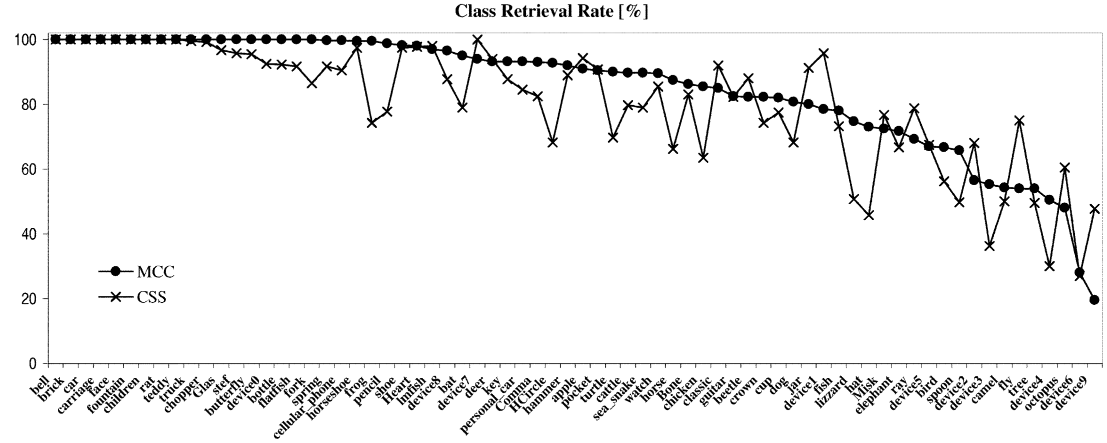

Fig. 7. Retrieval rate for all classes from MPEG-7 core experiment (CE-Shape1) part B for both MCC and CSS (sorted by retrieval rate of MCC).

using a rich multi-scale feature for each contour point. This is contrary to the majority of other contour matching approaches based on dynamic programming where different penalty factors, usually chosen in an *ad hoc* fashion [14], drive the matching process.

V. Computational Complexity

In this section we analyze the complexity for the MCC representation extraction stage and the matching stage independently, as they are usually performed separately.

The description extraction mainly involves iterative filtering of the original contour and calculating the distance between evolved contours. Therefore, the complexity of the extraction stage is $O(KN^2)$, where K denotes the number of scale levels used in the MCC representation. In all our experiments we used ten scale levels as using higher scale resolution does not lead to improvement in recognition efficiency. For comparison, the CSS image is usually extracted (as specified in [21], [6]) using approximately 100 iterations, each involving filtering of the original contour three times.

The matching algorithm has quite a high order of complexity, as our main intention was to find the optimal match between two descriptors and show the maximum retrieval capabilities of the proposed shape representation. The original DTW algorithm has complexity of $O(N^2)$, where N is the number of contour points (columns in the distance table). The entire matching algorithm has complexity $O(N^3)$ since N circular shifts are checked. However, the practical implementation is still quite fast due to the very simple operations which have to be performed during the matching process.

In our implementation on a regular Pentium IV/1600 MHz workstation, approximately 10 s were required to calculate MCC representations for the entire MPEG-7 database containing 1400 shape images. A single comparison, including mirrored matching, takes roughly 7 ms. As such, querying the entire MPEG-7 database with a single shape requires less than 10 seconds.

The complexity could be significantly reduced (perhaps close to linear) by using only sparse representations to roughly align the shapes before more detailed matching. One possibility could be to utilize maxima/minima at high scale levels in a voting scheme similar to that proposed in [14] in order to estimate the most likely global alignments (circular shifts). Currently we are investigating statistics calculated using the proposed representation for indexing large databases. Our preliminary results show that using such an approach, a significant number of shapes can be discarded from the search without performing any matching.

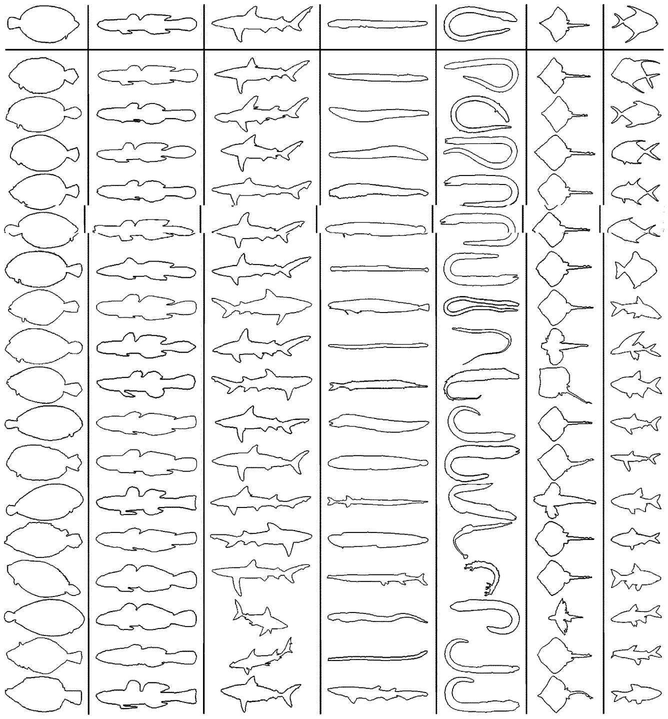

Fig. 8. Illustrative retrieval results obtained for selected entries from a database of shapes of marine creatures. The top row shows the query shape, whilst subsequent rows show the first 17 top ranked matches.

VI. Results

In this section, we present the results obtained using the MCC approach. For this purpose we simulated the main part (part *B*) of the Core Experiment "CE-Shape-1" specified in the MPEG-7 document [15]. In this experiment, a set of semantically classified images with an associated ground truth is used. The total number of images in the database is 1400 consisting of 70 classes of various shapes, each class with 20 images. Each class was used as a query, and the number of similar images (which belong to the same class) was counted

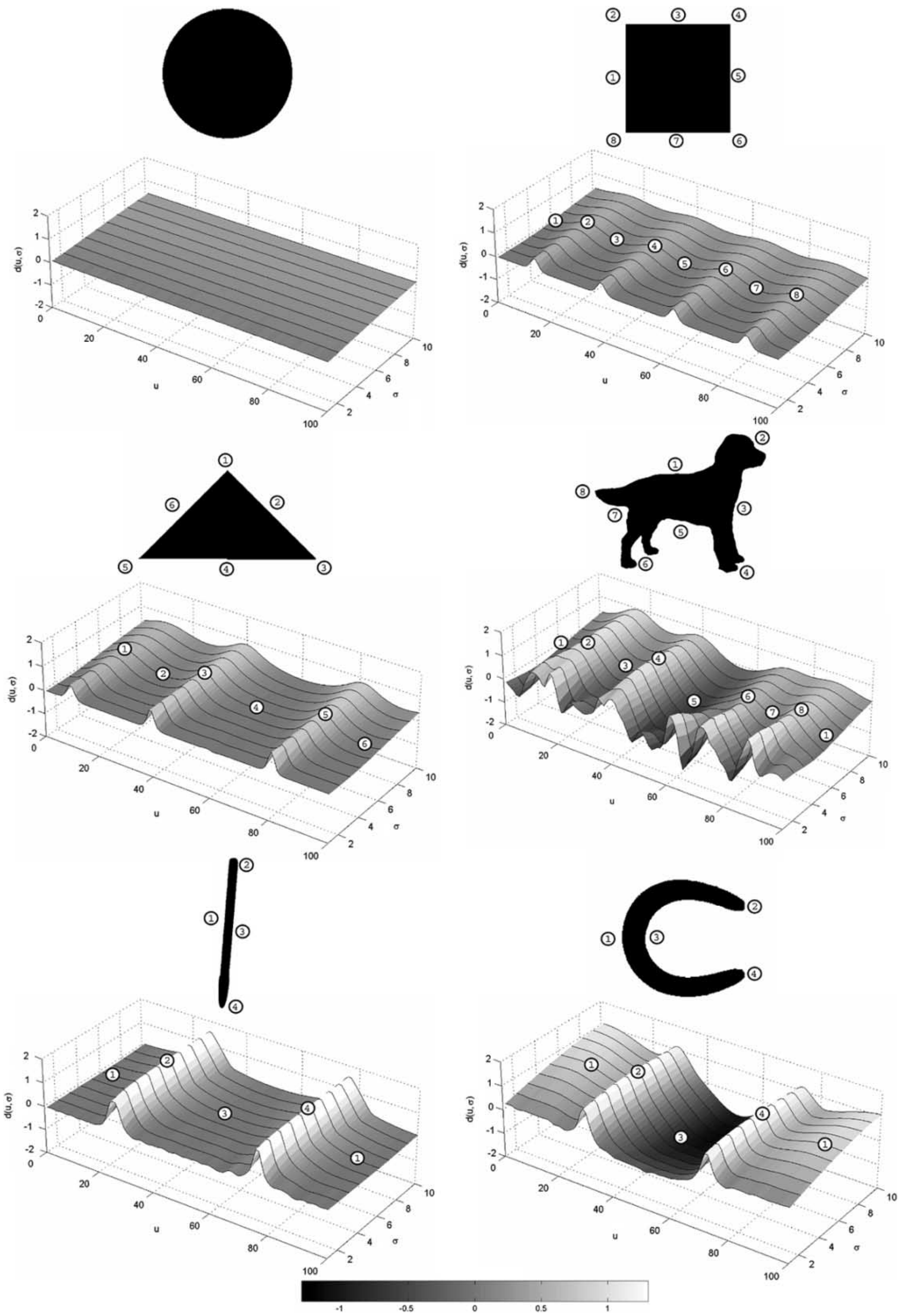

Fig. 9. Examples of MCC shape representations.

in the top 40 matches (bulls-eye test). Examples of shapes used in this experiment are shown in Fig. 6.

As outlined in [16], a 100% retrieval rate is not possible, since some classes contain objects whose shape is significantly different so that is not possible to group them into the same class using only shape-based descriptors. To illustrate this, an example is shown in [16] whereby two spoon shapes are more similar to shapes in different classes than to themselves.

The best overall performance for a method based on shape context descriptors attached to each contour was reported in [24] as 76.51%. In the Core Experiment "CE-Shape-1" performed as part of the MPEG-7 standardization process, the best performance for the method based on the best possible correspondence of visual parts [17], [18] was reported as 76.45%. In these experiments, a performance of 75.44% was obtained by a method based on CSS. Results obtained using a more recent optimized version of the CSS approach [25], [26] show a performance of 80.54%. In comparison, the total performance of our proposed MCC representation is 84.93%, which shows that our method outperforms the above approaches.

Fig. 7 shows details of retrieval results for all classes of shapes from the MPEG-7 database for both MCC and CSS approaches. In Fig. 7, the classes are sorted according to the performance of MCC, with the corresponding performance of CSS for each class overlaid. This clearly shows where the MCC approach outperforms CSS. It can be seen from Fig. 7 that for all totally or almost totally convex classes present in the MPEG-7 database, such as pencil, HCircle, device3, device4, our algorithm performs much better then CSS. The results illustrate that the proposed technique can capture important (from a human's point of view) shape features. Very good performance was obtained for classes such as *fork*, *spring*, *frog*, *pencil*, *device8*. Retrieval results obtained for classes *bat*, *cattle*, *horse*, and *lizzard* are also encouraging, taking into account the high diversity within these classes. The poor results obtained for the two classes *device6* and *device9*, can be explained in both cases by the fact that they contain shapes representing geometrical shapes (pentagons and circles respectively) with very deep cracks and holes—see Fig. 6. These shapes are similar according to region-based similarity criteria, but are very different according to contour-based similarity criteria. Clearly, for shape-based retrieval of such classes, region-based descriptors rather than contour-based descriptors should be used.

Illustrative retrieval results obtained using the MCC approach on a database of 1100 shapes of marine creatures [6], are illustrated in Fig. 8.

VII. Conclusion

In this paper, a new multi-level shape representation for single closed contours, termed Multi-scale Convexity Concavity, has been presented. Robust matching of two such descriptors using dynamic programming was developed. Based on this matching a dissimilarity measure between two contours was also proposed. The proposed scheme is very efficient and invariant to several kinds of transformations including some articulations and modest occlusions. Experimental results reveal that our approach supports search for shapes that are semantically similar for humans, even when significant intra-class variability exists. The MCC representation can be used as a starting point to obtain more compact descriptors.

Acknowledgment

The authors would like to express their gratitude to Dr. M. Bober of the Visual Information Laboratory of Mitsubishi Electric Information Technology Center Europe for generating and providing the CSS results illustrated in Fig. 7. The authors would also like to thank the Centre for Vision, Speech, and Signal Processing of the University of Surrey for making the database of shapes of marine creatures available.

References

[1] S. Lambert, E. de Leau, and L. Vuurpijl, "Using pen-based outlines for object-based annotation and image-based queries," in *Visual Information and Information Systems—Proc. 3rd Int. Conf.*, D. Huijsmans and A. Smeulders, Eds., Amsterdam, The Netherlands, June 1999, pp. 585–592.

[2] S. Loncarin, "A survey of shape analysis techniques," *Pattern Recognit.*, vol. 31, no. 5, pp. 983–1001, 1998.

[3] R. Veltkamp and M. Hagedoorn, "State of the art in shape matching", Utrecht, The Netherlands, Tech. Rep. UU-CS-1999-27, 1999.

[4] E. G. M. Petrakis, A. Diplaros, and E. Millos, "Matching and retrieval of distorted and occluded shapes using dynamic programing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, pp. 1501–1516, Nov. 2002.

[5] H. Kauppinen, T. Seppanen, and M. Pietikainen, "An experimental comparison of autoregressive and fourier-based descriptors in 2-D shape classification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 201–207, Feb. 1995.

[6] F. Mokhtarian, S. Abbasi, and J. Kittler, "Efficient and robust retrieval by shape content through curvature scale space," in *Proc. Int. Workshop Image Database and Multimedia Search*, Amsterdam, The Netherlands, 1996, pp. 35-42, [Online.] Available: http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html.

[7] M. Bober, "MPEG-7 visual shape descriptors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 716–719, June 2001.

[8] H. Blum, "A transformation for extracting new descriptors of shape," in *Whaten-Dunn, Editor, Models for Perception of Speech and Visual Forms*. Cambridge, MA: MIT Press, 1967, pp. 362–380.

[9] ——, "Biological shape and visual science (part 1)," *J. Theoretic. Biol.*, no. 38, pp. 205–287, 1973.

[10] T. Sebastian, P. Klein, and B. Kimia, "Recognition of shapes by editing shock graphs," in *Proc. 8th Int. Conf. Computer Vision*, Vancouver, BC, Canada, 2001, pp. 755–762.

[11] J. W. Gorman, O. R. Mitchell, and F. P. Kuhl, "Partial shape recognition using dynamic programing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 257–266, Feb. 1988.

[12] N. Ansari and E. J. Delp, "Partial shape recognition: A landmark-based approach," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 470–483, May 1990.

[13] N. Ueda and S. Suzuki, "Learning visual models from shape contours using multiscale convex/concave structure matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 337–352, Apr. 1993.

[14] Y. Gdalyahu and D. Weinshall, "Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 1312–1328, Dec. 1999.

[15] S. Jeannin and M. Bober, "Description of core experiments for MPEG-7 motion/shape," in *MPEG-7, ISO/IEC/JTC1/SC29/WG11/MPEG99/N2690*, Seoul, Korea, Mar. 1999.

[16] L. J. Latecki, R. Lakämper, and U. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2000, pp. 424–429.

[17] L. J. Latecki and R. Lakämper, "Contour-based shape similarity," in *Proc. Int. Conf. Visual Information Systems*, vol. LNCS 1617, Amsterdam, The Netherlands, June 1999, pp. 617–624.

[18] L. Latecki and R. Lakämper, "Shape similarity measure based on correspondence of visual parts," *IEEE Trans. Pattern Anal. Machine Intell.*, pp. 1185–1190, Oct. 2000.

[19] F. Mokhtarian and A. K. Mackworth, "A theory of multiscale, curvature-based shape representation for planar curves," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 789–805, Aug. 1992.

[20] F. Mokhtarian, "Silhouette-based isolated object recognition through curvature scale space," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 539–544, May 1995.

[21] F. Mokhtarian, S. Abbasi, and J. Kittler, "Robust and afficient shape indexing through curvature scale space," in *Proc. British Machine Vision Conf.*, 1996, pp. 53–62.

[22] A. Quddus, F. A. Cheikh, and M. Gabbouj, "Wavelet-based multi-level object retrieval in contour images," in *Proc. Very Low Bit Rate Video Coding (VLBV99) Workshop*, Kyoto, Japan, Oct. 1999, pp. 43–46.

[23] C. M. Myers, L. R. Rabiner, and A. E. Rosenberg, "Performance tradeoff in dynamic time warping algorithms fo isolated word recognition," *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-28, pp. 623–635, Dec. 1980.

[24] S. Belongie, J. Malik, and J. Puzicha, "Matching shapes," *Proc. 8th IEEE Int. Conf. Computer Vision*, pp. 454–463, July 2001.

[25] J. Atkinson and M. Bober, "Report on further optimization of the contour shape descriptor," in *MPEG-7, ISO/IEC/JTC1/SC29/WG11/MPEG00/M6039*, Geneva, Switzerland, May 2000.

[26] M. Bober and W. Price, "Report on results of the ce-5 on contour-based shape," in *MPEG-7, ISO/IEC/JTC1/SC29/WG11/MPEG00/M6293*, Beijing, China, July 2000.

Tomasz Adamek received the M.Sc. degree in telecommunications from Poznan University of Technology, Poznan, Poland in June 2001.

In 2000 he was an intern with the Visual Media Processing Group in the Research Institute for Networks and Communications Engineering, Dublin City University. In Oct 2001, he joined the Centre for Digital Video Processing at Dublin City University as a full-time Ph.D. student in the School of Electronic Engineering. His research interests include shape analysis and representation, region-based and object-based segmentation and tracking, and video sequence analysis for feature extraction for indexing applications.

Noel E. O'Connor (M'02) received the primary degree and Ph.D. degree from Dublin City University (DCU), Dublin, Ireland, in 1992 and 1998, respectively.

From September 1992 to July 1999, he was a Research Assistant with the Video Coding Group of Teltec Ireland—DCU. During this time, he was responsible for Teltec's involvement in a number of EU-funded projects and acted as the Irish national representative to the ISO MPEG. He is currently a Lecturer with the School of Electronic Engineering, DCU, where he is Programme Chair of the B.Eng. degree in digital media engineering degree programme. He is a Principal Investigator in the Centre for Digital Video Processing, an inter-disciplinary University Designated Research Centre in Dublin City University. He is a member of the Steering Committee of the Research Institute for Networks and Communications Engineering and the Science Foundation Ireland Adaptive Information Cluster. His current research interests include audio–visual analysis for knowledge extraction from digital content, region-based and object-based segmentation, scene-level analysis, automatic summarization, and power-efficient hardware architectures for video processing.