Efficient Contour-based Shape Representation and Matching

Tomasz Adamek
Centre for Digital Video Processing
Dublin City University
Dublin, Ireland
adamekt@eeng.dcu.ie

Noel O'Connor
Centre for Digital Video Processing
Dublin City University
Dublin, Ireland
oconnorn@eeng.dcu.ie

ABSTRACT

This paper presents an efficient method for calculating the similarity between 2D closed shape contours. The proposed algorithm is invariant to translation, scale change and rotation. It can be used for database retrieval or for detecting regions with a particular shape in video sequences. The proposed algorithm is suitable for real-time applications. In the first stage of the algorithm, an ordered sequence of contour points approximating the shapes is extracted from the input binary images. The contours are translation and scale-size normalized, and small sets of the most likely starting points for both shapes are extracted. In the second stage, the starting points from both shapes are assigned into pairs and rotation alignment is performed. The dissimilarity measure is based on the geometrical distances between corresponding contour points. A fast sub-optimal method for solving the correspondence problem between contour points from two shapes is proposed. The dissimilarity measure is calculated for each pair of starting points. The lowest dissimilarity is taken as the final dissimilarity measure between two shapes. Three different experiments are carried out using the proposed approach: letter recognition using a web camera, our own simulation of Part B of the MPEG-7 core experiment "CE-Shape1" and detection of characters in cartoon video sequences. Results indicate that the proposed dissimilarity measure is aligned with human intuition.

Categories and Subject Descriptors

I.4.7 [**Image Processing and Computer Vision**]: Feature Measurement—*feature representation, size and shape.*

General Terms

Algorithms, Measurement, Performance.

Keywords

Shape representation and matching.

MIR'03, November 7, 2003, Berkeley, California, USA.

1. INTRODUCTION

Shape analysis methods play an important role in systems for object recognition, matching, registration, and analysis. A user survey presented in [6] regarding cognition aspects of image retrieval shows that users are more interested in retrieval by shape than by color and texture. Many shape analysis techniques have been proposed over the past three decades [9, 15]. However, retrieval by shape is still considered to be one of the most difficult aspects of content-based search. Common criteria used for shape representation for reliable shape matching and retrieval include: uniqueness, robustness to distortion and noise, invariance to translation, scale, rotation and symmetric transformations, scalability and efficiency [12]. The choice of the most suitable shape analysis method is often a compromise between retrieval efficiency and computational complexity. For example, the speed requirements of real-time applications often constrain the type of techniques that can be used.

Well known methods for compact shape representation and matching of 2D contours include Fourier Descriptors (FD) [5], Curvature Scale Space (CSS) [10] and approaches based on the classical Hausdorff distance [1]. The major advantage of FD methods are that they are easy to implement and are based on the well developed theory of Fourier analysis. The disadvantage is that after the Fourier transform, local shape information is distributed to all coefficients and not localized in the frequency domain [9]. The CSS descriptor is more robust to shape deformation than FD and because of this and other advantages like fast matching and compact representation, has been incorporated in the ISO/IEC MPEG-7 content description standard. Unfortunately, the retrieval accuracy of the CSS method can be poor for curves which have a small number of concavities or convexities. In particular, this representation cannot distinguish between various convex curves [8]. Another disadvantage of the compact descriptors is that their extraction is usually computationally expensive. Although this is not a problem for creating databases (since feature extraction is performed off-line), this makes it difficult (or even impossible) to use them for fast on-line matching of two shapes provided as binary masks. A difficulty with the traditional Hausdorff distance is that it is very sensitive to noise and as such a number of related approaches have been developed that address this [15]. Our approach can be considered to be an alternative to these approaches.

Another class of contour matching algorithms includes methods which attempt to solve the correspondence prob-

lem between two shapes and measure the similarity based on this assignment. In [2] the shape context is assigned to each contour point in order to solve the correspondence problem. In [12] the optimal match between contour segments is found using *Dynamic Programming*. The main advantage of the above methods is robustness against occlusions. The main drawback is their high computational complexity.

In this paper, we present a fast sub-optimal method for solving the correspondence problem between vertices of closed contours. The proposed similarity/dissimilarity measure is based only on the geometrical distances between corresponding contour points and attempts to quantify shape similarity in a way aligned with human intuition. The retrieval rate of the algorithm is comparable to methods using compact descriptors (FD, CSS) but is much less computationally expensive compared to methods which attempt to find the optimal match between shapes. Our assumptions are verified in extensive tests, including our own simulations of MPEG-7 core experiments.

The remainder of this paper is organized as follows: In the second section the proposed shape matching algorithm is described in detail. The complexity of the algorithm is discussed in section 3. In section 4, experimental results are presented and discussed. Finally, conclusions are formulated in section 5.

2. ALGORITHM STRUCTURE

The structure of the proposed algorithm is shown in Figure 1. It operates in two main stages: *feature extraction* and *feature matching*. In the first stage, the boundaries of the shapes from the input binary images are traced. The extracted contours are translation and scale-size normalized, and a small set of the most likely starting points for both shapes are estimated. In the second stage, the dissimilarity measure is calculated using features extracted in the first stage. The starting points from both shapes are assigned into pairs. Before evaluating the dissimilarity measure for a given pair of starting points, the rotation between these points is estimated and one of the shapes is rotated. The dissimilarity between two shapes is calculated for each pair of starting points. The lowest overall dissimilarity is taken as the dissimilarity measure between the two shapes.

In the remainder of this section a detailed description of the most important stages of the algorithm is presented.

2.1 Contour tracing and down-sampling

In this first step, an ordered sequence of contour points approximating the input shape is extracted. For this purpose, a contour tracing algorithm (also known as border following or boundary following) similar to that of Pavlidis [11] is applied to the input binary image. The algorithm ignores any "holes" present in the shape providing an ordered sequence of the contour pixels. The tracing algorithm can be easily and efficiently implemented using look-up tables for selecting the pixels in the current pixel neighborhood for checking and deciding in which order they should be checked.

In the next step, a vector of equally distributed points along the curve is extracted from the ordered sequence of contour pixels. This stage is important because the final dissimilarity measure is based on distances between corresponding contour points of the shapes being compared. As such, an approximately equal number of contour points for both shapes is required (see Figure 1). There are many

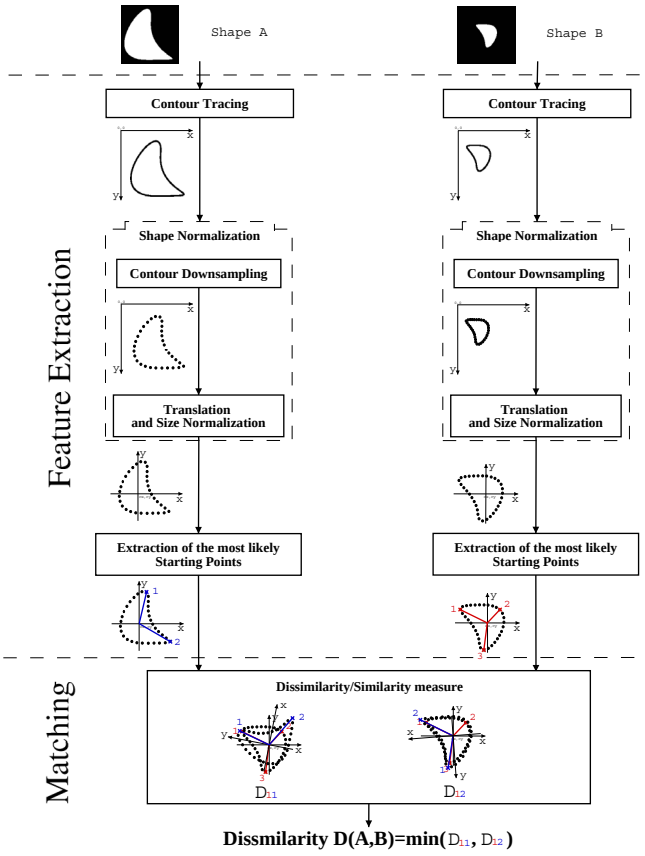

Figure 1: Main steps of the shape analysis algorithm.

approaches for approximating contour curves using a small number of vertices or segments. In our approach, a very simple and efficient downsampling scheme is utilized. It extracts contour points in such a way that they will be approximately equally distributed along the contour. This can be easily performed during one scan of the contour pixel sequence. From this point on, the contour of the shape is represented by the ordered sequence $\Big(x(s), y(s)\Big)$ of contour points (vertices), where s denotes the position along the contour.

2.2 Translation and Scale Normalization

Clearly, moving a region from one location in the image to another and/or changing the size of it should not affect shape analysis. Since the centroid of the shape is invariant to translation, rotation and scaling, we utilize its location to solve the translation problem. In the presented approach, the object curve is translated to place the centroid at coordinates $(0, 0)$. Scaling invariance is obtained by normalizing the shape curve by shape area (see for example [14]). This translation and scaling invariance can be achieved using the regular moments. In the presented scheme all low order moments needed for shape normalization are derived from the boundary contour points. This approach is much faster in comparison to calculating moments from definition equations. A similar method to the one shown in [3] for deriving the area (m_{00}) of the shape from a simple chain code is used. Here this method is extended for moments m_{01} and m_{10}.

The definition for a regular moment m_{pq} is:

$$m_{pq} = \iint x^p y^q f(x,y)\, dx\, dy \tag{1}$$

As illustrated in Figure 2, the algorithm moves from one contour point to another in a clockwise direction around the contour. The moments are calculated as the sum of appropriate moments of trapezoids created from the current and previous contour point and the x axis. An example of such a trapezoid is shown in Figure 2(b). The vertices of the trapezoid are $\Big(x(c), y(c)\Big)$, $\Big(x(p), y(p)\Big)$, $\Big(x(p), 0\Big)$ and $\Big(x(c), 0\Big)$ where c and p denote current and previous contour point respectively. From equation 1 the formulas for the low

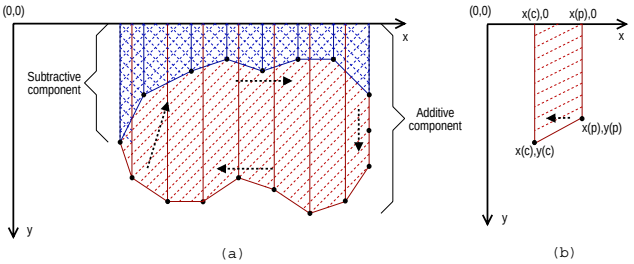

Figure 2: Calculating the moments of the shape.

order moments of trapezoid t as functions of its vertices can be easily derived:

$$m^t_{00} = \frac{1}{2}[x(p) - x(c)][y(c) + y(p)] \tag{2}$$

$$m^t_{10} = \frac{1}{4}[x(p) - x(c)][x(c) + x(p)][y(c) + y(p)] \tag{3}$$

$$m^t_{01} = \frac{1}{4}[y(c) - y(p)][x(c) + x(p)][y(c) + y(p)] \tag{4}$$

Note that m^t_{00} and m^t_{10} have negative values for vertices from the upper border of the shape (subtractive component) and positive values for vertices from the lower border (additive component). Similarly m^t_{01} has negative values for the left part of the shape and positive values for the right part (not shown in the figure).

Low order moments of the entire shape can be calculated as the sum of moments of all trapezoids as follows:

$$m_{pq} = \sum_{\forall t \in V} m^t_{pq}, \tag{5}$$

where V denotes the set of all vertices of the shape.

Finally, the translation and scale normalization of the contour is obtained by changing the coordinates of the contour points according to the following transformation:

$$x'(s) = \frac{x(s) - x_c}{\alpha}, \qquad y'(s) = \frac{y(s) - y_c}{\alpha} \tag{6}$$

where $[x_c, y_c]$ denote coordinates of the centroid calculated as:

$$x_c = \frac{m_{10}}{m_{00}}, y_c = \frac{m_{01}}{m_{00}} \tag{7}$$

and α denotes a scaling factor defined as:

$$\alpha = \sqrt{\frac{m_{00}}{AREA}}, \tag{8}$$

where $AREA$ is a chosen constant - the same for all shapes compared.

For the remainder of this paper we use $\Big(x(s), y(s)\Big)$ to represent the translation and scale normalized contour.

2.3 Rotation Alignment

Rotating the object region or changing the point from which the tracing of the boundary of the region starts should not affect shape analysis. In order to avoid evaluation of the dissimilarity measure for every possible pair of starting contour points and still maintain robustness to shape deformations we propose to extract a small set of the most likely starting points for each shape. The similarity/dissimilarity measure is evaluated only for pairs of chosen starting points. The accuracy and speed of the search for the optimal rotational match depends on the number of chosen starting points. The criteria used to extract starting points and the procedure used for choosing pairs of starting points for dissimilarity evaluation can be adjusted to a given application.

In this paper we propose the following general scheme for extracting sets of starting points. Initially, the contour points that are a greater distance from the centroid than a threshold distance T_d are marked as potential starting points. T_d can be expressed as a fraction of the maximum distance d_{max}:

$$T_d = p_d \cdot d_{max}, \tag{9}$$

where parameter p_d is a real number in the range $[0, 1]$. A single marked contour point is selected from each continuous group of marked contour points. To avoid the situation where too many contour points will be represented by just one starting point, a maximum number of marked points T_n which can be represented by one starting point is introduced:

$$T_n = \lceil p_n \cdot N \rceil, \tag{10}$$

where N denotes the number of all contour points representing the shape and p_n denotes a number within the range $[0, 1]$. If the group has more points than T_n, it is divided into smaller groups of cardinality less than or equal to T_n. Finally, from each group the contour point most distant from the centroid is chosen as the starting point.

The intensity of the search for the optimal rotation alignment between two shapes can be varied by parameters p_d and p_n. The extracted starting points are used to narrow down the search space for the best rotation alignment. For each rotation the dissimilarity measure is calculated. The lowest dissimilarity measure obtained during this search is taken as the final dissimilarity between two shapes. A number of different search strategies can be used depending on application. A general searching algorithm is described in the following.

The reference starting point, sp_{ref}, is taken as the most distant starting point of the shape with the largest number of starting points, or the largest initial area. For each starting point on the second shape, sp_i, the shape is rotated to align its starting point with sp_{ref}, and the dissimilarity measure is calculated. The lowest dissimilarity value found during the search is taken as the final dissimilarity measure between the shapes.

2.4 Dissimilarity Measure

The presented dissimilarity metric is based on the distances between the contour points of two shapes. Before

calculating the dissimilarity, the correspondence problem between the contour points from both shapes must first be solved. In the presented method a semi-optimal solution to the contour point correspondence problem is proposed. It is robust to deformation and noise and yet the complexity is $O(N)$.

The approach is based on two control points cp_A and cp_B moving along the contours A and B, beginning from the aligned starting points. They move along the contours in N' steps and for each step the distance between them is obtained. Each step consists of two stages. During the first stage, each control point moves one position forward in the contour point sequence. For the new position, the Euclidean distance $|d_c(i)|$ between the coordinates of these control points is calculated. During the second stage, the distances between the control points and the next contour points in each contour ($|d_{AB}(i)|$ and $|d_{BA}(i)|$) are calculated. If one of these is smaller than $|d_c(i)|$ the appropriate control point moves one position forward to minimize the distance between cp_A and cp_B. The final distance $|d(i)|$ for step i is obtained as the minimum of the distances $|d_c(i)|$,$|d_{AB}(i)|$ and $|d_{BA}(i)|$. In this way, control

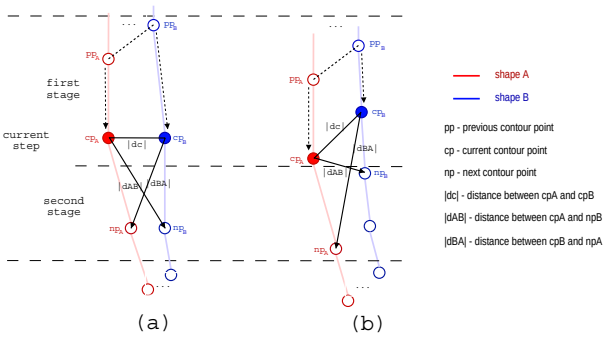

Figure 3: An illustration of the proposed contour point assignment.

points move together along the contours as closely as possible. Usually they move only one position forward during one step, but one can move two positions if it produces a smaller distance. In this way, some contour points are skipped - i.e. not assigned to any of the contour points from the other contour. An example of a step while skipping one contour point is shown in Figure 3(b).

The distances between the contour points assigned as outlined above are used to calculate the dissimilarity between two shapes. The dissimilarity measure should quantify the dissimilarity between shapes in ways aligned with human intuition. We assume that if two shapes are similar then there should be strong correspondence between contour points and the distances between corresponding contour points should be small. The proposed dissimilarity metric is based on both the geometrical distances between assigned contour points and their variations along the contours. The dissimilarity measure is defined as:

$$D(A,B) = \frac{c}{cir_{min}^2} \cdot |\overline{D}| \cdot |V|, \qquad (11)$$

where:

- $|\overline{D}|$ denotes average distance between assigned contour points,
- $|V|$ denotes the standard deviation of the distances between assigned contour points,
- c denotes a chosen constant which rescales values of the dissimilarity measure to a convenient range of values (calibration of the algorithm),
- cir_{min} is the minimum circumference of the circumferences of both normalized shapes.

The function of cir_{min} is to adjust the dissimilarity metric to human perception of dissimilarity depending on the complexity of the shapes being compared. It has been observed that humans perceive even small differences between two simple shapes as significant, whereas small differences between two more complicated shapes are not so important. Assuming, that the shape is size normalized the circumference can be used as a measure of the complexity of the shape.

3. COMPLEXITY

In this section we discuss the complexity of the overall algorithm. The complexity of all feature extraction stages are $O(N)$, where N denotes the initial number of contour points. The complexity of the matching stage depends on the number of starting point pairs for which the dissimilarity measure has to be evaluated. The calculation of the dissimilarity measure for one pair of chosen starting points requires $O(N')$ operations, where N' denotes the number of contour points used to approximate the contours. In the worst case scenario, when all possible contour points are chosen as potential starting points, the total complexity of the matching stage is $O(N'^2)$. When only a small set of starting points is used for each shape the matching stage complexity converges to $\sim O(N')$. For all experiments described in this paper the values $p_d = 0.7$ and $p_n = 0.05$ where chosen which lead to approximately 2 starting points for elongated shapes and 20 for circular shapes.

The proposed algorithm was implemented in the C++ programming language and run on a standard PC with a Pentium III 600 MHz processor. The algorithm was able to perform approx 1000 shape matchings per second (where the contours were approximated with 100 points i.e. $N = 100$).

4. EXPERIMENTAL RESULTS

The evaluation of the performance of a shape matching algorithm is a difficult task, because shape similarity is a subjective matter. Human judgments of shape similarity differ noticeably [10]. Consistent evaluation criteria do not exist for shape description and matching methods [9]. Because of the above difficulties, we present the results from three different experiments: letter recognition using a web camera, our own implementation of Part B of the MPEG-7 Core Experiment "CE-Shape-1" [4], and detection of characters in cartoon video sequences. For the last two experiments the dissimilarity is evaluated for input contours and their mirrored equivalent.

4.1 Letter recognition using a web camera

Discriminatory ability and robustness to rigid transformations of the proposed algorithm was tested by performing a similar experiment to one used in [13]. Several letters of the alphabet {m, n, u, h, l, t, f} were used as a test set. Images

were created by printing out the letters on a laser printer and digitizing the page using a simple web camera. Large and small letters were printed using 240pt and 128pt respectively. Rotated letters were created by rotation of the large letters by an angle of 165 degrees. All letters were Arial font.

(a)

original letters

original letters	m	n	u	h	l	t	f
m	0,00	1,74	2,81	1,18	5,42	2,58	2,72
n	1,74	0,00	0,01	0,33	7,27	3,49	3,25
u	2,81	0,01	0,00	0,29	8,45	4,04	5,14
h	1,18	0,33	0,29	0,00	9,18	4,16	3,70
l	5,42	7,27	8,45	9,18	0,00	0,24	0,36
t	2,58	3,49	4,04	4,16	0,24	0,00	0,32
f	2,72	3,25	5,14	3,70	0,36	0,32	0,00

(b)

original leters

resized letters	m	n	u	h	l	t	f
m	0,00	1,76	2,85	1,11	5,34	2,79	1,81
n	1,87	0,01	0,01	0,36	7,03	3,48	3,37
u	2,37	0,01	0,02	0,27	9,33	4,34	5,38
h	1,44	0,39	0,36	0,01	4,41	3,32	3,92
l	5,67	7,23	8,23	7,76	0,01	0,24	0,42
t	2,44	3,32	3,81	4,03	0,29	0,01	0,30
f	2,15	3,70	3,59	2,98	0,42	0,44	0,02

(c)

original leters

rotated letters	m	n	u	h	l	t	f
m	0,01	1,44	2,92	1,08	7,85	3,00	3,92
n	1,66	0,01	0,01	0,32	6,60	3,59	3,26
u	1,44	0,01	0,01	0,36	7,64	3,67	3,50
h	1,42	0,39	0,30	0,01	9,93	4,52	3,76
l	5,19	6,90	8,18	8,18	0,01	0,20	0,32
t	2,59	3,49	3,97	4,64	0,33	0,03	0,28
f	1,83	3,67	4,96	3,89	0,56	0,39	0,08

(d)

Figure 4: Letter recognition using web camera ($c = 1500$): (a)-test set, (b)-dissimilarities between original letters, (c)-dissimilarities between original and resized letters, (d)-dissimilarities between original and rotated letters.

The Tables in Figure 4 show distances between the original shapes and the rotated and scaled set. As expected, "n" and "u" match quite closely, since they are only rotated versions of each other. "h" matches "n" and "u" better than the other letters. Also "l", "t" and "f" match better than other letters since all are very elongated. We can see that discretization introduces some noise and shape deformations, thus the distances between the same letters after rotation and resizing are not exactly zero. However, distances between different letters are always much larger (10 to 100 times) than those between the same letters.

4.2 MPEG7 Core Experiment CE-Shape-1

In this section, results obtained by the proposed method for Part B of the Core Experiment "CE-Shape-1" as specified in the MPEG-7 document [4] are presented. In this experiment, a set of semantically classified images with a ground truth is used. The total number of images in the database is 1400: 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 in the top 40 matches (bulls-eye test). As concluded in [8], 100% retrieval 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 information.

A retrieval rate of 76.51% was reported in [2] for a method based on shape context descriptors attached to each contour points. A performance of 76.45% was reported in [8] for a method based on the best possible correspondence of visual parts [7]. In comparison, the total performance of the proposed shape analysis technique for our experiments is 77.76%. Figure 5 shows the detailed performance for all classes of shapes from the MPEG-7 database.

(a) (b) (c)

Figure 6: Manually chosen set of templates used for character detection: (a)-basic set of templates for "Bart", (b)-extension of template set for "Bart", (c)-templates for "Homer"

4.3 Detection of characters in cartoons

In the final experiment, we demonstrate recognizing shapes of regions extracted automatically from video sequences. Because automatic object segmentation in generic video content is very difficult we decided to use simple cartoon video sequences for which the segmentation process is relatively straightforward. In the test, three different sequences of "The Simpsons" cartoon were used. In this cartoon all characters have yellow skin and very distinctively shaped heads, so the shape of the head was a natural choice for character recognition.

For each tested frame "yellow color segmentation" was performed. Morphological dilation was used to fill small holes present in the extracted regions. Every extracted region was matched with a manually chosen set of templates. The region was classified as the head of the searched character if the dissimilarity between the region and one of the template shapes was below a chosen threshold.

Three manually marked up video sequences, each around 20 minutes long, were used for the tests. Three different tests were performed: two for "Bart Simpson" and one for "Homer Simpson" detection. The set of template shapes used for these tests are shown in Figure 6. Every tenth frame was searched. The results obtained were compared with human judgment, and are presented in terms of precision and recall in Table 1.

General observation shows that even in the case of simple cartoons, the majority of recognition mistakes was caused by improper segmentation. The lower precision for "Homer" detection can be explained by the higher variety of head shapes for this character. To solve this problem, a more objective method for determining the most representative shapes for each character should be utilized. The test was performed in real time on a standard PC with Pentium III 600 MHz processor.

5. CONCLUSIONS

In this paper an efficient approach for similarity calculation between closed contours is proposed. The approach can deal with noisy and distorted shapes and is invariant to translation, scale change and rotation. The proposed dissimilarity measure is based on the geometrical distances between corresponding contour points.

Extensive performance experiments on several data sets were carried out. The experiments confirmed that the proposed method is well suited to shape matching and retrieval of shapes with moderate amounts of noise and deformations. The results indicate that our dissimilarity measure closely conforms to the common perception of dissimilarity by humans. Because of its low complexity the proposed algorithm

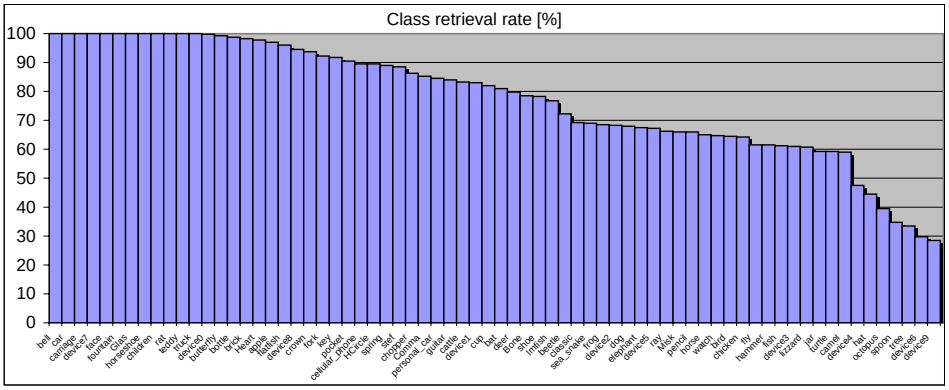

Figure 5: Retrieval rate for all classes in MPEG7 core experiment (CE-Shape1) part B (sorted by retrieval rate).

Experiment:	Bart, template (a)				Bart, templ. (a) and (b)				Homer, template (c)			
Sequence:	1	2	3	avg	1	2	3	avg	1	2	3	avg
Precision [%]	73	78	75	75	85	89	83	86	64	51	50	55
Recall [%]	97	94	98	96	84	67	83	78	96	97	96	96

Table 1: Character recognition results ($threshold = 0.15$, $c = 1500$).

is suitable for real-time applications.

6. ACKNOWLEDGMENT

The support of the Informatics Research Initiative of Enterprise Ireland is gratefully acknowledged. This material is based upon work supported by the IST programme of the EU in the project IST-2000-32795 SCHEMA.

7. REFERENCES

[1] H. Alt, B. Behrends, and J. Blomer. Approximate matching of polygonal shapes. *Annals of Mathematics and Artificial Intelligence*, pages 251–265, 1995.

[2] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *Eighth IEEE International Conference on Computer Vision*, July 2001.

[3] H. Freeman. Computer processing of line-drawing images. *Computing Surveys*, 6(1), March 1974.

[4] S. Jeannin and M. Bober. Description of core experiments for mpeg-7 motion/shape. *MPEG-7, ISO/IEC/JTC1/SC29/WG11/MPEG99/N2690, Seoul*, March 1999.

[5] H. Kauppinen, T. Seppanen, and M. Pietikainen. An experimental comparison of autoregressive and fourier-based descriptors in 2d shape classification. *IEEE Transactions on Pattern Analysis and Machine Inteligence*, 17:201–207, 1995.

[6] S. Lambert, E. de Leau, and L. Vuurpijl. Using pen-based outlines for object-based annotation and image-based queries. In D. Huijsmans and A. Smeulders, editors, *Visual Information and Information Systems - Proceedings of the Third International Conference VISUAL'99, Amsterdam, The Netherlands*, LNCS 1614, pages 585–592. Springer, June 1999.

[7] L. Latecki and R. Lakämper. Shape similarity measure based on correspondence of visual parts. *IEEE Trans. Pattern Analysis and Machine Inteligence*, pages 1185–1190, October 2000.

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

[9] S. Loncarin. A survey of shape analysis techniques. *Pattern Recognition*, 31(5):983–1001, 1998.

[10] F. Mokhtarian, S. Abbasi, and J. Kittler. Robust and efficient shape indexing through curvature scale space. In *British Machine Vision Conference*, 1996.

[11] T. Pavlidis. *Algorithms for Graphics and Image Processing*. Computer Science Press, Rockville, Maryland, 1982.

[12] E. G. M. Petrakis, A. Diplaros, and E. Millos. Matching and retrieval of distorted and occluded shapes using dynamic programing. *IEEE Transactions on Pattern Analysis and Machine Inteligence*, 24:1501–1516, 2002.

[13] Y. Rui, A. C.She, and T. S. Huang. Modified fourier descriptors for shape representation - a practical approach. In *First International Workshop on Image Databases and Multi Media Search*, 1996.

[14] D. Shen and H. Horace. Discriminative wavelet shape descriptors for recognition of 2-d patterns. *Pattern Recognition*, 32(2):151–156, 1999.

[15] R. C. Veltkamp. Shape matching: Similarity measures and algorithms. Technical report UU-CS-2001-03, Utrecht University, 2001.