Efficient Curvature-based Shape Representation for Similarity Retrieval

Farzin Mokhtarian, Sadegh Abbasi and Josef Kittler

Centre for Vision Speech and Signal Processing
Department of Electronic and Electrical Engineering,
University of Surrey England
Guildford, GU2 5XH
Email F.Mokhtarian@ee.surrey.ac.uk

ABSTRACT

The Curvature Scale Space (CSS) image is a multi-scale organisation of the inflection points of a closed planar curve as it is smoothed. It consists of several arch shape contours, each related to a concavity or a convexity of the curve. In our recent work, we have used the maxima of these contours to represent the boundary of objects in shape similarity retrieval. In our new approach, each segment of a shape is represented by the relevant maximum of the CSS image as well as the average curvature on the segment at certain level of scale. In this paper we explain how this new representation together with its matching algorithm improve the performance of our shape similarity retrieval system.

To evaluate the proposed method, we created a small classified image database. We then measured the performance of the system on this database. The quantified results of this test provided supporting evidence for the performance superiority of the proposed method.

1 Introduction

The objective of researchers in *content based image database* area is to allow users to search through databases using sketches, texture, color, shapes and other iconic and graphical information. The users usually specify an image, or a part of it, and want the system to find all images in the database *like* that (part of the) image.

A number of methods have been proposed and implemented for shape similarity retrieval. Some of them are based on conventional shape representation methods like Fourier descriptors [11] and polygonal approximation [4]. In other methods, shapes have been represented by a number of global parameters like eccentricity, circularity, number of holes [3], major axis orientation and a set of algebraic moment invariants [10], and efforts have been focused on an efficient and rapid indexing.

In our recent work, we have used the maxima of the Curvature Scale Space (CSS) image [5][9] as shape features for similarity retrieval [1][6][7]. We have also addressed the problem of shallow concavities in this representation [2] [8]. Shallow and deep concavities may both create large contours in the CSS image. As a result, a shape with curved segments may appear as the output of the system in response to a query containing a shape with shallow concavities. A rather complicated remedy was proposed in [2]. Here we introduce a simple and effective way to overcome this problem. We also discuss the shortcomings of the new method and compare it with the previous one.

The following is the organisation of this paper. In section 2 the CSS image and the problem of shallow concavities is reviewed. It is followed by introducing the new representation in section 3. The new method employs the average curvature on each segment of the shape as well as the maxima of the CSS image. Results and discussions are presented in section 5. It is followed by conclusion remarks in section 6.

2 CSS image and the problem of shallow concavities

Let Γ be a closed planar curve, and let u be the normalised arc length parameter on Γ:

$$\Gamma = \left\{ \big(x(u), y(u)\big) \;\middle|\; u \in [0,1] \right\}$$

If each coordinate function of Γ is convolved with a 1-D Gaussian kernel of width σ , the resulting curve, Γ_σ, will be smoother than Γ . As σ increases, Γ_σ becomes smoother and the number of curvature zero crossings on it decreases. When σ becomes sufficiently high, Γ_σ will be a convex curve with no curvature zero crossings (see Figure 1). The process can be terminated at this stage and the resulting points can be mapped to the (u, σ) plane. The result of this process will be a binary image called Curvature Scale Space image of the curve (see Figure 2). The horizontal axis in this image represents the normalised arc length u. To normalise the arc length, we re-sample the boundary and represent it by 200 equally distant points. Every point on the boundary has a correspondence on the horizontal axis of the CSS image. The vertical axis represents σ, the width of the Gaussian kernel. The intersection of every horizontal line with the contours in this image indicates the locations of curvature zero crossings on the corresponding evolved curve Γ_σ.

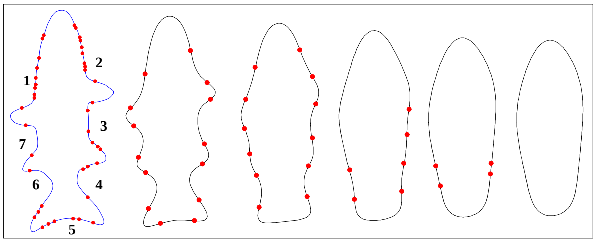

Figure 1: Shrinkage and smoothing of the curve and decreasing of the number of curvature zero crossings during the evolution, from left: $\sigma = 1, 4, 7, 10, 12, 14$.

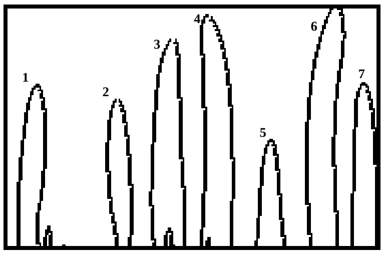

Figure 2: The CSS image of the shape of Figure 1.

The number of columns of the CSS image is equal to the number of equidistant samples on the related curve. In fact, each column relates to a point on the resampled curve. Since the number of these samples is usually less than the number of original points on the 8-connected representation of the curve, re-sampling removes some details of the curve.

The number of rows of the CSS image reflects the number of iterations needed to convert the curve to a convex curve. Two factors are involved in this value, first the shape of the original curve and second, the difference between the filter width in two successive iterations, namely $\Delta\sigma$.

If a curve contains deep and wide concavities, a filter with larger value of σ is needed to convert it to a convex curve. Therefore, the process of evolution will not end rapidly as the value of σ increases.

Each row of the CSS image corresponds to a certain level of smoothing. This is related to a value of sigma which is higher by one $\Delta\sigma$ than the previous row and lower by one $\Delta\sigma$ than the next row. The value of $\Delta\sigma$ is chosen as 0.1 in our experiments. Note that the process of smoothing starts with $\sigma = 1$. The resulting curvature zero crossings for $\sigma = 1$ is shown in bottom row of the CSS image.

The actual CSS image which is used in experiments is a digital binary image. The maxima of the contours of this image are extracted directly from the actual CSS image. x-coordinate of a maximum indicates the location where a pair of curvature zero crossings merge.

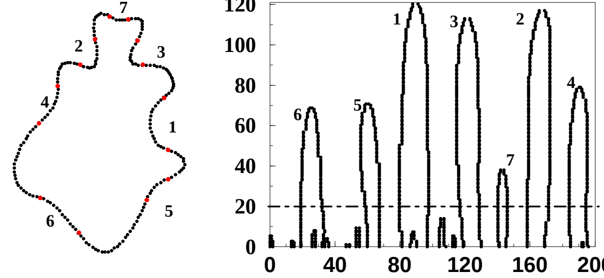

Figure 3: The appropriate level of smoothing to calculate the average curvature.

y-coordinate of a maximum indicates the number of iterations needed until this merge actually happens.

We normalise the x and y-coordinates of the CSS image by considering the length of x-axis as one. To normalise the x and y-coordinates of a maximum, it should then be divided by the number of samples.

The values of 200 and 0.1 for the number of samples and $\Delta\sigma$ are chosen so that the aspect ratio of the CSS image stays roughly around unity. This results in almost the same weight for x and y coordinates of the contour maxima of the CSS image.

Noise or real maximum Small contours of the CSS image correspond to noise or small ripples of the curve. They are not included in the representation to prevent a complicated and inefficient matching algorithm. In most cases, even if small contours are considered, they do not play a major role in the final matching value between an input shape and a model from the database. In our system, if a maximum height is less than $\frac{1}{6}$ of the number of rows of the CSS image, it is considered as noise. As a result, major concavities and convexities of a shape will contribute in the representation.

3 The new representation

In our previous approach [7][6] we represented an object boundary by a set of two dimensional points, the locations of maxima of its CSS image contours. Each CSS contour, and hence its maximum correspond to a segment of the input contour with two curvature zero crossings at the end-points. In our new approach, the average curvature between these zero crossings is calculated and after normalisation, is considered as the third dimension of CSS maxima. Therefore, an object boundary will be represented by a set of three dimensional points. Each point consists of x and y coordinates of a maximum of the CSS image, and the normalised average curvature of the corresponding segment at a certain level of scale as the z coordinate. This level is equivalent to $\frac{1}{6}$ of the height of the CSS image.

An example is given in Figure 3. The number of rows of the CSS image is 120. The level of smoothing for curvature calculation is then calculated as $\frac{120}{6} = 20$. Since the first row of the CSS image corresponds to $\sigma = 1$ and each row reflects a 0.1 increase in σ, the equivalent level of smoothing is calculated as $\sigma = 1+0.1*20 = 3$. In left side of Figure 3 the smooth curve is presented. After extracting the maxima of the CSS image and calculating the average curvature for each segment, we determine the shape representation as follows.

$$\{(0.44\ \ 0.6\ \ 0.034), (0.83\ \ 0.56\ \ 0.073), (0.83\ \ 0.58\ \ 0.073),$$

$$(0.6\ \ 0.56\ \ 0.069), (0.85\ \ 0.4\ \ 0.019), (0.3\ \ 0.35\ \ 0.025),$$

$$(0.3\ \ 0.35\ \ 0.025)\}$$

The first two dimensions are x and y coordinates of each maxima(which are divided by the number of columns, eg 200) and the third one is the average curvature at $\sigma = 3$ level. As we can see from this example, the average curvature is normally less than the other two dimensions. Moreover, it is not scale invariant, and therefore it must be normalised.

Normalisation of the average curvature Normalisation of curvature is applied in two stages. Since curvature is not scale invariant, at the first stage, we re-scale all shapes of the database so that the perimeter of all shapes be equal. This will ensure that the comparison between different shapes based on their curvature is meaningful. At the second stage, we should re-scale the curvature so that the absolute value of the third dimension of each point falls in a suitable range in comparison with the other dimensions. Since we use the Euclidean distance later in the matching algorithm, if the absolute value of the average curvature is too low, then its contribution in the final matching value would be too small and vice versa.

The scaling factor for the average curvature is proportional to the importance of it in comparison with the location of the maximum. The x coordinate of a maximum will be in the range of $[0, 1]$ with a resolution of 0.05, these figures are effectively $[0.04, 1.5]$ and 0.05 for the y-coordinate.

After performing a sequence of experiments, we concluded that by leaving the scale-factor of the average curvature as 1 and limiting the upper band of average curvature at 0.1, the best results can be achieved. We also studied the distribution of average curvature for all segments of all objects of our database to achieve this result.

4 Matching algorithm

In this section we explain the basic concepts of our matching algorithm which compares two sets of maxima, one from the input *image* and the other from one of the *models* of the database.

The first step in CSS matching is to shift one of the two sets of maxima so that the effect of any possible change in orientation is compensated. Since the exact value of the required shift is not available, the best choice is a value that shifts one CSS image so that its major maximum covers the major maximum of the other CSS image. Other possible choices include those values which accomplish the same with the second and possibly the third major maxima. For each case, the Euclidean distances between the nearest pairs of (three dimensional) maxima are calculated and the summation over the whole sets of maxima is considered as the matching value for that particular shift. The lowest matching value among different possible shifts is then selected as a measure of similarity between the two objects.

If the difference between the average curvature of a maximum of the model and its nearest maximum of the image is higher than a threshold, they are considered as unmatched maxima. For all unmatched maxima, the length of the projection on the $y - z$ plane is added to the matching value.

5 Results

We first tested the proposed method on a database of 1100 images of marine animals. Each image consisted of just one object on a uniform background. The system software was developed using the C language under Unix operating system. The response rate of the system was less than one second for every user query. [1] To have an objective evaluation of the performance of the method, we have created a small classified database which is a subset of our large database. There are 17 classes in this database, each consisting of about 8 objects. To evaluate the performance of the system on one object, it is chosen as the input and the system is asked to return the best 15 similar shapes from the database. The number of outputs from the same group is then considered as the *performance measure* of the system for that input. It is then possible to find the performance measure of the system on one group and then on all groups by extending this method.

Several classes are represented in Figure 4. For groups c, f and g, both methods showed a 100% result. It means that whenever a member of these classes was the input, all other members appeared in the first 15 outputs of the system. This figure for class a and class b were very low by the previous method and showed a considerable improvement by using the new method.

Group h consists of objects with sharp corners. Due to high curvature on corners, the average curvature on the relevant segments is always higher than the upper limit. Therefore, the new method seems to be less effective for such shapes. A small decrease in performance measure was observed for this group.

[1] A demo of our work is available at the following web site: http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html

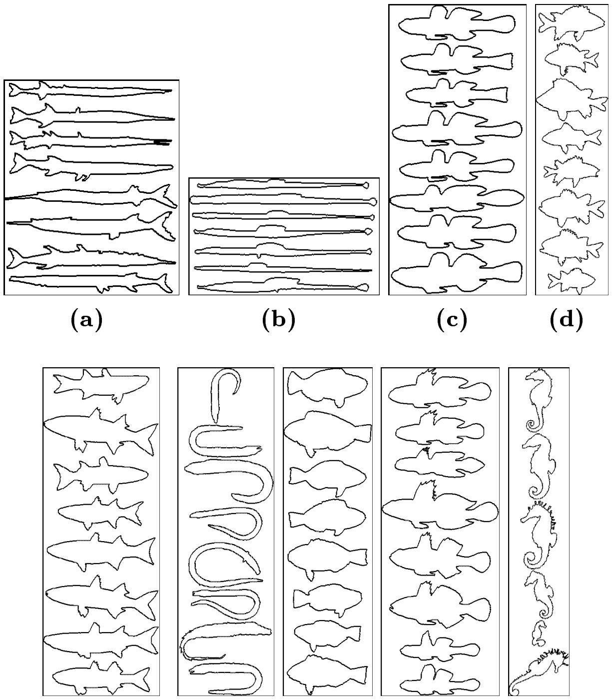

Figure 4: Several classes of shapes used for testing the methods.

In comparison to the height adjusted CSS image [2], the new method enjoys the advantages of simplicity and higher speed. However, the trade off will be a competitive but slightly lower performance measure.

6 Conclusions

This paper described a method to solve the problem of shallow concavities in Curvature Scale Space representation of planar curves. The method was tested on a database of 1100 images of marine creatures and a small database of aircrafts and helicopters. It was then tested on a small classified database to obtain an objective evaluation of its performance. The results showed a major improvement in performance over the old CSS matching. The improvement was particularly significant for shapes with shallow concavities.

Acknowledgements Sadegh Abbasi is on leave from the University of Guilan, Rasht, Iran. He is grateful to the Ministry of Culture and Higher Education of Iran for its financial support during his research studies.

References

[1] S. Abbasi, F. Mokhtarian, and J. Kittler. Reliable classification of chrysanthemum leaves through curvature scale space. In *Proceedings of the Scale-Space'97 Conference*, Utrecht, Netherlands, July 1997.

[2] S. Abbasi, F. Mokhtarian, and J. Kittler. Shape similarity retrieval using a height adjusted curvature scale space image. In *Proceedings of 2nd International Conference on Visual Information Systems*, San Diego, California, USA, 15-17 December 1997.

[3] P. Eggleston. Content based feature indexing and retrieval for image databases. *SPIE*, 1819, 1992.

[4] E. Gary and R. Mehrorta. Shape similarity-based retrieval in image database systems. In *Proceedings of Image Storage and Retrieval Systems*, volume 1662, pages 2–8, San Jose, CA, USA, Feb 1992. SPIE.

[5] F. Mokhtarian. Silhouette-based isolated object recognition through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):539–544, May 1995.

[6] F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space. In *Proceedings of the First International Workshop on Image Database and Multimedia Search*, pages 35–42, Amsterdam, The Netherlands, August 1996.

[7] F. Mokhtarian, S. Abbasi, and J. Kittler. Robust and efficient shape indexing through curvature scale space. In *Proceedings of the sixth British Machine Vision Conference, BMVC'96*, volume 1, pages 53–62, Edinburgh, September 1996.

[8] F. Mokhtarian, S. Abbasi, and J. Kittler. A new approach to computation of curvature scale space image for shape similarity retrieval. In Alberto Del Bimbo, editor, *Image Analysis and Processing, Proceedings of 9th Internation Conference*, volume 2, pages 140–147, Florence, Italy, September 1997. Speringer.

[9] F. Mokhtarian and A. K. Mackworth. A theory of multi-scale, curvature-based shape representation for planar curves. *IEEE Trans. on Pattern Anal. Mach. Intell.*, PAMI-14:789–805, August 1992.

[10] W. Niblack et al. The qbic project; querying images by content using color texture and shape. *SPIE*, 1908, 1993.

[11] R. Yong, A.C. She, and T.S. Huang. Modified fourier descriptors for shape representation, a practical approach. In *Proceedings of the First International Workshop on Image Databases and Multimedia Search*, pages 115–122, Amsterdam, The Netherlands, Aug 1996.