Shape-based Retrieval on a Fish Database of Taiwan

Chwen-Jye Sze *, Hsiao-Rong Tyan ** , Hong-Yuan Mark Liao * ,
Chun-Shien Lu * and Shih-Kun Huang *

*Institute of Information Science, Academia Sinica, Taipei, Taiwan
**Institute of Computer Science and Information Engineering,
Chung Yuan Christian University, Chung-Li, Taiwan

Abstract

"Digital Museum/Library" is an ongoing project sponsored by the National Science Council of Taiwan. The content of this paper is partial result of one of its subprojects called `` Image content-based retrieval on a fish database of Taiwan." The objective of this project is to develop an image content-based search engine which can perform identity check of a fish. It is well known that conventional fish databases can only be retrieved by text-based query. In this study we shall use shape, color, and other features extracted from a fish to search the fish database of Taiwan. The developed technique is able to perform scale, translation, and rotation invariant matching between two fishes. Currently, the database contains several hundreds of fishes. In the future, we shall enhance the capability of the search engine to deal with more than 2,000 fish species, which is the total amount of fish species along the coast of Taiwan.

Key words: Content-based image retrieval, Multimedia, Curvature scale space

1. Introduction

Content-based Multimedia Retrieval [1-4,6,7,9-12,] has become one of the most popular research areas in the past decade. The domain of multimedia covers a wide variety of media, including images, videos, texts, audios, graphics, and so on. Among the existing media-retrieval techniques, content-based image retrieval (CBIR) [1-4,6,7,9-12,] is an area which started earlier and thus has better progress. In contrast to conventional text-based retrieval, a CBIR system uses image content instead of text to retrieve the counterparts in the database. In general, there are two ways to retrieve information from a database: one is the global approach which uses the complete information contained in an image to search the database; the other is the local approach which selects a region-of-interest (ROI) as the base to perform search. The advantage of the former is that less human intervention is involved, but at the sacrifice of retrieving relatively incorrect data due to introducing too much irrelevant information for querying. The disadvantage of the latter is that human intervention is needed, but the advantageous point is that the retrieving process is much better in terms of efficiency and correctness. However, using a device such as a mouse to choose an ROI still cannot completely and accurately locate a target object; for example, when one uses a mouse to draw a bounding rectangle around a fish(as shown in Figure 1), there are still some irrelevant data included. In order for performing a correct retrieving process, the background part should be excluded. After the foreground/background separation problem is resolved, one has to deal with the scale, translation, and rotation problem so that the comparison can be made under a fair base.

This paper will seriously address the above mentioned problems. In the foreground/background separation process, the characteristics of a Gaussian distribution will be used to determine where the background pixels are. Bascially, these pixels should be discarded before the search process is activated. As for finding a scale, translation,

and rotation invariant representation for a fair comparison, we shall propose a new representation scheme which is an improvement of the Mokhtarian et al.'s curvature scale space (CSS) representation scheme[9]. In 1996, Mokhtarian et al.[8,10] proposed an excellent representation scheme known as the CSS representation scheme. They calculated the curvature magnitudes of all the points along the silhouette of a target object. The problem associated with the CSS representation scheme is that when a different starting point is adopted, the resulting CSS map will be completely different. However, it is also well known that up to now there is still no systematic means to uniquely determine a starting point. Under these circumstances, Mokhtarian et al.[8,10] used a number of complicated heuristics to solve the above mentioned problem. In this paper, we propose a new representation scheme, called “circular vector map”, to solve the problem. The new scheme first locates the maximum vectors existent in the CSS map. These vectors are then mapped onto a circle based on every vector's distance to the starting point. After the mapping, we call the circle which contains the set of mapped vectors a circular vector map. Based on the concept of force equilibrium borrowed from physics, we know there always exists a representative vector which can neutralize (i.e., cancel out) the total vector generated by adding up all the circular vectors on the map. For every target object, we rotate its representative vector to align with the positive x-axis. We call the orientation where the representative vector is rotated to - the “canonical orientation”. When all the model objects are adjusted to the ``canonical orientation,'' the subsequent comparisons can be executed efficiently and accurately. In this paper, we shall use the shape-based retrieval methodology on a fish database of Taiwan as an example to show the effectiveness of our approach.

The rest of the paper is organized as follows. In Section 2, we shall discuss how to convert the background pixels into a Gaussian distribution and then use the statistics of this distribution to perform foreground/background separation. Section 3 will first provide a brief review on the curvature scale-space representation scheme. Then, the means of transforming a CSS map into a circular vector map and the way to calculate the representative vector based on the force equilibrium concept will be introduced. Section 4 will show a series of experiments to support our theory. Concluding remarks will be given in Section 5

.

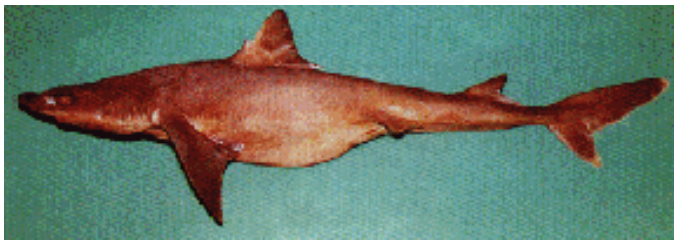


Figure 1. A fish selected by a mouse is enclosed by a bounding rectangle.

2. Foreground/Background Separation

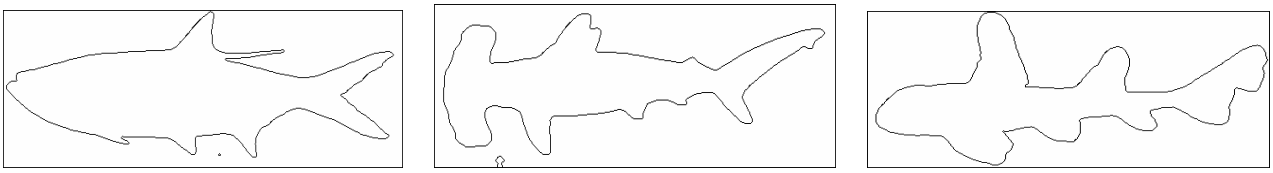
It is well known that image segmentation has been an open problem since it was first defined due to its ill-posed nature. Some of algorithms might be able to solve the segmentation problem in a specific domain while losing its power in another domain. Therefore, it is reasonable to introduce some assumptions on the domain of images to be processed. In this paper, we assume that the fish body of a query image should be located in the middle part of the image. Therefore, it is reasonable to assume that the boundary regions are background. We randomly select part of the background pixels and assume that they form a Gaussian distribution. Using two times of the variance of this distribution as a threshold, we are able to decide on the categories of the remaining pixels. If a pixel value deviates from the mean of the distribution by a magnitude larger than the pre-determined threshold, then it is classified as a foreground pixel; otherwise, it is a member of the background. Using this simple rule, we are able to perform foreground/background separation correctly in most cases. However, for some cases such as transparent fish body or extremely complicated background, the above mentioned foreground/ background separation rule may fail. Figure 2 shows some results of the foreground /background separation process.



(a) Some fish images captured by bounding rectangles.



(b) Results after foreground/background separation.



(c) Results after contour tracing.

Figure 2.

3. A New Representation Scheme for Shape Matching

3.1 Review of the CSS representation scheme

In the beginning of this section, we give a brief review of the curvature scale space theory[8,10]. Let $\vec{L}(t) = (u(t), v(t))$ be a 2D curve on a plane. The range of the parameter t is normalized between 0 and 1. The curvature of (u(t); v(t)) along curve $\vec{L}$ can be computed by the following equation[8,10]:

$$k(t) = \frac{\frac{\partial^2 u}{\partial t^2} \cdot \frac{\partial v}{\partial t} - \frac{\partial^2 v}{\partial t^2} \cdot \frac{\partial u}{\partial t}}{[(\frac{\partial u}{\partial t})^2 + (\frac{\partial v}{\partial t})^2]^{\frac{3}{2}}}. \quad (1)$$

There are several existing approaches[12] that can be used to compute the curvatures of a digital curve. Curve evolution[8] is one of the famous and effective approaches. Let g(t,σ) be a 1-D Gaussian of width σ, then an evolved curve $\vec{L}(t,\sigma)$ of $\vec{L}(t)$ can be represented by (U(t, σ), V (t,σ)), where

$$U(t; oe) = u(t) * g(t, \sigma),$$

and

$$V(t, \sigma) = v(t) * g(t, \sigma):$$

The "*" sign indicates "convolution". In order for calculating curvatures, the first derivatives and the second derivatives of U and V with respect to t, i.e, U_t, V_t, U_{tt}, and V_{tt}, should be determined in advance. They are

$$U_t(t,\sigma) = \frac{\partial U(t,\sigma)}{\partial t} = u(t) * \frac{\partial g(t,\sigma)}{\partial t},$$

$$V_t(t,\sigma) = \frac{\partial V(t,\sigma)}{\partial t} = v(t) * \frac{\partial g(t,\sigma)}{\partial t}$$

$$U_{tt}(t,\sigma) = \frac{\partial^2 U(t,\sigma)}{\partial t^2} = u(t) * \frac{\partial^2 g(t,\sigma)}{\partial t^2}$$

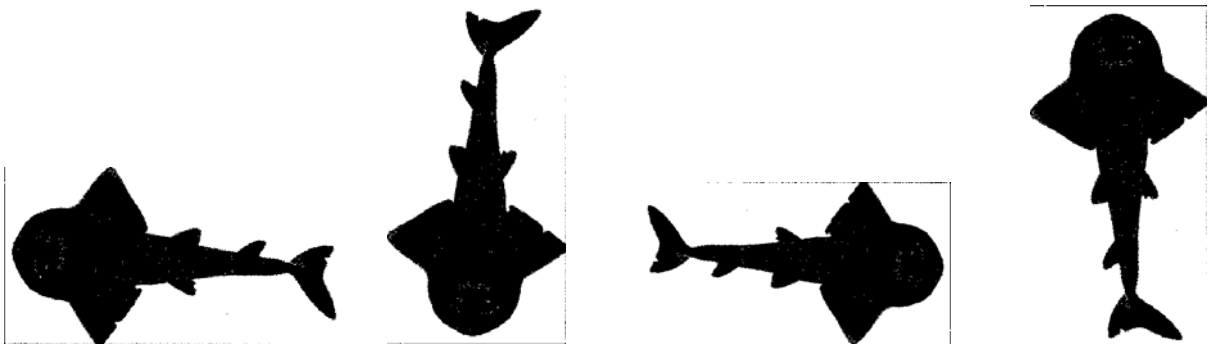
$$V_{tt}(t,\sigma) = \frac{\partial^2 V(t,\sigma)}{\partial t^2} = v(t) * \frac{\partial^2 g(t,\sigma)}{\partial t^2}$$

Therefore, the curvature $\kappa(t,\sigma)$ of $\vec{L}(t,\sigma)$ can be computed by the following equation[8]:

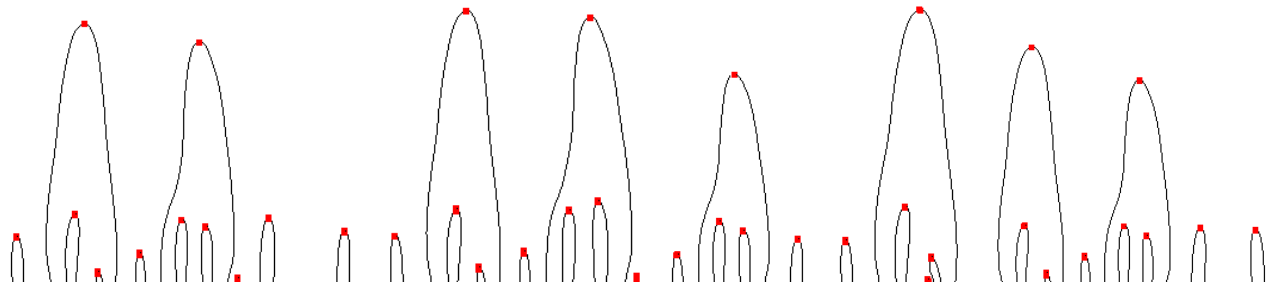
$$\kappa(t,\sigma) = \frac{U_{tt}(t,\sigma) \cdot V_t(t,\sigma) - V_{tt}(t,\sigma) \cdot U_t(t,\sigma)}{[U_t(t,\sigma)^2 + V_t(t,\sigma)^2]^{\frac{3}{2}}} \quad (2)$$

In [9], the implicit equation k(t, σ) = 0 defined the curvature scale space (CSS) image of $\vec{L}$. Mokhtarian et al.[8,10] extracted the maxima from the CSS contours as feature points to describe shapes of object boundary contours. This is due to their capability on characterizing the differences between two shapes. Figures 3(a)-(d)show a fish with Fig. four different orientations. The corresponding CSS maps of Figures 3(a)-(d) are shown in Figures

3(e)-(h), respectively. The black dots shown in Figures 3(e)-(h) are used to indicate the maximum points of the contours in CSS maps. From Figure 3, it is obvious that the distributions of the maximum points are very similar. However, the corresponding points which are necessary for matching are still difficult to be located. Due to the discrete nature caused by the digitization process, the curvatures of the same point which are obtained from different orientations may be significantly different. Mokhtarian and Mackworth[8,10] proposed a complicated algorithm which used a number of heuristics to deal with the above mentioned problem. However, a systematic means which can solve the problem in an efficient way is always preferable. In this paper, we propose a regularization method to adjust the orientation of objects which are to be compared into canonical orientations. Once all objects are adjusted to canonical orientations, the comparisons between any pair of objects can be performed easily. In what follows, the details of our approach will be introduced.



(a) $\theta = 0^o$ (b) $\theta = 90^o$ (c) $\theta = 180^o$ (d) $\theta = 270^o$



(e)CSS map of (a) (f)CSS map of (b) (g)CSS of map(c) (h) CSS of map(d)

3.2 Circular Vector-based Representation and Orientation Regularization

In this section, we shall describe how one can map a CSS map onto a circular vector map. Circular vector-based representation will be introduced first. Then, the theory about how to use the circular vector-based representation scheme to perform orientation regularization will be elaborated. Let $\vec{a}_0$ and $\vec{b}_0$ be two distinct vectors in 2D space with non-zero lengths. From the principle of force equilibrium, it is known that there exists a vector $\vec{c}_0$ which can always make $\vec{a}_0 + \vec{b}_0 + \vec{c}_0 = \vec{0}$. If both $\vec{a}_0$ and $\vec{b}_0$ are rotated by an unknown angle θ and assume their rotated versions are $\vec{a}_\theta$ and $\vec{b}_\theta$, respectively. Then, there must exist a vector $\vec{c}_\theta$ such that $\vec{a}_\theta + \vec{b}_\theta + \vec{c}_\theta = \vec{0}$. The value of θ can be computed by the following equation:

$$\theta = cos^{-1}\frac{\vec{c}_0 \cdot \vec{c}_\theta}{|\vec{c}_0||\vec{c}_\theta|}. \quad (3)$$

Let R be a 2D rotation matrix such that

$$R(\theta) \cdot \vec{a}_0 = \vec{a}_\theta$$

and

$$R(\theta) \cdot \vec{b}_0 = \vec{b}_\theta.$$

In order to derive the above equation and make

it become a general form, let $\Gamma_0=\{\vec{f}_i|1 \le i \le N\}$ where N is the number of vector in this set. Let $\Gamma_\theta = \{\vec{g}_i|\vec{g}_i = \mathbf{R}(\theta)\cdot\vec{f}_i \text{ , } where\ 1 \le i \le N\}$ be the θ-gree rotated version of Γ_0 . If $\sum_{i=1}^{N}\vec{f}_i \neq \vec{0}$,

Then there exits a vector $\vec{F}_0$ satisfying

$$\sum_{i=1}^{N}\vec{f}_i + \vec{F}_0 = \vec{0}. \quad (4)$$

$\vec{F}_0$ in Eq. (4) can be called the principal vector of Γ_0
and its direction is known as the principal direction. Form similar reasoning one can expect that there also exits a vector$\vec{F}_\theta$ such that

$$\sum_{i=1}^{N}\vec{g}_i + \vec{F}_\theta = \vec{0}. \quad (5)$$

Under the circumstances, the rotation angle θ , can be determined by

$$\theta = cos^{-1}\frac{\vec{F}_0\cdot\vec{F}_\theta}{|\vec{F}_0|\cdot|\vec{F}_\theta|} \quad (6)$$

Since all the vectors in Γ_θ and $\vec{F}_\theta$ in Eq.(6) can be brought back to the original principal direction by $-\theta$, we have

$$\mathbf{R}(-\theta)\cdot[\sum_{i=1}^{N}\vec{g}_i + \vec{F}_\theta] = \sum_{i=1}^{N}[\mathbf{R}(-\theta)\cdot\vec{g}_i] + \mathbf{R}(-\theta)\cdot\vec{F}_\theta = \vec{0}.$$

Because $\mathbf{R}(-\theta)\cdot\vec{F}_\theta = \vec{F}_0$, one can derive the following equation

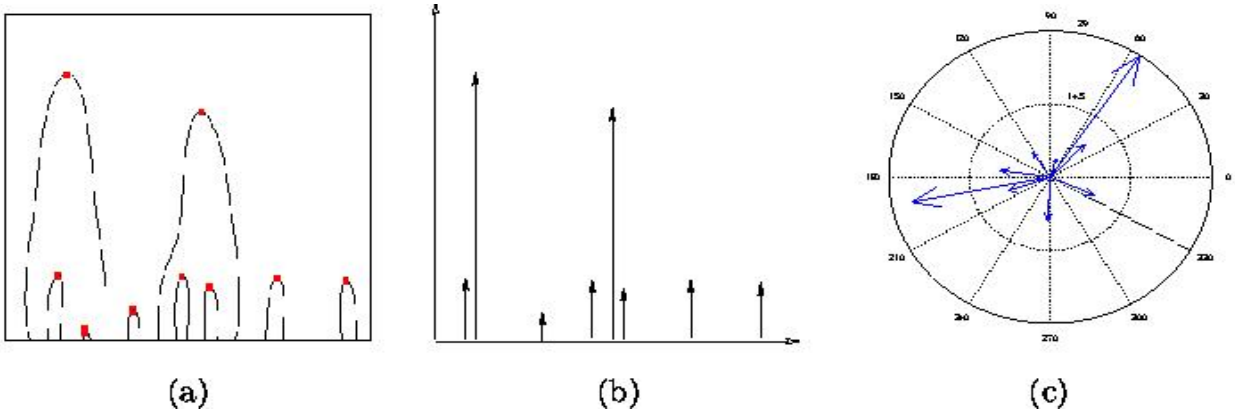
$$\sum_{i=1}^{N}[\mathbf{R}(-\theta)\cdot\vec{g}_i] = \sum_{i=1}^{N}\vec{f}_i \quad . \quad (7)$$

One can say Γ_θ is "regularized" to Γ_0 for the purpose of fair matching. The standard orientation, Γ_θ, is known as the canonical orientation. In what follows, we shall introduce how the abovementioned method can be applied to the shape matching problem which is crucial in content-based image retrieval.

We have mentioned that the CSS map is basically circular shift[8,10]. Figure 4(a) shows the curvature scale-space map of a curve. The vertical axis indicates the magnitude of the variance, located curvature maximum points, and the horizontal axis represents a normalized range of t such that $0 \leqq t \leqq 1$. The curves located in Figure 4(a) represents the maxima of curvature scale space contours. If we convert the located maximum in Figure 4(a) into local maximum vectors, then the outcome is shown in Figure 4(b). We propose a new representation scheme called circular vector map as illustrated in Figure 4(c). The characteristics of a circular vector map are described as follows. First, the normalized range of a curve can be changed from [0, 1] to [0, 2π]. Second, the located maximum vectors (Figure 4(b)) can be projected onto a circular vector map (Figure 4(c)). The orientation of every located maximum vector shown in Figure 4(b) can be determined by calculating its distance from the starting point and then project it accordingly onto the circular vector map shown in Figure 4(c). Based on the above mentioned circular vector map, we can derive a new shape matching strategy. Let $\Lambda = \{\vec{v}_i = (\theta_i, r_i)|1 \le i \le M \text{ and } 0 \le r_i \le 1\}$ be a set of located curvature maximum vectors which have been projected onto a circular vector map. M indicates the number of elements in the set and r_i represents the length of $\vec{v}_i$. Equation (3) tells us that there always exists a vector $\vec{V}$
such that

$$\sum_{i=1}^{M}\vec{v}_i + \vec{V} = \vec{0}. \quad (8)$$

The $\vec{V}$ vector in Equation (8) is a vector that equilibrates the total vector $\sum_{i=1}^{M}\vec{v}_i$ in the sense of force equilibrium. In other words, we can always use $\vec{V}$ to represent the total vector $\sum_{i=1}^{M}\vec{v}_i$, which was generated by adding up all vectors located on the circular vector map. Figure 5 is an example showing the representative vector derived from a set of vectors on a circular vector map. Since the comparison may be made under a condition that both shapes are obtained from two distinct categories, one has to make the representative vectors of all the model shapes in database align with the positive x-axis. That is, the orientation of all representative vectors of models should have 0 degree. In order for making comparison between a model shape and an unknown shape, one has to calculate the CSS of an unknown shape first. Then, the set of corresponding local maximum vectors has to be determined. Once all the local maximum vectors are decided, they will be mapped onto the circular vector map. Based on the vectors shown on the map, one can calculate a representative vector that will equilibrate the above mentioned vectors. This representative vector has to be rotated such that it is aligned with the positive x-axis. The final step of the comparison procedure is to compare the two shapes at the pre-determined canonical orientations.



(a) (b) (c)

Figure 4. (a)Curvature scale-space map (b)Local maximum vector (c)Circular vector map

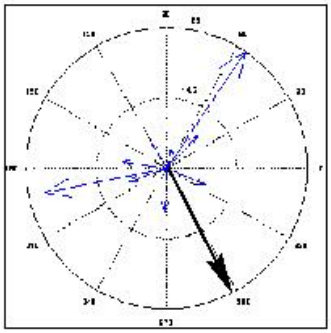


Figure 5. A representative vector (bold line) calculated from the set of vectors located on a circular vector map.

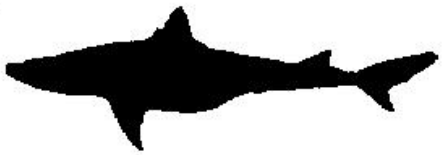


Figure 6. Atest fish shape

4. Experimental Results

In the experiments, we use fish images to demonstrate the effectiveness of our approach. Figure 6 is a test fish image. Assume the contour tracing algorithm was started at three different locations (the situation happens all the time), then it is possible to generate three completely different CSS maps as shown in Figures 7(a), (e), and (i), respectively. Figures 7(b), (f), and (j) were the three circular vector maps corresponding to Figures 7(a), (e), and (i), respectively. Based on the circular vector maps of (b), (f), and (j), their corresponding representative vectors were shown in Figures 7(c), (g), and (k), respectively. After aligning the three representative vectors shown in Figures 7(c), (g), and (k) with the positive x-axis, their corresponding circular vector maps became the ones shown in Figures 7(d), (h), and (l), respectively. Under the circumstances, any comparison can be made easily and efficiently. Figures 8 and 9 show two fish images with each rotated by 90^{o} . It is obvious that the regularized circular vector maps shown in Figures 8(c), (f) were very close to each other if they were obtained from the same object but different instances. The example shown in Figure 9 was another example.

In what follows, we shall show the system interface of our current fish query system. Fig. 10 illustrates a fish querying example. The fish at the left hand side was the query fish which was the result after performing foreground/background separation. When one clicks the box under the query fish, the system performs contour tracing and then calculates the CSS map. Next, the system will convert the CSS map into the circular vector map and perform orientation regularization. The regularized circular vector map was then compared with the map of every fish kind in the database. The right hand side of Fig. 10 shows the comparison results from top to bottom based on the similarity degree with the query fish. If one would like to know more details about one of the retrieved fish kinds, he/she needs to click the rectangle box right after the retrieved fish image. Fig. 11 shows how to further explore the detail information of a retrieved fish kind. Fig. 12 is another querying example.

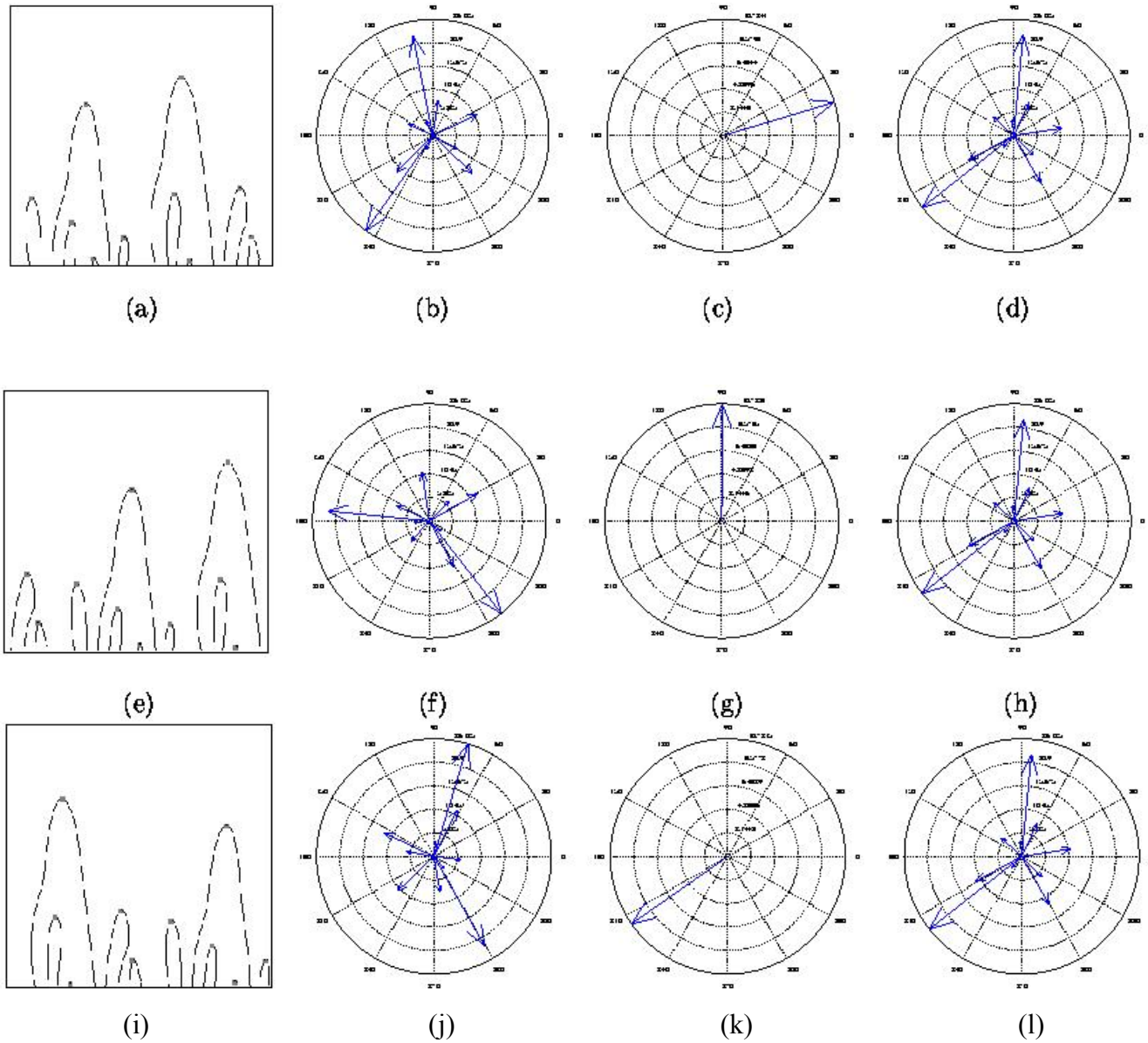
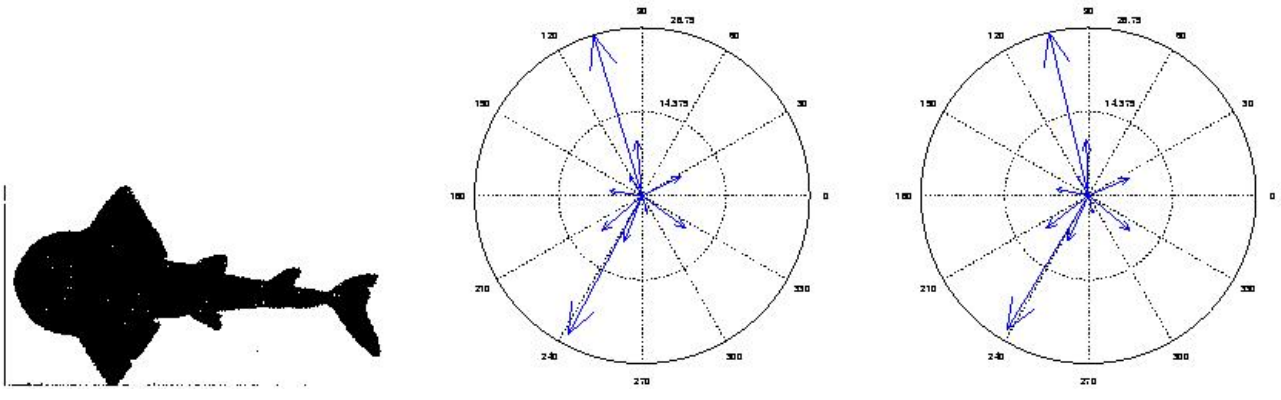


Figure 7. (a) CCS map No 1. (b) Circular vector map of (a). (c) Representative vector of (b). (d) Regularized circular vector map of (c). (e) CCS map No 2. (f) Circular vector map of (e). (g) Representative vector of (f). (h) Regularized circular vector map of (f). (i) CCS map No 3. (j) Circular vector map of (i). (k) Representative vector of (j). (l) Regularized circular vector map of (j).



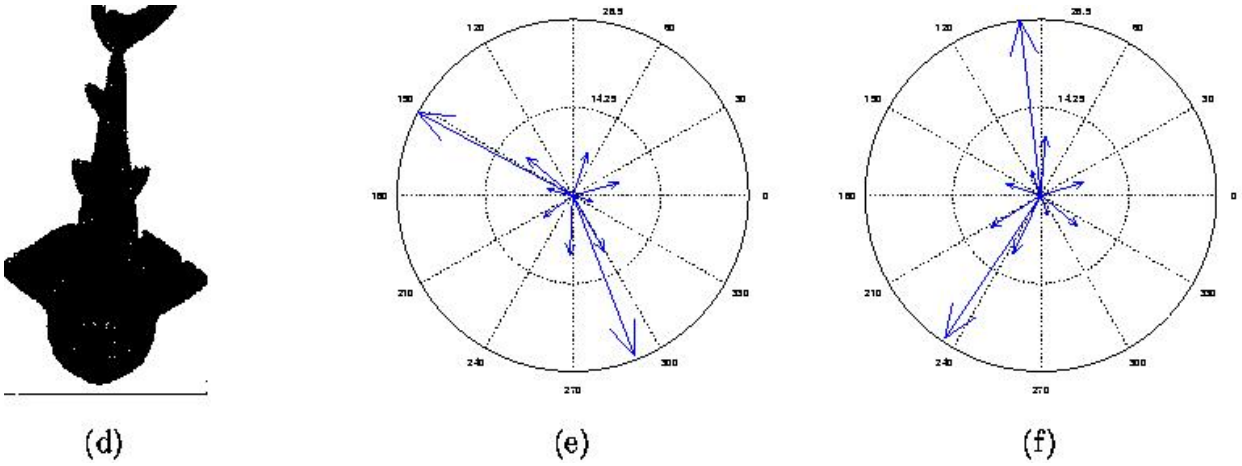


Figure 8. An example of rotation case. (a) 0 ^{o} . (b) Circular vector map of (a). (c) Regularized circular vector map of (b). (d) 90 ^{o} . (e) Circular vector map of (d). (f) Regularized circular vector map of (e).

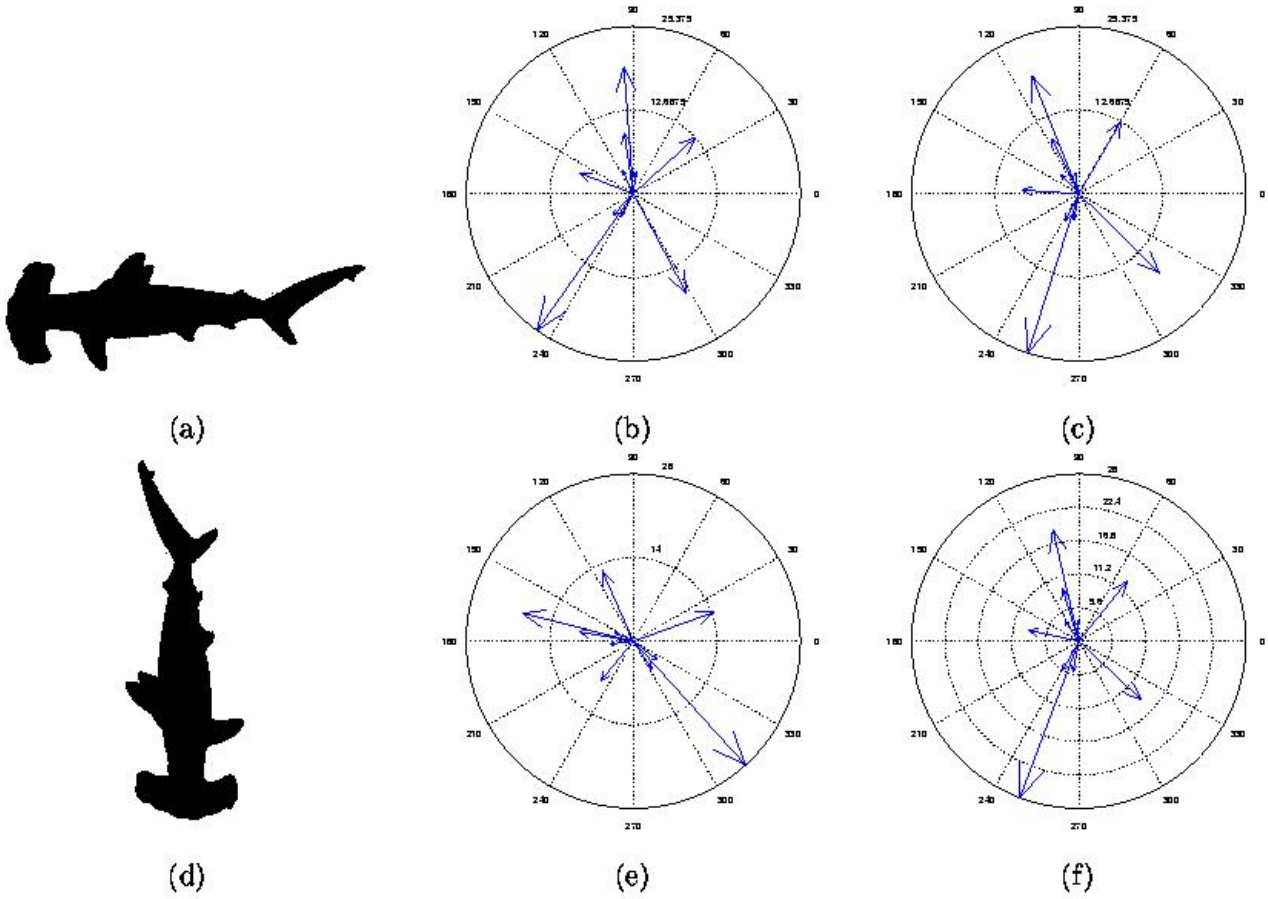


Figure 9. Another example. (a) 0 ^{o} . (b) Circular vector map of (a). (c) Regularized circular vector map of (b). (d) 90 ^{o} . (e) Circular vector map of (d). (f) Regularized circular vector map of (e).

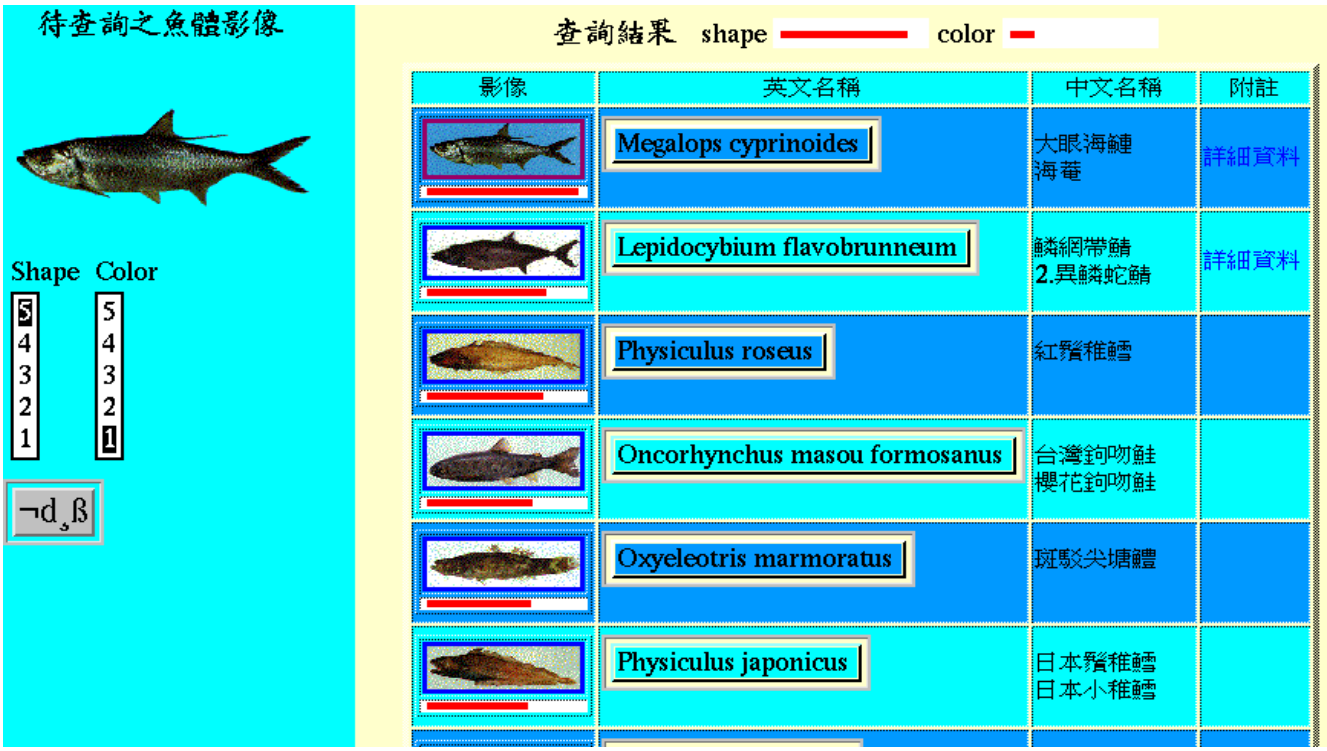


Figure 10. A querying example.

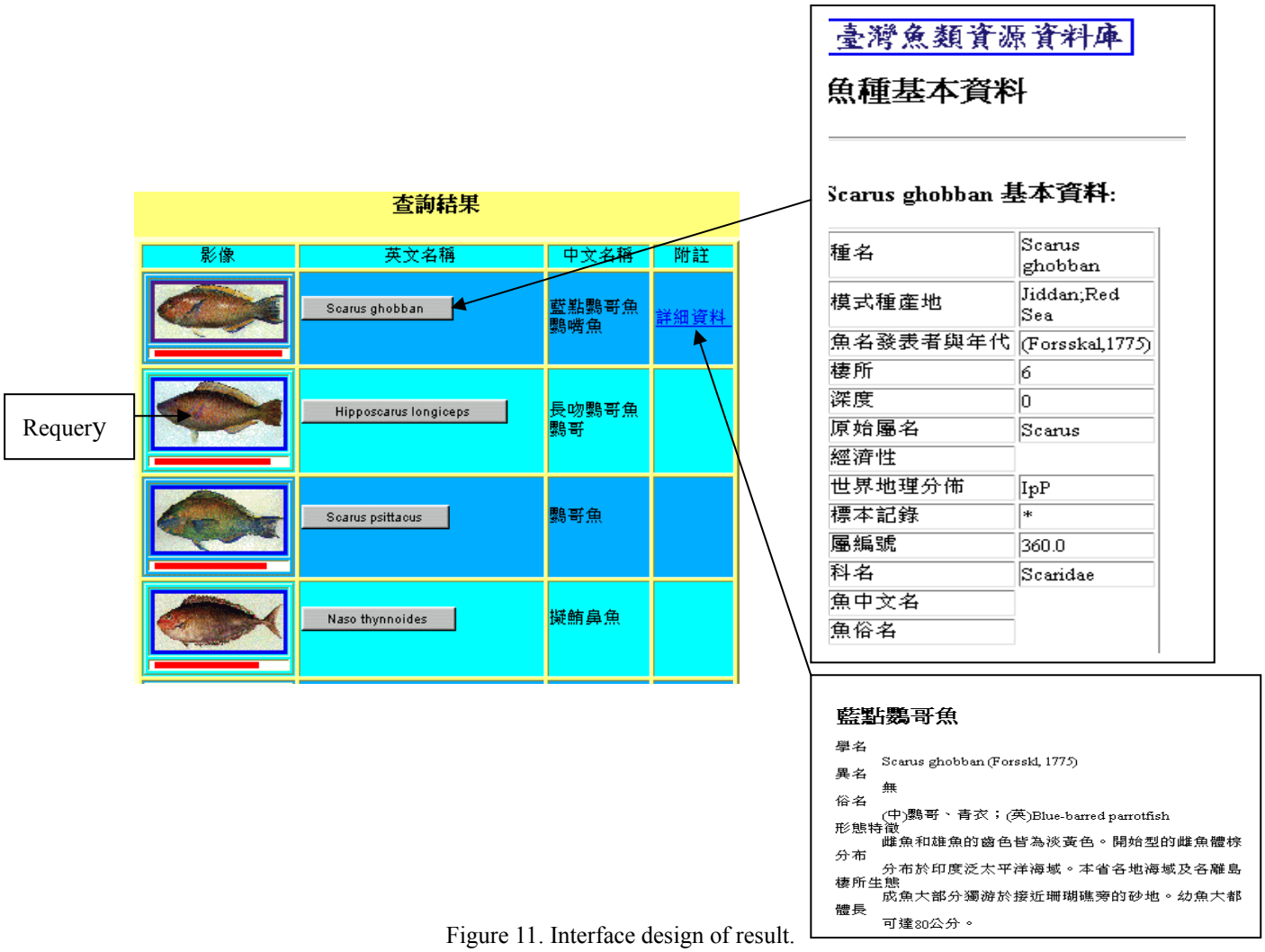


Figure 11. Interface design of result.

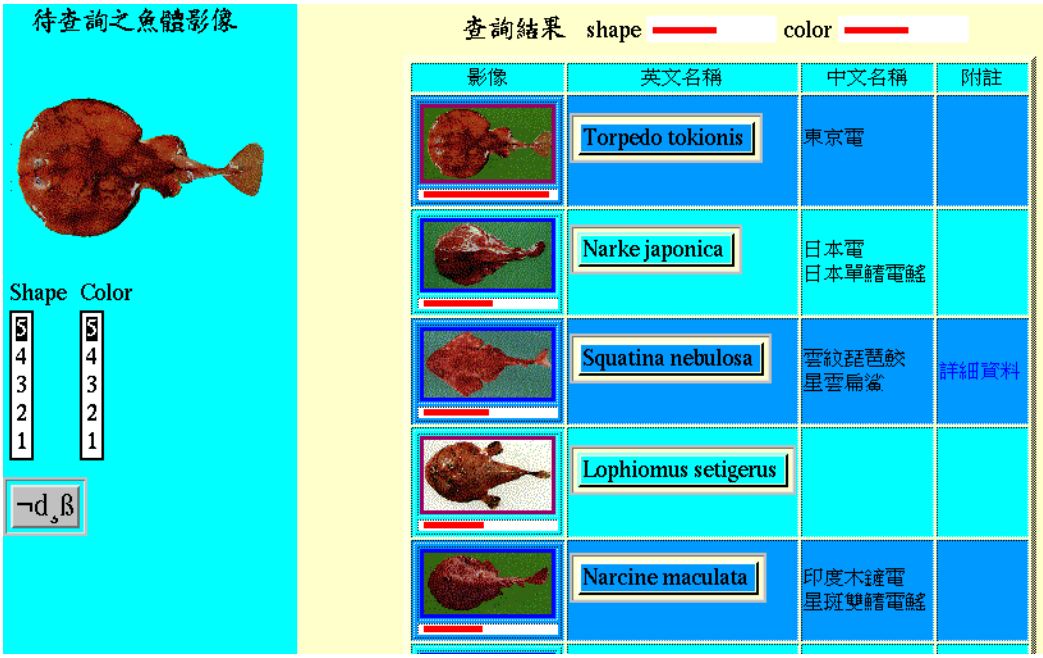


Figure 12. Another querying example.

5. Conclusion

In this paper, a new approach has been proposed to deal with the shape-based retrieval problem. A target object selected by a bounding rectangle has to be processed by a foreground/background separation step. The target object (foreground part) is then converted into a CSS map. In order for performing rotation invariant matching, we further convert the CSS map into a circular vector map and then find its representative vector based on the concept of force equilibrium. After rotating the representative vector into the canonical orientation, every unknown object can be compared with the model objects efficiently. The proposed scheme makes the performance of a CSS-based system much better because every comparison was made at a standard canonical orientation. Experiments have confirmed that our approach is indeed superb.

Reference

[1] Eakins, J. P., Boardman, J. M., and Graham, E., "Similarity retrieval of trademark images," *IEEE Multimedia*, April-June (1998).

[2] Flicker, M. and et al., "Query by image and video content: The qbic system," *Computer*, pp. 23-32, (1995).

[3] Gudivada, V. N. and Raghavan, V. V., "Content-based image retrieval systems," *Computer*, pp. 18-22, (1995).

[4] Huang, C. L. and Huang, D. H., "Content-based retrieval system," *Image and Vision Computing*, Vol. 67, pp. 149-163, (1998).

[5] Jain, A. K. and Vailaya, A., "Shape-base retrieval: A case study with trademark image database", *Pattern Recognition*, Vol. 31, No. 9, pp. 1369-1390, (1998).

[6] Kim, Y. S., and Kim, W. Y., "Content-based trademark retrieval system using a visually salient feature, " *Image and Vision Computing*, Vol. 16, pp. 931-939, (1998).

[7] Mehrotra R., and Gary, J. E., "Similar-shape retrieval in shape data management," *Computer*, pp. 57-62, September (1995).

[8] Mokhtarian, F., Abbasi, S. and Kittler, J., "Robust and efficient shape indexing through curvature scale space," in *Proceedings of the 1996 British Machine and Vision Conference BMVC'96*, Edinburgh, U.K. , pp. 53-62, September (1996).

[9] Mokhtarian, F. and Mackworth, A., "A theory of multiscale, curvature-based shape representation for planar curves," *IEEE Transcations on PAMI*, Vol. 14, No. 8, pp. 789-805, Aug. (1992).

[10] Mokhtarian, F., Abbasi, S., and Kittler, J., "Efficient and robust retrieval by shape content through curvature scale space," in *Proceedings of International Workshop on Image Databases and Multimedia Search*, Amsterdam, pp. 35-42, Aug. (1996).

[11] Srihari, R. K., "Automatic indexing and content-based retrieval of captioned images," *Computer*, pp. 49-56, (1995).
[12] Wu, J. K., Narasimhalu, A. D., Mehtre, B. M., Lam, C. P., and Gao, Y. J., "Core: A content-based retrieval engine for multimedia information systems," *Multimedia Systems*, Vol. 3, pp. 25-41, (1995).

Accepted: Sept. 10, 1999