INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Contour Salience Descriptors for Effective Image Retrieval and Analysis

Ricardo da S. Torres *Alexandre X. Falcão*

Technical Report - IC-04-011 - Relatório Técnico

October - 2004 - Outubro

Contour Salience Descriptors for Effective Image Retrieval and Analysis

Ricardo da S. Torres* Alexandre Xavier Falcão*

Abstract

This work exploits the resemblance between content-based image retrieval and image analysis with respect to the design of image descriptors and their effectiveness. In this context, two shape descriptors are proposed: *contour saliences* and *segment saliences*. Contour saliences revisits its original definition, where the location of concave points was a problem, and provides a robust approach to incorporate concave saliences. Segment saliences introduces *salience values* for contour segments, making it possible to use an optimal matching algorithm as distance function. The proposed descriptors are compared with convex contour saliences, *curvature scale space*, and *beam angle statistics* using a fish database with 11,000 images organized in 1,100 distinct classes. The results indicate segment saliences as the most effective descriptor for this particular application and confirm the improvement of the contour salience descriptor in comparison with convex contour saliences.

1 Introduction

Recent technological improvements in image acquisition and storage have supported the dissemination of large databases, where the design of information retrieval systems based on image properties becomes a challenge [34]. In these Content-Based Image Retrieval (CBIR) systems, image properties are usually represented by shape, color, and texture of objects/regions within the image. A CBIR system essentially consists of an image database, a *descriptor*, and a data structure for image indexation. The descriptor is a pair, *feature vector* and *distance metric*, used for image indexation by similarity. The feature vector subsumes the image properties and the distance function measures the dissimilarity between two images with respect to their properties. Each image can be interpreted as a "point" in the underlying metric space, where similar images form groups of points. For given user-defined specification or pattern (e.g., shape sketch, query image), the CBIR system aims at retrieving groups of similar images which are relevant to the query (*effectiveness*) as fast as possible (*efficiency*). Clearly, the efficiency of the system depends on the indexing structure (e.g., a Metric Access Method [7, 35]) and on the complexity of the distance function, while its *effectiveness* is solely related to the ability of the descriptor in representing

*Institute of Computing, University of Campinas, Av. Albert Einstein, 1251, CEP 13084-851, Campinas, SP, Brasil

distinct groups of relevant images as far as possible in the metric space. That is, different descriptors define different CBIR systems with distinct degrees of effectiveness, where the goal of research is to find the descriptor with maximum effectiveness for given application. The descriptors are also important in image analysis, where the groups of relevant images form classes or patterns for recognition [15]. The present paper is mainly concerned with shape descriptors and their effectiveness for image retrieval and analysis.

Costa *et al.* [8] proposed the use of *shape saliences* for object representation. The saliences of a shape are defined as the maximum influence areas of its higher curvature points, considering a *narrow band* in both sides of the curve and the Voronoi regions of its points. A contour point, for example, is considered convex when its influence area is greater outside than inside the contour, and concave otherwise. The narrow band is used to reduce as much as possible cross-influence of opposite parts of the curve, which come close to each other. Torres *et al.* [10] presented a more efficient way to compute shape saliences using the *image foresting transform* [20] and a contour salience descriptor for image retrieval [12] and analysis [11]. In both works, the contour salience descriptor was compared with several other shape descriptors, including the popular *curvature scale space* [1, 31] and the recently proposed *beam angle statistics* [2, 3]. However, the contour salience descriptor never considered concave salience points, because its effectiveness was very sensitive to the precise location of these points. This work solves the problem, incorporating concave points to the contour salience descriptor. In addition, it proposes another shape descriptor based on the *salience values* of contour segments.

The methods use the image foresting transform to compute the salience values of contour pixels and to locate salience points along the contour by exploiting the relation between a contour and its internal and external skeletons [26]. The contour salience descriptor consists of the salience values of salient pixels and their location along the contour, and on a heuristic matching algorithm as distance function. The contour is also divided into a fixed number of segments and the influence areas of their pixels inside and outside the contour are used to compute *segment saliences*. The segment salience descriptor consists of the salience values of contour segments and an optimal matching algorithm as distance function.

The article describes the computation of shape saliences using the image foresting transform in Section 2. Section 3 provides a detailed description of the algorithm to locate salient contour pixels via multiscale skeletonization. The new contour and segment salience descriptors are presented in Section 4 and compared with the convex contour saliences, curvature scale space, and beam angle statistics in Section 5. Section 6 states the conclusion and discusses the current research on CBIR systems.

2 Shape saliences

The algorithm proposed by Costa *et al.* [8] to determine shape saliences is based on the concept of *Exact Dilation with Label Propagation (EDLP)*. The EDLP of a given labeled seed set S assigns to each image pixel t a value $C(t)$ and a label $L(t)$, which are the minimum Euclidean distance between t and S (Euclidean distance transform) and the label of its closest pixel in S (discrete Voronoi regions), respectively.

The EDLP algorithm can take contour pixels as seeds and determine the influence areas of each seed as the areas of its discrete Voronoi regions inside and outside the contour. The influence areas of higher curvature points, namely *salience points*, are expected to be greater than the influence areas of other contour pixels. Moreover, the influence area of a convex point (points A, B, D, and E in Figure 1) is greater outside than inside the contour, and the other way around is true for a concave point (point C in Figure 1). The influence area of each salience point relates to the aperture angle θ, illustrated in Figure 1, by the formula:

$$Area = \frac{\theta \times r^2}{2}, \tag{1}$$

where r is a dilation radius. Costa *et al.* [8] proposed to use as *salience value* of a contour point the maximum influence area between the areas computed outside and inside the contour for a low value of r (e.g., 10), in order to avoid cross-influence of opposite parts of the contour which come close to each other. They also suggested to locate the salience points along the contour by thresholding their salience values (i.e. $Area \geq \frac{\theta \times r^2}{2}$, for some value of θ).

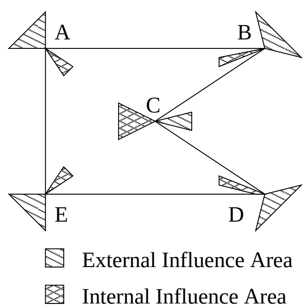


Figure 1: Internal and external influence areas of convex (A, B, D, and E) and concave (C) points.

2.1 Shape saliences by image foresting transform

Costa's algorithm [8] can be more efficiently implemented (in time proportional to the number of pixels) by using the Image Foresting Transform (IFT) [10]— a graph-based approach to the design of image processing operators based on connectivity [18–20, 28].

The IFT reduces image partition problems based on a given seed set to the computation of a *minimum-cost path forest* in a directed graph, whose nodes are the pixels and whose arcs are defined by an *adjacency relation* between pixels. The cost of a path in this graph is determined by an application-dependent *path-cost function*, which usually depends on local image properties along the path — such as color, gradient, and pixel position. For suitable path-cost functions, the IFT assigns to each image pixel a minimum-cost path from the seed set, such that the union of those optimum paths form an oriented forest spanning the whole image. The nodes of each rooted tree in the forest are composed by pixels that are "more closely connected" to its root pixel than to any other seed, in some appropriate sense. The IFT assigns to each pixel three attributes: its predecessor in the optimum path (predecessor map P), the cost of that path (cost map C), and the corresponding root (root map R) or some label associated with it (label map L).

For given set S of seed pixels, the IFT can provide the simultaneous computation of the Euclidean distance transform in the cost map C and of the discrete Voronoi regions in the root map R [20]. This operator asks for an Euclidean adjacency relation A and a path-cost function f_{euc} defined for any path $\pi =< p_1, p_2, ..., p_n >$ in the graph as:

$$q \in A(p) \Longrightarrow (x_q - x_p)^2 + (y_q - y_p)^2 \leq \rho^2, \tag{2}$$

$$f_{euc}(\pi) = \begin{cases} (x_{p_n} - x_{p_1})^2 + (y_{p_n} - y_{p_1})^2, & \text{if } p_1 \in S, \\ +\infty, & \text{otherwise,} \end{cases} \tag{3}$$

where ρ is the adjacency radius and (x_{p_i}, y_{p_i}) are the (x, y) coordinates of a pixel p_i in the image. Note that, the main idea is to find for every image pixel p_n a path $P^*(p_n)$ from a seed pixel $p_1 \in S$, such that $f_{euc}(P^*(p_n))$ is minimum. The exact Euclidean distance transform will depend on the appropriate choice of ρ, as demonstrated in [20]. However, for most practical situations involving 8-connected curves, such as contours and skeletons, $\rho = \sqrt{2}$ is enough [19]. Algorithm 1 below presents the IFT procedures for f_{euc}.

Algorithm 1:
Input: An image I, a set S of seed pixels in I, and an Euclidean adjacency relation A;
Output: An optimum-path forest P, and the corresponding cost map C and root map R.
Auxiliary Data structures: A priority queue Q.

1. For every pixel p of the image I, set $C(p) \leftarrow +\infty$;
2. For every $p \in S$, set $P(p) \leftarrow nil$, $R(p) \leftarrow p$, $C(p) \leftarrow 0$, and insert p in Q;
3. While Q is not empty, do

   3.1. Remove from Q a pixel $p = (x_p, y_p)$ such that $C(p)$ is minimum;

   3.2. For each pixel $q = (x_q, y_q)$ such that $q \in A(p)$ and $C(q) > C(p)$, do

   3.2.1. Set $C' \leftarrow (x_q - x_{R(p)})^2 + (y_q - y_{R(p)})^2$, where $R(p) = (x_{R(p)}, y_{R(p)})$ is the root pixel of p;

   3.2.2. If $C' < C(q)$, then

   3.2.2.1. If $C(q) \neq +\infty$, then remove q from Q.

   3.2.2.2. Set $P(q) \leftarrow p$, $C(q) \leftarrow C'$, $R(q) \leftarrow R(p)$, and insert q in Q.

Note that, the IFT algorithm is essentially Dijkstra's shortest-path algorithm [14], slightly modified to multiple sources and general path-cost functions. Its correctness for weaker conditions that are applied to only optimum paths in the graph is presented in [20].

A natural extension of this algorithm to compute contour saliences consists of obtaining one histogram of the resulting root map for each side of the contour, restricted to a small neighborhood of the curve in order to eliminate the cross-influence of its opposite parts. Each bin of the histograms indicates the area of influence of the respective root inside (or outside) the contour. The root is classified as *convex*, when the external area is greater than the internal area, and otherwise as *concave*.

As in the original approach [8], a point of the curve is classified as salient by thresholding its maximum influence area [10]. This approach, however, may miss important salience points when opposite parts of the contour come too close to each other, even for a small radius r in Equation 1. It has otherwise been particularly effective for skeletons and for simple contours, such as polygons, but it fails in finding the salience points of more complex and intricate curves. Torres *et al.* [11, 12] have proposed a partial solution for this problem, which is described next.

3 The use of skeletons for contour saliences

First, multiscale skeletons [19] are computed for the contour (Section 3.1), and one internal skeleton and one external skeleton are chosen by thresholding the multiscale skeletons. Second, the internal and external skeleton saliences are found similarly to as described in the previous section (Section 3.2). The location of the contour saliences are determined by relating the salience points of the internal skeleton to convex contour points and the salience points of the external skeleton to concave contour points (Section 3.3).

3.1 Multiscale skeletonization

Given a contour with N pixels, its internal skeleton is defined as the geometric location of the centers of maximal disks contained in the contour [24]. A similar definition is valid for the external skeleton.

Algorithm 1 applied to the contour creates a root map R. Multiscale skeletons [19] can be computed from R if each contour pixel p (root) is assigned to a subsequent label value $\lambda(p)$, varying from 1 to N, while circumscribing the contour (Figure 2a). A label map L can be created by computing $L(R(p))$ to each image pixel p (Figure 2b). A more efficient way, however, is to propagate the labels of the contour pixels during Algorithm 1. In this case, the labeling function λ is used in step (2), when the contour pixels are inserted in Q, and the label map L is created similarly and simultaneously to the root map R. A difference image D results from the label map L by computing the following for each pixel p inside and outside the contour (Figure 2c):

$$D(p) = \max_{\forall q \in A_4(p)} \{\min\{\delta(p,q), N - \delta(p,q)\}\}, \tag{4}$$

where $\delta(p,q) = L(q) - L(p)$ and $A_4(p)$ is the set of pixels q that are 4-neighbors of p. The difference image represents the multiscale internal and external skeletons by label propagation [9, 19]. One-pixel wide and connected skeletons can be obtained by thresholding the difference image at subsequent integer values (Figures 2d-f). The higher the threshold value, the more simplified the skeletons become, with smaller details being progressively removed as the threshold increases.

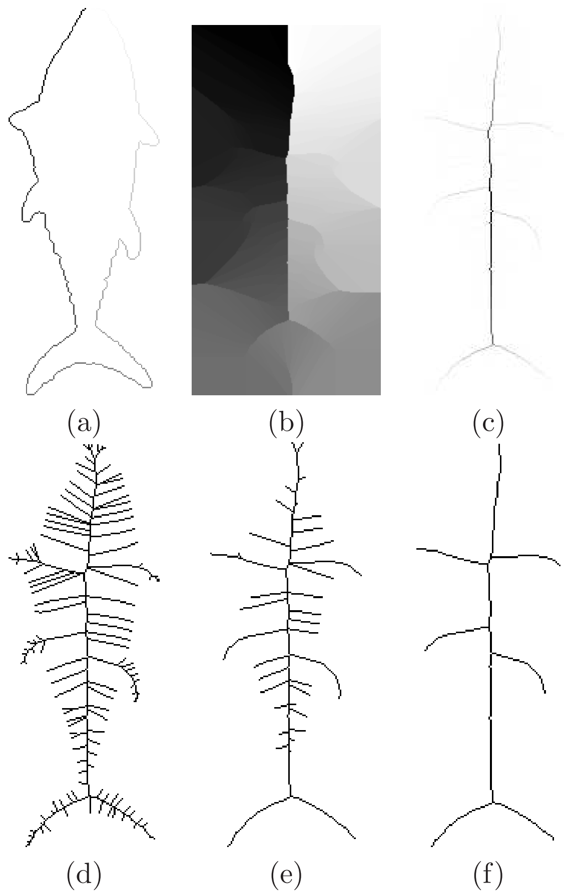
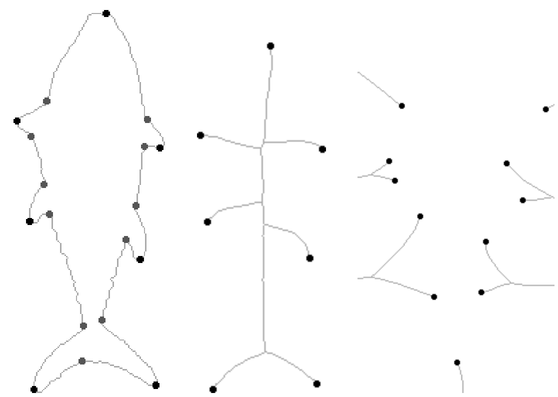


Figure 2: Multiscale skeletonization by label propagation inside a contour. (a) Labeled contour, (b) label map, (c) difference image, and (d-f) internal skeletons at three different scales.

3.2 Skeleton saliences

For small scales (low thresholds – e.g., 5% of the number N of contour pixels), each salience point of the internal skeleton corresponds to one convex point of the contour and each salience point of the external skeleton corresponds to one concave point of the contour (see Figure 3). The salience points of the skeletons are determined similarly to as described in Section 2.1 by taking the skeleton points as seed pixels and executing Algorithm 1 for each skeleton separately. For a small dilation radius ($r = 10$), the histogram of the root map gives the influence areas of each skeleton point. The salience points of the skeletons are those with influence area greater than the area threshold obtained by setting $\theta = 70$ in Equation 1.



(a) (b) (c)

Figure 3: (a) Salience points of the contour of a fish and (b-c) salience points of its internal and external skeletons.

3.3 Contour saliences via skeletons

The relation between the contour and its internal and external skeletons [26] is directly obtained by applying Algorithm 1 to the contour [11, 12]. Equation 4 assigns to each pixel inside and outside the contour the maximum length of the shortest contour segment between two roots equidistant to that pixel according to the cost map. Figure 4a illustrates this situation for a salience point c of the skeleton, which is related to a salience point a of the contour. The difference value $D(c)$ is the length of the segment $\overline{dab}$. Suppose b is the root pixel of c, point a can be reached from point c by skipping $\overline{dab}/2$ pixels in the anti-clockwise orientation along the contour, starting from b. Similarly, point a could be found from c through d following the clockwise orientation, when d is the root pixel of c. The method only needs to determine which is the root pixel, either b or d. If the contour pixels are labeled in clockwise orientation, the root pixel of c will be b whenever $\delta(p,q) > N - \delta(p,q)$ in Equation 4 for $L(q) = L(d)$ and $L(p) = L(b)$. Otherwise, the root pixel of c will be d for $L(q) = L(b)$ and $L(p) = L(d)$. The same rule is applied for the external skeleton. Figures 4b-c illustrate the same concept applied to a real shape.

The correct orientation (clockwise or anti-clockwise) can be encoded in the difference image D by signaling it. Equation 4 must be substituted by the following algorithm applied to all pixels p in image D:

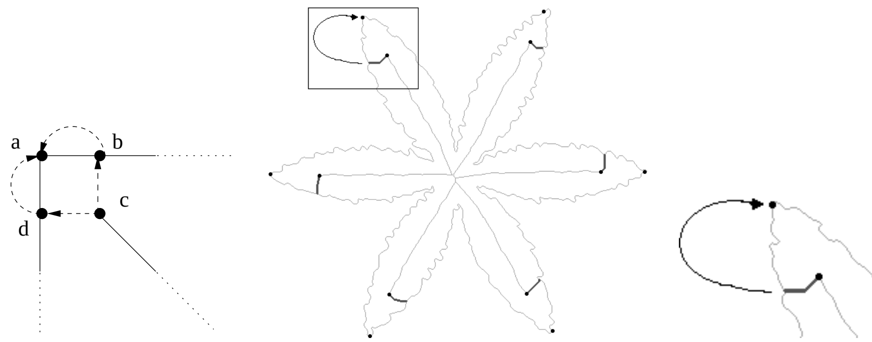
Algorithm 2:
Input: A root label map L.
Output: A signed difference image D.

1. For every pixel p of the image D, do

   1.1. Set $\delta_{\max} \leftarrow -\infty$.

   1.2. For each pixel $q \in A_4(p)$, do

   1.2.1. Set $\Delta \leftarrow \min\{\delta(p,q), N - \delta(p,q)\}$ and $s \leftarrow 1$.



(a) (b) (c)

Figure 4: (a) Relation between skeleton and contour saliences. (b) The same concept applied to a contour. (c) A zoomed region of the figure in (b).

1.2.2. If $\Delta = N - \delta(p, q)$, then

1.2.2.1. Set $s \leftarrow -1$.

1.2.3. If $\Delta > \delta_{\max}$, then

1.2.3.1. Set $\delta_{\max} \leftarrow \Delta$ and $sign \leftarrow s$.

1.3. Set $D(p) \leftarrow sign \times \delta_{\max}$.

The pixels of D with absolute values greater than 5% of N are chosen to represent the internal and external skeletons. The salience points of the skeletons can be obtained by the area thresholding method described in Section 2. Finally, the signaled values of the skeleton salience points in D and their roots on the contour are used to locate the corresponding contour salience points, as illustrated in Figure 4.

Although the method works fine for convex contour points, it adds non-relevant concave points, because the external skeleton may present spurious branches due to contour rotation and scaling. Unfortunately, these non-relevant concave saliences reduce the performance of the contour salience descriptor [11, 12]. Also, if the threshold of 5% is increased to eliminate the spurious branches of the external skeleton, the method misses relevant concave points of the contour. In this paper, the spurious branches are eliminated by an alternative skeleton labeling process and the problem is solved as follows.

The branches of the external skeleton are labeled with both, the label of their related root pixel on the contour and the length of the branch. The length-labeled skeleton image is thresholded and the resulting binary image is multiplied by the root-labeled skeleton image. These last steps remove concave contour saliences related to small branches and preserve the relevant concave saliences.

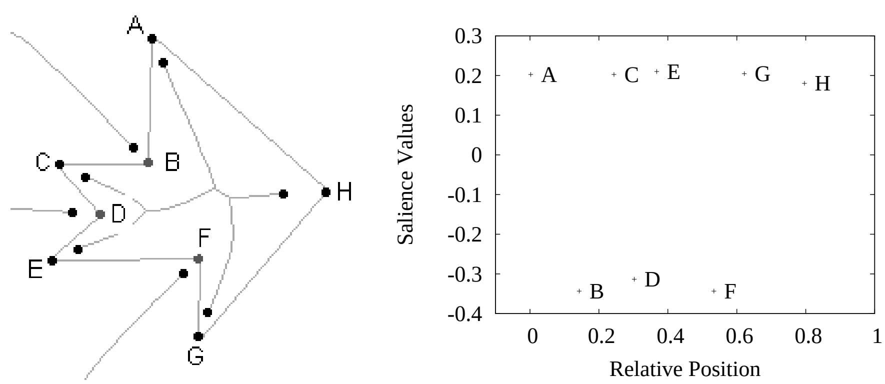
4 Contour Salience Descriptors

Although the salience values along the contour can not be used to locate salience points in the case of intricate and complex contours, they encode important local and global information about the contour which can be exploited to create effective shape descriptors.

An example is the descriptor based on the convex contour saliences presented in [11, 12]. Since, the problem of estimating concave points is solved now, this paper proposes the same contour salience descriptor including the concave points (Section 4.1) and a new shape salience descriptor for contour segments (Section 4.2).

4.1 Contour Saliences (CS)

After determining the salience points along the contour (Section 3), concave points have their salience values signed negative and the salience values of convex points remain positive. One arbitrary salience point on the contour is taken as reference and the method computes the relative position of each salience point with respect to the reference point. Thus, the signed salience values and the relative position of the points form two feature vectors of the same size, which are used in the contour salience descriptor. Figure 5 illustrates these feature vectors for a polygon. The contour of the polygon, its reference point, the internal and external skeletons, and the respective salience points are indicated in Figure 5a. The plot shown in Figure 5b indicates the salience values versus the relative position of the points along the contour.



(a) (b)

Figure 5: (a) Contour and skeletons of a polygon, where salience points are indicated by dots. (b) The salience values of the polygon in (a).

Whenever two contours of the same object appear in different positions (e.g., rotations and scales), they should be represented by the same salience points. However, the point taken as reference may not be the same in both. Also, the feature vectors of distinct objects may have different sizes. Therefore, the contour salience descriptor uses a heuristic matching algorithm between contours which registers their feature vectors using the reference points and computes their similarity taking into account their difference in size. This matching algorithm is based on the algorithm proposed by Abbasi and Mokhtarian [1, 31] to match Curvature Scale Space (CSS) images, and it is described in [11, 12].

4.2 Segment Saliences (SS)

The segment salience descriptor is a variation of the contour salience descriptor which incorporates two improvements: the *salience values* of contour segments, in the place of salience values of isolated points, and another matching algorithm that replaces the heuristic matching by an optimum approach.

The salience values along the contour are computed as described in Section 2.1 and the contour is divided into a predefined number s of segments of the same size. The internal and external influence areas of each segment are computed by summing up the influence areas of its corresponding pixels. A contour segment is considered *convex*, when its accumulated external area is greater than its accumulated internal area, and it is *concave* otherwise. The *difference* between them is defined as the salience value of the contour segment, which is positive when it is convex, and negative when it is concave. These signed salience values form the feature vector of the segment salience descriptor. Algorithm 3 below presents the procedures to compute this feature vector for a given contour.

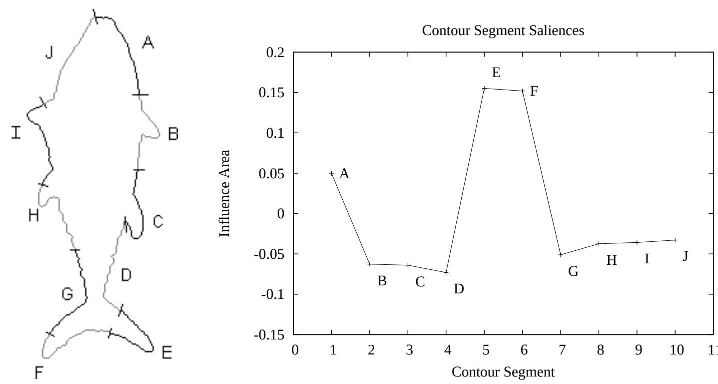
Algorithm 3:
Input: A contour ζ in an image I; number s of segments.
Output: A feature vector SS encoding the contour segment saliences.

1. Apply Algorithm 1 using the pixels in ζ as seeds and create a label map L as described in Section 3.1.
2. For each $t \in \zeta$, compute its internal ($H_{int}(t)$) and external ($H_{ext}(t)$) influence areas.
3. Split the contour ζ into a set $S = \{Seg_1, Seg_2, ..., Seg_s\}$ with s segments of the same size.
4. For each segment in S, compute its internal ($A_{int}(Seg_i)$) and external ($A_{ext}(Seg_i)$) influence areas as follows:

   4.1. $A_{int}(Seg_i) = \sum_{t \in Seg_i} H_{int}(t)$
   4.2. $A_{ext}(Seg_i) = \sum_{t \in Seg_i} H_{ext}(t)$
5. Compute the feature vector SS of size s as:

   5.1. $SS(i) = A_{ext}(Seg_i) - A_{int}(Seg_i)$, for $1 \leq i \leq s$

Figure 6 illustrates this feature vector for a contour, which is divided into 10 segments (Figure 6a). The curve shown in Figure 6b indicates the salience value of each segment along the contour.



(a) (b)

Figure 6: (a) A contour with 10 segments. (b) The salience values of the segments.

The fixed number of segments per contour allows the use of the optimal correspondent subsequence (OCS) algorithm [36] to match feature vectors between contours. This matching algorithm is the same used in the Beam Angle Statistics (BAS) descriptor [3]. Feature vectors of the same size also simplify the storage and access methods of the image database.

5 Evaluation

The evaluation process consists of defining a shape database, an effectiveness measure and a set of shape descriptors for comparison.

5.1 Shape database

The shape database is a set with one thousand and one hundred fish contours obtained from [33]. Since there is no semantic definition of relevant images (classes of contours) for this database, each group of relevant images is defined as one fish contour and 9 different manifestations of rotation and scaling applied to it. Therefore, the problem consists of 1100 classes with 10 shapes each.

5.2 Effectiveness measure

The experiments adopted the *query-by-example (QBE)* [4] paradigm. In the CBIR context, an image is given as an input and two types of searches are possible: *similarity range* and *similarity rank*. The search by similarity range returns the images of the database whose distance from the query image is less than a given search radius. The search by similarity rank returns a specified number of images in the increasing order of distance with respect to the query image. In both cases, the effectiveness of the system is related to the relevance of the retrieved images. It is expected that the relevant images return before non-relevant images in the second case and the non-relevant images do not return in the first case. In

Classes	Descriptor 1
class 1	$\{(1.50, 2.50), (1.50, 2.00), (2.00, 2.00), (1.00, 2.00), (1.50, 1.50)\}$
class 2	$\{(1.00, 1.00), (1.00, 2.00), (1.00, 3.00), (1.00, 4.00), (1.00, 5.00)\}$
Classes	Descriptor 2
class 1	$\{(2.00, 1.00), (2.00, 2.00), (2.00, 3.00), (2.00, 4.00), (2.00, 5.00)\}$
class 2	$\{(1.40, 1.40), (1.60, 1.40), (1.60, 1.20), (1.40, 1.20), (1.50, 1.30)\}$
Classes	Descriptor 3
class 1	$\{(1.50, 2.50), (1.50, 2.00), (1.75, 2.25), (1.25, 2.00), (1.50, 1.50)\}$
class 2	$\{(1.50, 5.50), (1.25, 5.00), (1.50, 5.00), (1.15, 5.00), (1.50, 4.50)\}$

Table 1: Coordinates of each image in classes 1 and 2 for the three hypothetical descriptors.

some applications, the relevance of the retrieved images depends on the user's opinion. However, there are several other applications where predefined classes determine groups of relevant images independent of user. Any query image in a given class should return the images of the database belonging to this class first. In such a case, it makes sense to compare descriptors based on objective measures.

The experiments of this paper evaluate the ability of shape descriptors to distinguish between different fish contours and to identify a fish contour independent of possible rotation and scaling transformations. Note that the effectiveness of the shape descriptors apply for image retrieval and image analysis, considering the resemblance between both problems. Since each shape descriptor represents a contour as a "point" in the corresponding metric space, its effectiveness will be higher as more separate the clusters of relevant contours are in the metric space; and as more compact the clusters are in the metric space, higher will be the robustness of the shape descriptor with respect to an increase in the number of classes. Therefore, a "good" effectiveness measure should capture the concept of *separability*, and perhaps the concept of *compact-ability* for sake of robustness. More formally, the compact-ability of a descriptor indicates its invariance to the object characteristics that belong to a same class, while the separability indicates its discriminatory ability between objects that belong to distinct classes. While these concepts are commonly used to define validity measures in cluster analysis [13, 17], they seem to not have caught much attention in the literature of CBIR systems, where one of the most used effectiveness measures is *Precision* $\times$ *Recall* [32].

A simple example can be used to illustrate that *Precision* $\times$ *Recall* does not capture the separability and compact-ability concepts, and therefore, it should not be used as effectiveness measure. Consider the existence of two classes (*class 1* and *class 2*) composed by 5 images each and three different image descriptors (*descriptor 1*, *descriptor 2*, and *descriptor 3*), whose extraction algorithms create feature vectors belonging to $\mathbb{R}^2$ space. Table 1 shows the coordinates of each image in each class for these three hypothetical descriptors.

Figures 7, 8, and 9 show the classes 1 and 2 in the Cartesian plane for descriptors 1, 2 and 3, respectively.

Note that, it is reasonable to expect that the descriptor 3 will be more effective than the descriptor 2, which will be more effective than the descriptor 1. However, Figure 10 shows the *Precision* $\times$ *Recall* graph for these descriptors, and even though descriptor 3 presents

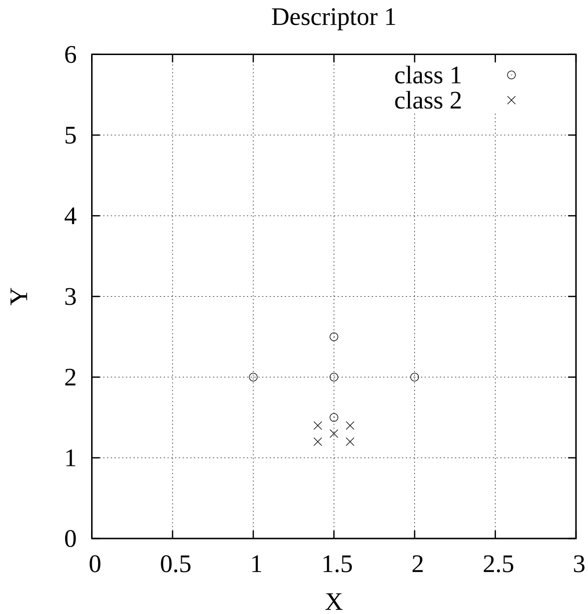


Figure 7: Descriptor 1

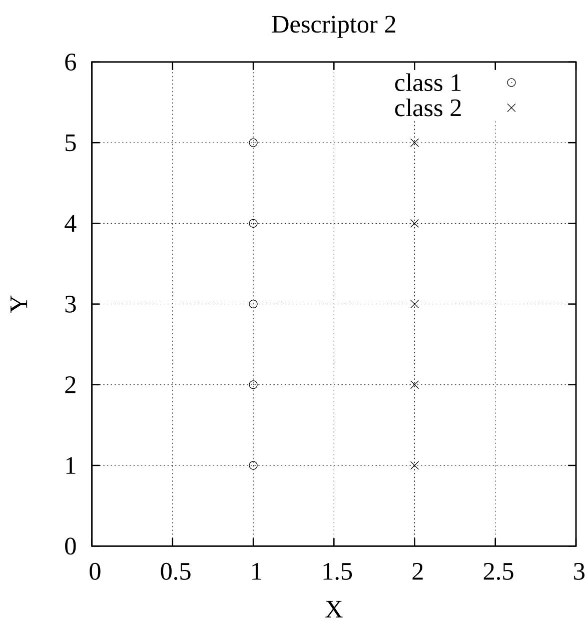


Figure 8: Descriptor 2

the best $Precision \times Recall$ curve, descriptor 1 outperforms descriptor 2.

On the other hand, the concepts of separability and compact-ability seem to be better

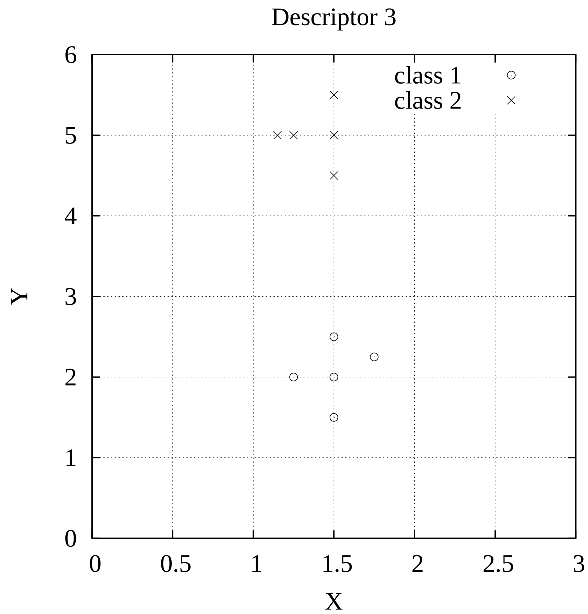


Figure 9: Descriptor 3.

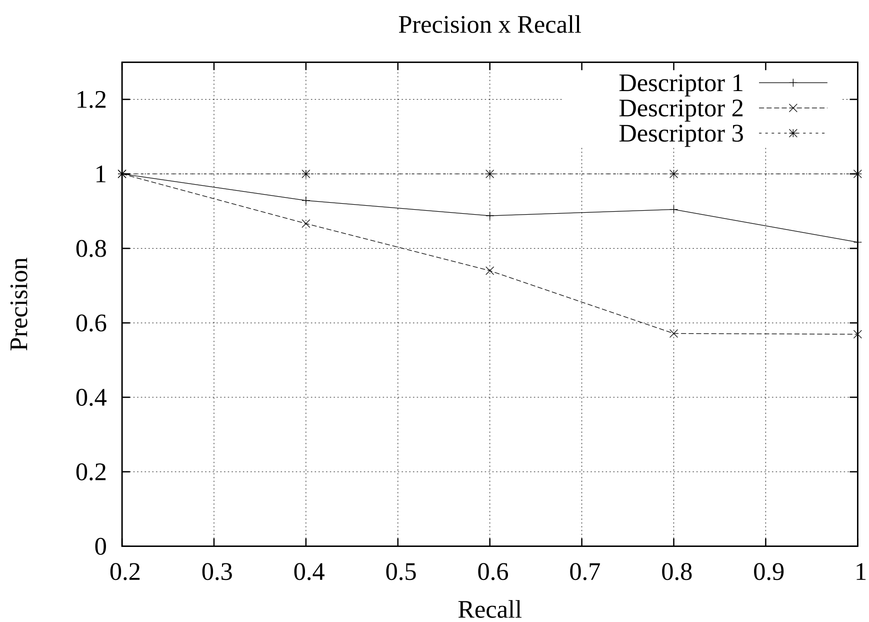


Figure 10: Precision vs Recall: as higher is the curve, as better is the descriptor.

represented by the measures proposed in [11]. Figure 11 shows, for example, the *multiscale separability* curves for the three descriptors. Note that, descriptor 3 presents the best curve again. However, curves of descriptors 1 and 2 have the opposite behavior when compared to the *Precision* $\times$ *Recall* graph. Now, descriptor 2 is more effective than descriptor 1, as expected.

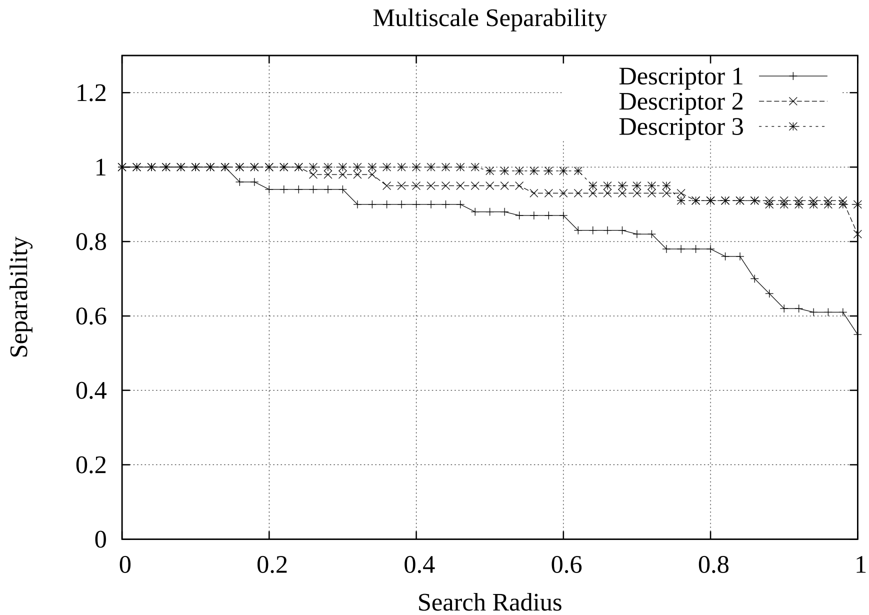


Figure 11: Multiscale separability: as higher is the curve, as better is the descriptor

Due to these observations, the present paper uses the concepts of compact-ability and multiscale separability proposed in [11] to evaluate the shape descriptors. The *Segment Saliences (SS)* implementation considered in this experiment used 30 segments.

5.3 Evaluated descriptors

The proposed shape descriptors, contour saliences (CS) and segment saliences (SS), are compared with the following shape descriptors.

Curvature Scale Space (CSS) [1, 31]:. The CSS descriptor is used in the MPEG-7 standard and represents a multiscale organization of the curvature zero-crossing points of a planar curve. In this sense, the dimension of its feature vectors varies for different contours, thus a special matching algorithm is necessary to compare two CSS descriptors (e.g., [11]). The implementation of the CSS descriptor is a C version of the Matlab prototype presented in [30].

Beam Angle Statistics (BAS) [2, 3]: The BAS descriptor has been compared with several others [5, 6, 23, 25, 27, 31], including the CSS descriptor. In [3], it was shown that the BAS functions with 40 and 60 samples outperform all of them. The experiments of the present paper use the BAS descriptor with 60 samples. Basically, the BAS descriptor is based on the *beams* originated from a contour pixel. A beam is defined as the set of lines connecting a contour pixel to the rest of the pixels along the contour. At each contour pixel,

Descriptor Id	Descriptor Name
SS	Segment Saliences
CS	Contour Saliences
CCS	Convex Contour Saliences
CSS	Curvature Scale Space
BAS	Beam Angle Statistics

Table 2: List of evaluated descriptors.

Descriptor Id	Compact-ability
SS	0.93
CS	0.73
CCS	0.70
CSS	0.73
BAS	0.95

Table 3: Compact-ability values of the evaluated descriptors.

the angle between a pair of lines is calculated, and the shape descriptor is defined by using the third-order statistics of all the beam angles in a set of neighborhoods. The similarity between two BAS moment functions is measured by an optimal correspondent subsequence (OCS) algorithm, as shown in [3].

Convex Contour Saliences (CCS) [11, 12]: The CCS is the same descriptor described in Section 4.1, without the concave saliences. The CCS has outperformed Multiscale Fractal Dimension [11], Fourier Descriptors [21, 29], Moment Invariants [16, 22], CSS [1, 31] and BAS [3] with respect to the multiscale separability measure [11]. Experiments with Precision $\times$ Recall have also showed better results with the CCS as compared to CSS, Fourier Descriptors, and Moment Invariants [12]. Since the fish database is the same used in these experiments, only BAS and CSS were maintained for comparison.

Table 2 summarizes the set of evaluated shape descriptors.

5.4 Experimental results

Figure 12 shows the separability curves of the evaluated descriptors. Observe that the *Contour Saliences* (CS) presents a better separability curve than the *Convex Contour Saliences* (CCS) for search radii less than 80% of their maximum distance. This indicates that the CS descriptor encodes more information (due to the concave points) than the CCS. The most relevant result is certainly the best separability curve of the *Segment Saliences* (SS) for almost all search radii.

Table 3 presents the compact-ability values of the evaluated shape descriptors. The higher values were found for Beam Angle Statistic (BAS) and SS, while CCS presented the lowest value. According to these experiments, the SS descriptor is more effective than the others, since it provides the best separability, and the second most robust (due to its compact-ability). This is certainly a very relevant result.

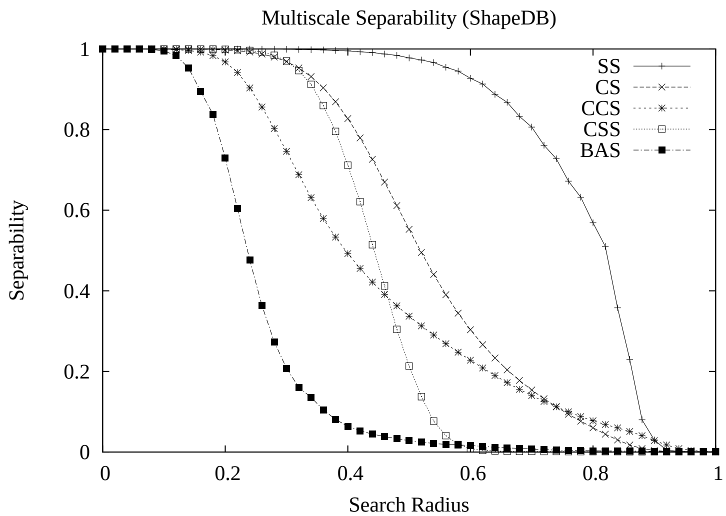


Figure 12: Multiscale Separability curves.

6 Conclusion

This paper has presented a more robust approach to incorporate concave saliences into the contour salience descriptor and a new shape descriptor based on salience values of contour segments. They both make use of the image foresting transform as a general tool for the design of image processing operators. The results indicate segment saliences as the most effective descriptor among contour saliences, convex contour saliences [11, 12], curvature scale space [1, 31], and beam angle statistics [2, 3], using a fish database with 11,000 images organized in 1,100 classes. They also confirm the improvement of incorporating concave saliences into the contour salience descriptor. It is important to notice that the segment salience descriptor does not require the location of salient points along the contour. In this sense, it is much simpler than the contour salience descriptor, which together with its high compact-ability make the results even more relevant.

The effectiveness in image retrieval was discussed with respect to the Precision×Recall measure and the multiscale separability [11] was proposed as a more appropriate effectiveness measure.

Ongoing developments consider the creation of shape descriptors, which combine the salience features with color- and texture-based descriptors, and applications in CBIR that use the proposed shape descriptors as effective indexing vectors.

Acknowledgements

This work was supported by FAPESP (Proc. 01/02788-7), CAPES (Proc. BEX844/03-9), and CNPq (Proc. 302966/02-1).

The authors are grateful to Sadegh Abbasi, Farzin Mokhtarian, and Josef Kittler for the fish database, and to Nafiz Arica and Fatos Vural for the BAS source code.

References

[1] S. Abbasi, F. Mokhtarian, and J. Kittler. Enhancing CSS-based Shape Retrieval for Objects with Shallow Concavities. *Image and Vision Computing*, 18(3):199–211, February 2000.

[2] N. Arica and F. T. Y. Vural. A Perceptual Shape Descriptor. In *Proceedings of the International Conference on Pattern Recognition*, pages 375–378, Madison, Wisconsin, USA, 2002.

[3] N. Arica and F. T. Y. Vural. BAS: A Perceptual Shape Descriptor Based on the Beam Angle Statistics. *Pattern Recognition Letters*, 24(9-10):1627–1639, June 2003.

[4] Y. A. Aslandogan and C. T. Yu. Techniques and Systems for Image and Video Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):56–63, January/February 1999.

[5] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24):509–522, 2002.

[6] G. Chuang and C.-C. Kuo. Wavelet Descriptor of Planar Curves: Theory and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(1):56–70, 1996.

[7] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In *Proceedings of 23rd International Conference on Very Large Data Bases*, pages 426–435, Athens, Greece, 1997.

[8] L. Costa, A. Campos, and E. Manoel. An Integrated Approach to Shape Analysis: Results and Perspectives. In *Proceedings of the International Conference on Quality Control by Artificial Vision*, pages 23–34, Le Creusot, France, May 2001.

[9] L. da F. Costa and L. F. Estrozi. Multiresolution Shape Representation without Border Shifting. *Electronic Letters*, 35(21):1829–1830, 1999.

[10] R. da S. Torres, A. X. Falcão, and L. da F. Costa. Shape Description by Image Foresting Transform. In *Proceedings of the 14th International Conference on Digital Signal Processing*, volume 2, pages 1089–1092, Santorini, Greece, July 2002.

[11] R. da S. Torres, A. X. Falcão, and L. da F. Costa. A Graph-based Approach for Multiscale Shape Analysis. *Pattern Recognition*, 37(6):1163–1174, June 2004.

[12] R. da S. Torres, E. M. Picado, A. X. Falcão, and L. da F. Costa. Effective Image Retrieval by Shape Saliences. In *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing*, pages 49–55, São Carlos, São Paulo, Brazil, 12-15 October 2003.

[13] D. Davies and D. Bouldin. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.

[14] E. Dijkstra. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1:269–271, 1959.

[15] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.

[16] S. A. Dudani, K. J. Breeding, and R. B. McGhee. Aircraft Identification by Moment Invariants. *IEEE Transactions on Computers*, 26(1):39–45, January 1977.

[17] J. Dunn. Well Separated Clusters and Optimal Fuzzy Partitions. *Cybernetics*, 4:95–104, 1974.

[18] A. Falcão, B. S. da Cunha, and R. A. Lotufo. Design of Connected Operators Using the Image Foresting Transform. In M. Sonka and K. Hanson, editors, *Proceedings of SPIE on Medical Imaging*, volume 4322, pages 468–479, San Diego, CA, Feb 2001.

[19] A. Falcão, L. da F. Costa, and B. da Cunha. Multiscale Skeletons by Image Foresting Transform and its Applications to Neuromorphometry. *Pattern Recognition*, 35(7):1571–1582, April 2002.

[20] A. X. Falcão, J. Stolfi, and R. A. Lotufo. The Image Foresting Transform: Theory, Algorithms, and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):19–29, Jan 2004.

[21] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, Reading, MA, USA, 1992.

[22] M. K. Hu. Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.

[23] A. Khotanzan and Y. H. Hong. Invariant Image Recognition by Zernike Moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–487, 1990.

[24] R. Kimmel, D. Shaked, N. Kiryati, and A. M. Bruckstein. Skeletonization via Distance Maps and Level Sets. *Computer Vision and Image Understanding*, 62(3):382–391, 1995.

[25] L. J. Latecki and R. Lakamper. Shape Similarity Measure Based on Correspondence of Visual Parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1185–1190, 2000.

[26] M. Leyton. Symmetry-Curvature Duality. *Computer Vision, Graphics, and Image Processing*, 38(3):327–341, 1987.

[27] L.-J. Lin and S.-Y. Kung. Coding and Comparison of DAGs as a Novel Neural Structure with Applications to On-Line Handwriting Recognition. *IEEE Transactions on Signal Processing*, 45(11):2701–2708, 1997.

[28] R. Lotufo and A. Falcão. The Ordered Queue and the Optimality of the Watershed Approaches. In J. Goutsias, L. Vincent, and D. S. Bloomberg, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 341–350, Palo Alto, USA, Jun 2000. Kluwer Academic.

[29] B. M. Mehtre, M. S. Kankanhalli, and W. F. Lee. Shape Measures for Content Based Image Retrieval: A Comparison. *Information Processing and Management*, 33(3):319–337, 1997.

[30] C. Y. Ming. Shape-Based Image Retrieval in Iconic Image Databases. Master's thesis, Chinese University of Hong Kong, June 1999.

[31] F. Mokhtarian and S. Abbasi. Shape Similarity Retrieval Under Affine Transforms. *Pattern Recognition*, 35(1):31–41, January 2002.

[32] H. Muller, W. Muller, D. M. Squire, S. M. Maillet, and T. Pun. Performance Evaluation in Content-Based Image Retrieval: Overview and Proposals. *Pattern Recognition Letters*, 22:593–601, 2001.

[33] ShapeDB. www.ee.surrey.ac.uk/research/vssp/imagedb/demo.html, October 2004.

[34] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-Based Image Retrieval at the End of the Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, December 2000.

[35] C. Traina, B. Seeger, C. Faloutsos, and A. Traina. Fast Indexing and Visualization of Metric Datasets Using Slim-Trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(2):244–60, March/April 2002.

[36] Y. P. Wang and T. Pavlids. Optimal Correspondence of String Subsequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(11):1080–1087, 1990.