



Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Shape feature extraction and description based on tensor scale

F.A. Andaló, P.A.V. Miranda, R. da S. Torres*, A.X. Falcão

Institute of Computing, University of Campinas (Unicamp), 1251 CEP 13084-851, Campinas, SP, Brazil

ARTICLE INFO

Article history:

Received 12 April 2008

Received in revised form 16 June 2009

Accepted 23 June 2009

Keywords:

Shape analysis

Image processing

Tensor scale

Image Foresting Transform

Shape description

Shape saliencies

Content-based image retrieval

ABSTRACT

Tensor scale is a morphometric parameter that unifies the representation of local structure thickness, orientation, and anisotropy, which can be used in several computer vision and image processing tasks. In this article, we exploit this concept for binary images and propose a shape salience detector and a shape descriptor—*Tensor Scale Descriptor with Influence Zones*. It also introduces a robust method to compute tensor scale, using a graph-based approach—the Image Foresting Transform. Experimental results are provided, showing the effectiveness of the proposed methods, when compared to other relevant methods, such as Beam Angle Statistics and Contour Saliency Descriptor, with regard to their use in content-based image retrieval tasks.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

The shape of an object is an important and basic visual feature for describing image content [1,2], and can be thought of as a silhouette of the object [1], invariant to rotation, scale and translation [3]. Shapes are often the archetypes of objects belonging to the same pattern class [4], and can be used in a wide range of practical problems, such as document analysis (optical character recognition), visual arts (video restoration), internet (content-based image retrieval), security (fingerprint detection), etc. [4]. In content-based image retrieval (CBIR) systems, for example, shape descriptors can be used to encode image properties that are relevant to a query. For image registration tasks, shape saliencies can be used to identify correspondences between objects' contours.

The work presented in this article aims at shape feature extraction and description and, in this scenario, it is important to have a parameter for characterizing the structures presented in the images. In [5], Saha introduced a new concept, called *tensor scale*—a local morphometric parameter that yields a unified representation of structure thickness, orientation, and anisotropy. That is, at any image point, its tensor scale is represented by the largest ellipse centered at that point and within the same homogeneous region.

Scale may be thought of as a range of spatial resolutions needed to ensure a sufficient yet compact object representation. It is

important in determining the optimum trade-off between noise smoothing and perception/detection of structures [5]. The notion of local scale stemmed from the idea of identifying the optimal scale at each image point and, before the proposal of tensor scale, the approaches inferred only statistical information of local frequency and orientation. Tensor scale can be used as a measure of local morphometric scale because it actually yields a unified morphometric representation of structures. The tensor scale approach has been successfully applied in a variety of image processing tasks, such as segmentation [6], registration [7], diffuse filtering [5], shape description [8]; and also used as a regional parameter to analyze local structure morphology at low resolution [9,10].

In this article, we extend the applications of tensor scale for shape feature extraction and description. The proposed methods are a shape salience detector and a shape descriptor—*Tensor Scale Descriptor with Influence Zones (TSDIZ)*. The salience detector is responsible for detecting the higher curvature points along the shape contour [11] (vertex points along the contour with first derivative discontinuity [12]). TSDIZ descriptor is characterized by two functions: a feature vector extraction function and a similarity function. The feature vector represents the shape properties extracted from a binary image and the similarity function computes the similarity between images based on their feature vectors [13].

The key idea in the proposed methods is to compute the tensor scale ellipse for every object point and to map their orientations onto the object's contour. In binary images, the tensor scale ellipses closer to the object's border (often with higher anisotropy) usually better describe the object's shape. Therefore, we exploit these ellipses with higher anisotropy, mapping their orientation onto the object's contour to also capture space information. In the first method, the

* Corresponding author. Tel.: +55 19 3521 5887; fax: +55 19 3521 5847.

E-mail addresses: feandalo@ic.unicamp.br (F.A. Andaló),paulo_miranda@ic.unicamp.br (P.A.V. Miranda), rtorres@ic.unicamp.br(R. da S. Torres), afalcao@ic.unicamp.br (A.X. Falcão).

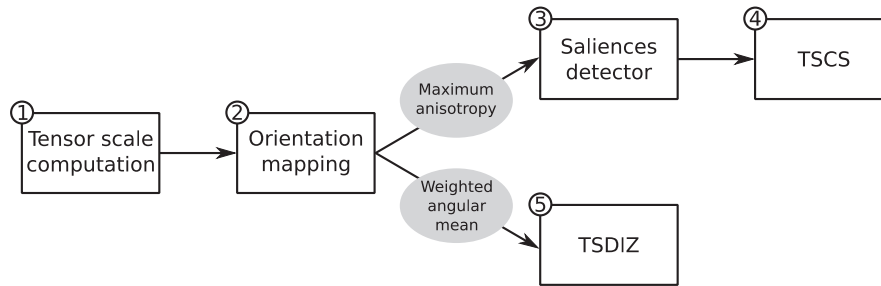


Fig. 1. Computation steps for shape salience detection and TSDIZ shape description.

mapped orientations are used for salience point detection. In the second method, the mapped orientations are used for shape description, forming the TSDIZ descriptor.

Although the tensor scale orientation mapping is different for each proposed method, they both exploit the discrete Voronoi regions (influence zones) of contour points inside the object. The influence zone of a contour point is defined by the object pixels that are closer to that point than to any other on the contour. The discrete Voronoi regions can be efficiently obtained by label propagation using the Euclidean Image Foresting Transform (Euclidean IFT) [14].

The salience detector computes the influence zone of the contour points and assigns, to each contour point, the orientation of the ellipse with maximum anisotropy inside its influence zone. The TSDIZ method first divides the contour into a predefined number of segments. Then, it computes the influence zone of these segments and assigns, to each segment, the weighted mean orientation [15] of the ellipses within its influence zone.

For the salience detector experiments, we constructed the Tensor Scale Contour Saliency (TSCS) descriptor, which is the Contour Saliency (CS) descriptor [16] with the salience points detected by the tensor-scale approach presented in this article. We have used the TSCS to validate our salience detector, comparing its results with CS; but its application for CBIR is not as competitive as TSDIZ.

Fig. 1 shows a diagram with the necessary steps for the methods. In addition, this article also presents a much faster algorithm for tensor scale computation, as compared to previous methods [5,8], by exploiting the Euclidean IFT.

The experiments are divided into two parts. The first part evaluates the quality of the estimated salience points by analyzing their impact in the Contour Saliency (CS) descriptor [16]. This experiment shows that the proposed method is more effective and faster than CS method for salience detection. The second part consists of comparing TSDIZ to other shape descriptors, with respect to two effectiveness measures used in CBIR: precision vs. recall [17] and multiscale separability [16]. This experiment shows that TSDIZ effectiveness is as good as or higher than recently proposed descriptors, including Beam Angle Statistics [18], Tensor Scale Descriptor [8], and Segment Saliencies [16].

The outline of this article follows the flowchart in Fig. 1. The article starts by presenting the tensor scale concept and its computation in Section 2. Section 2 also presents the Euclidean IFT, that is used in two steps: tensor scale computation and orientation mapping. The orientation mapping itself is described in Section 3. The salience point detector is presented in Section 4 and the TSDIZ shape descriptor is presented in Section 5. The experiments and results of this work are provided in Section 6. Finally, Section 7 states the conclusions and the directions for future work.

2. Tensor scale

In [5], Saha introduced the *tensor scale* of a pixel p in a gray-scale image as the largest ellipse within the same homogeneous

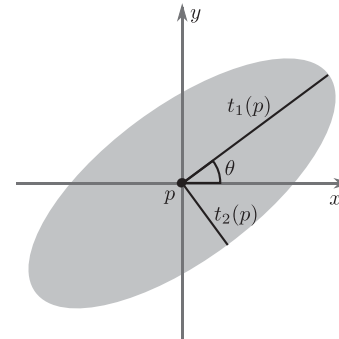


Fig. 2. Tensor scale factors.

region, centered at p . The homogeneous region is defined based on a predefined criterion and, for binary images, it is naturally defined by the object pixels.

The ellipse of the tensor scale is defined by three factors (see Fig. 2):

- $Orientation(p) = \text{angle between } t_1(p) \text{ and the horizontal axis;}$
- $Anisotropy(p) = \sqrt{1 - |t_2(p)|^2 / |t_1(p)|^2};$
- $Thickness(p) = |t_2(p)|;$

where $|t_1(p)|$ and $|t_2(p)|$, with $|t_1(p)| \geq |t_2(p)|$, denote the length of the two semi-axes of the ellipse centered at p . Fig. 2 illustrates the components to compute each one of these factors.

In the following subsection, we describe two previously proposed tensor scale computation methods for gray-scale images: the algorithms by Saha [5] and by Miranda et al. [8].

2.1. Tensor scale for gray-scale images: related works

In Saha's approach [5], a tensor scale ellipse is calculated from pairs of radially opposite sample lines, that are traced emerging from the center pixel (Fig. 3(a)). The axes of the ellipse are determined by first computing a pixel intensity profile on each sample line, then extracting two intensity connections from each profile, and finally applying a conventional edge detector to localize the edge points (end of the homogeneous region) on the sample lines (Fig. 3(b)). The next step consists of repositioning the edge points equidistant to the center pixel, following the axial symmetry of the ellipse (Fig. 3(c)). The computation of the best-fit ellipse to the repositioned edge points is done by Principal Component Analysis (PCA) (Fig. 3(d)).

These computations are performed for every pixel of the image. A critical drawback of Saha's approach is that the computational cost of the algorithm makes his method quite prohibitive for more complex tasks, such as image description in content-based image retrieval (CBIR) systems. For this reason, Miranda et al. [8] proposed

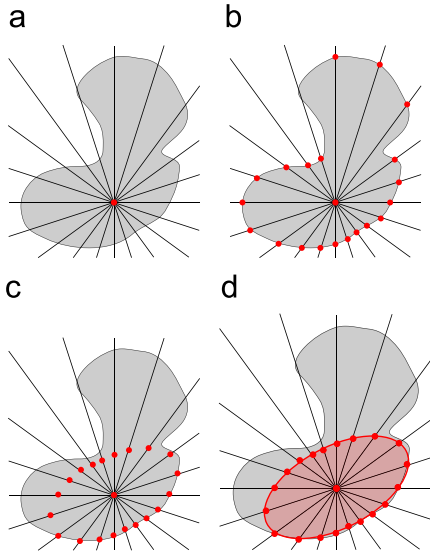


Fig. 3. Tensor scale computation.

an efficient implementation of the original method, which differs in the following aspects.

The first change was in the edge location phase. Miranda et al. propose to go along each pair of opposite sample lines, alternately and at the same time, instead of going along one entire line by turn. By doing this, when an edge point is found on a sample line, the opposite edge point is already in the correct position, and therefore, the reposition phase is no longer necessary.

The second change was the use of two connected thresholds, th_1 and th_2 , to improve and simplify the original method of detecting edge points. The first one detects abrupt discontinuities by setting the maximum accepted absolute intensity difference along sample lines, defining homogeneous regions in the image. The second is the maximum cumulative absolute intensity difference allowed inside a homogeneous region, which detects smooth intensity transitions outward the homogeneous region. Fig. 4 illustrates the use of th_1 and th_2 . The object O is a homogeneous region in an image, and P is a pixel in O . The pixels A and B , over the sample line, are the edge points that we want to find using th_1 and th_2 .

The third and final change was the improvement in the ellipse computation phase. Miranda et al. proposed a function g (Eq. (1)) that gives the angle of the ellipse directly, instead of using PCA. The ellipse orientation is obtained from the value of γ that minimizes the function g below:

$$g(\gamma) = \sum_{i=1,2,\dots,m} [x_{i\gamma}^2 - y_{i\gamma}^2], \quad (1)$$

where m is the number of sample lines, $x_{i\gamma} = x_i \cos(\gamma) - y_i \sin(\gamma)$, $y_{i\gamma} = x_i \sin(\gamma) + y_i \cos(\gamma)$, (x_i, y_i) are the relative coordinates of the edge points with respect to the center pixel $p = (x_p, y_p)$ of the ellipse, and $(x_{i\gamma}, y_{i\gamma})$ are the new coordinates after applying a rotation by the angle γ .

Considering these optimizations, Miranda et al. [8] proposed the Tensor Scale Descriptor (TSD) for gray-scale images. The idea of their shape descriptor stemmed from the observation that distinct objects often present different tensor scale local orientation distributions of their shapes (this is also valid for texture, i.e., their descriptor could be easily extended for colored images by changing the criterion to define homogeneous regions). The TSD computes the tensor scale parameters for the original image and then computes the local orientation histogram, used as feature vector. The matching between two given images by TSD is made by taking the absolute difference

of the area between their orientation histograms, after correcting their displacement by correlation (i.e., object rotations cause shifts in the histogram). For more information and for figures illustrating the method, refer to [8].

The TSD is used as baseline in the experiments with the TSDIZ. The idea of the TSDIZ descriptor was inspired on a deficiency of TSD (that is the same deficiency of every simply histogram-based method): the spatial information of the contour points is not considered in the histogram elements. But it is quite intuitive to perceive that the spatial information is relevant for describing shapes.

In the next subsection, we provide a faster tensor scale computation method, as compared to the described approaches [5,8] (when applied to binary images), by exploiting the Euclidean IFT.

2.2. Tensor scale for binary images

We have proposed some changes in the algorithm by Miranda et al., which are applied to binary images only.

The first change consists of eliminating the thresholds used for edge detection, given that homogeneous regions are defined by object pixels. Furthermore, the method can incorporate techniques to easily find the edges in the directions of the sample lines. Such techniques comprehend the use of the Euclidean Distance Transform, computed by the Image Foresting Transform, that is described in the next subsection.

2.2.1. Euclidean Distance Transform via Image Foresting Transform

The Image Foresting Transform (IFT) is a graph-based approach to the design of image processing operators based on connectivity [14], in which the images are represented by graphs—the pixels are considered as nodes and the arcs are defined by an adjacency relation between pixels. A path in the image graph is a sequence of distinct pixels $\pi = (p_1, p_2, \dots, p_n)$. The minimization of a path-cost function computes an optimum path forest in the graph, rooted at the minima of the path-cost function. In the forest, each root pixel defines an influence zone (optimum path tree) composed of the pixels that are “more closely connected” to that root than to any other.

The contour points may be used as seeds of a set S , which are forced to be roots (minima) of the forest. If we label each contour point s_i , $i = 1, \dots, n_c$, where n_c is the number of contour points, by subsequent integers $\lambda_1(s_i) = i$, the IFT can propagate the labels forming the discrete Voronoi regions (influence zones) of the contour points in a label map L . The optimum-path forest is represented by a predecessor map, which encodes the optimum paths from the root pixels on the contour. We are not using this information here, so the IFT algorithm can be simplified to output only three other attributes for each pixel p : the squared Euclidean distance $C(p)$ between p and its closest point s in the contour (forming an optimum cost map), its closest seed $R(p) = s$ (forming a root map), and the label $L(p) = L(s)$ (forming the label map).

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 and label maps R and L . This operator asks for an Euclidean adjacency relation A and a path-cost function f_{euc} defined for any path $\pi = (p_1, p_2, \dots, p_n)$ in the graph as

$$q \in A(p) \Rightarrow (x_q - x_p)^2 + (y_q - y_p)^2 \leq \rho^2,$$

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

where ρ is the adjacency radius and (x_p, y_p) are the (x, y) coordinates of a pixel p_i in the image.

Note also that contour segments (instead of points) may be used as seeds of a set S (minima of the forest). In this case, the labeling function assigns subsequent integer numbers to contour segments;

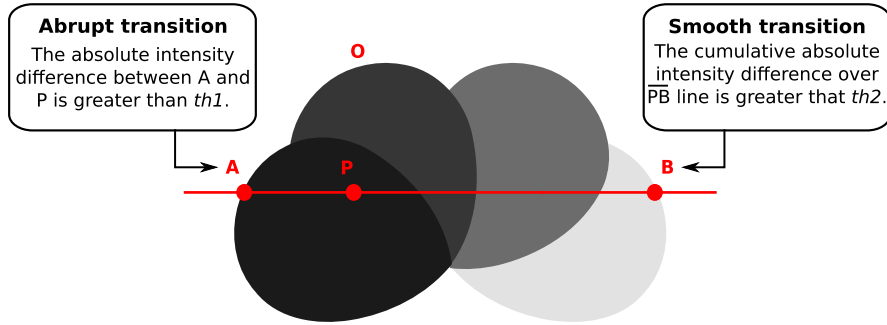


Fig. 4. Connected thresholds th_1 and th_2 .

i.e., to each contour segment s_j , $j=1, \dots, n_s$, where n_s is the number of contour segments, we have $\lambda_2(s_j)=j$. Therefore, the influence zones will be discrete Voronoi regions of those segments.

The algorithm below presents the IFT procedure with f_{euc} , i.e., the Euclidean IFT.

Algorithm 1.

Input: A binary image I , a set $S \subset I$ of seeds, an Euclidean adjacency relation A , and a labeling function λ .

Output: The cost map C , the root map R , and the label map L .

Auxiliary data structure: A priority queue Q .

```

foreach  $p \in I$  do
   $C(p) \leftarrow +\infty$ ;  $R(p) \leftarrow NIL$ ;  $L(p) \leftarrow NIL$ ;
foreach  $p \in S$  do
   $C(p) \leftarrow 0$ ;  $R(p) \leftarrow p$ ;  $L(p) \leftarrow \lambda(p)$ ;
  insert  $p$  in  $Q$ ;
while  $Q$  is not empty do
  remove from  $Q$  a pixel  $p = (x_p, y_p)$  such that  $C(p)$  is
  minimum;
  foreach  $q = (x_q, y_q)$  such that  $q \in A(p)$  and  $C(q) > C(p)$  do
     $C' \leftarrow (x_q - x_{R(p)})^2 + (y_q - y_{R(p)})^2$ , where  $R(p) = (x_{R(p)}, y_{R(p)})$ 
    is the root pixel of  $p$ ;
    if  $C' < C(q)$  then
      if  $C(q) \neq +\infty$  then
        remove  $q$  from  $Q$ ;
       $C(q) \leftarrow C'$ ;  $R(q) \leftarrow R(p)$ ;  $L(q) \leftarrow L(p)$ ;
      insert  $q$  in  $Q$ ;

```

The advantages of computing the Euclidean Distance Transform via IFT is that label propagation is executed on-the-fly and in linear time for small values of ρ (usually $\rho = \sqrt{2}$). The Euclidean IFT is used for two purposes in the proposed methods: faster tensor scale computation, that is described in the next subsection, and tensor scale orientation mapping (Section 3). Each of the proposed methods uses λ_1 or λ_2 as the labeling function for the Euclidean IFT.

2.2.2. Tensor scale computation via Image Foresting Transform

In this section, we describe a new tensor scale computation for binary images. The idea is to speed up the computation of the tensor scale by considering the maps obtained by the Euclidean IFT (Algorithm 1), using labeling function λ_1 . We show how to efficiently construct the set E of edge points (x_i, y_i) relative to the ellipse center over the sample line l_i :

$$E = \bigcup_{i=1,2,\dots,m/2} \{(x_i, y_i), (x_{i+m/2}, y_{i+m/2})\},$$

where m is the number of sample lines. By constructing E , we can easily obtain the orientation γ_{min} of the ellipse by considering every $(x_i, y_i) \in E$, $i = 1 \dots m$, in Eq. (1).

The proposed speed-up in the computation of the tensor scale for binary images is possible by exploiting the following aspect: if we have the shortest distance between the ellipse center pixel p and the contour, there is no need to search for edge points inside the circle with radius $\sqrt{C(p)}$ (Fig. 5(a)). For every pixel, the distance can be obtained from the cost map C returned by Euclidean IFT (Algorithm 1).

According to the algorithm by Miranda et al., edge points are searched along opposite sample lines, alternately. However, in our approach, the algorithm jumps along the lines and visits the opposite pixels q and r (the first pixels outside the circle with radius $\sqrt{C(p)}$ and along each sample line) at the same time (Fig. 5(b)).

The searching for edge points continues outside the area defined by the cost $C(p)$ in Fig. 5(b), and the minimum between $C(r)$ and $C(q)$ indicates the location for the next jump. These jumps may continue iteratively until the closest edge point along a sample line is found.

In the example, the edge over the sample line l_i is found at pixel $R(r)$ (i.e., at the contour point r' nearest to r). The algorithm defines that the two opposite edge points in the same direction are at $r' = (x_i, y_i)$ (coordinate of $R(r)$ relative to p) and at $q' = (x_{i+m/2}, y_{i+m/2}) = (-x_i, -y_i)$ (coordinate of the point diametrically opposite to r' over the sample line $l_{i+m/2}$, relative to p), as shown in Fig. 5(c).

By performing this procedure for all pairs of opposite sample lines, the algorithm defines the set E and uses Eq. (1) to find the orientation of the tensor scale ellipse.

The localization of the edge points is formalized in Algorithm 2.

Algorithm 2.

Input: The center of the tensor scale ellipse $p = (x_p, y_p)$, the number m of sample lines, and the cost map C returned by Algorithm 1.

Output: The set E that contains m edge points localized at each sample line.

```

for each sample line  $l_i$ ,  $1 \leq i \leq m/2$  do
   $v \leftarrow \sqrt{C(p)}$ ;
   $a_1 \leftarrow -1$ ;  $a_2 \leftarrow -1$ ;
  do
     $(x_i, y_i) \leftarrow (v * \cos(\theta_i), v * \sin(\theta_i))$ , where  $\theta_i$  is the orientation
    of  $l_i$ ;
     $d \leftarrow \min(a_1, a_2)$ ;
    if  $d = a_1$ 
       $a_1 \leftarrow \sqrt{C(x_p + x_i, y_p + y_i)}$ ;
    if  $d = a_2$ 
       $a_2 \leftarrow \sqrt{C(x_p - x_i, y_p - y_i)}$ ;
     $v \leftarrow v + d$ ;
  while  $C((x_i, y_i)) \neq 0$ ;
   $E \leftarrow E \cup \{(x_i, y_i), (-x_i, -y_i)\}$ , where  $(-x_i, -y_i)$  is over the sample
  line  $l_{i+m/2}$ ;

```

Fig. 6(a) illustrates the results obtained by the new tensor scale computation method for binary images. For visualization purposes,

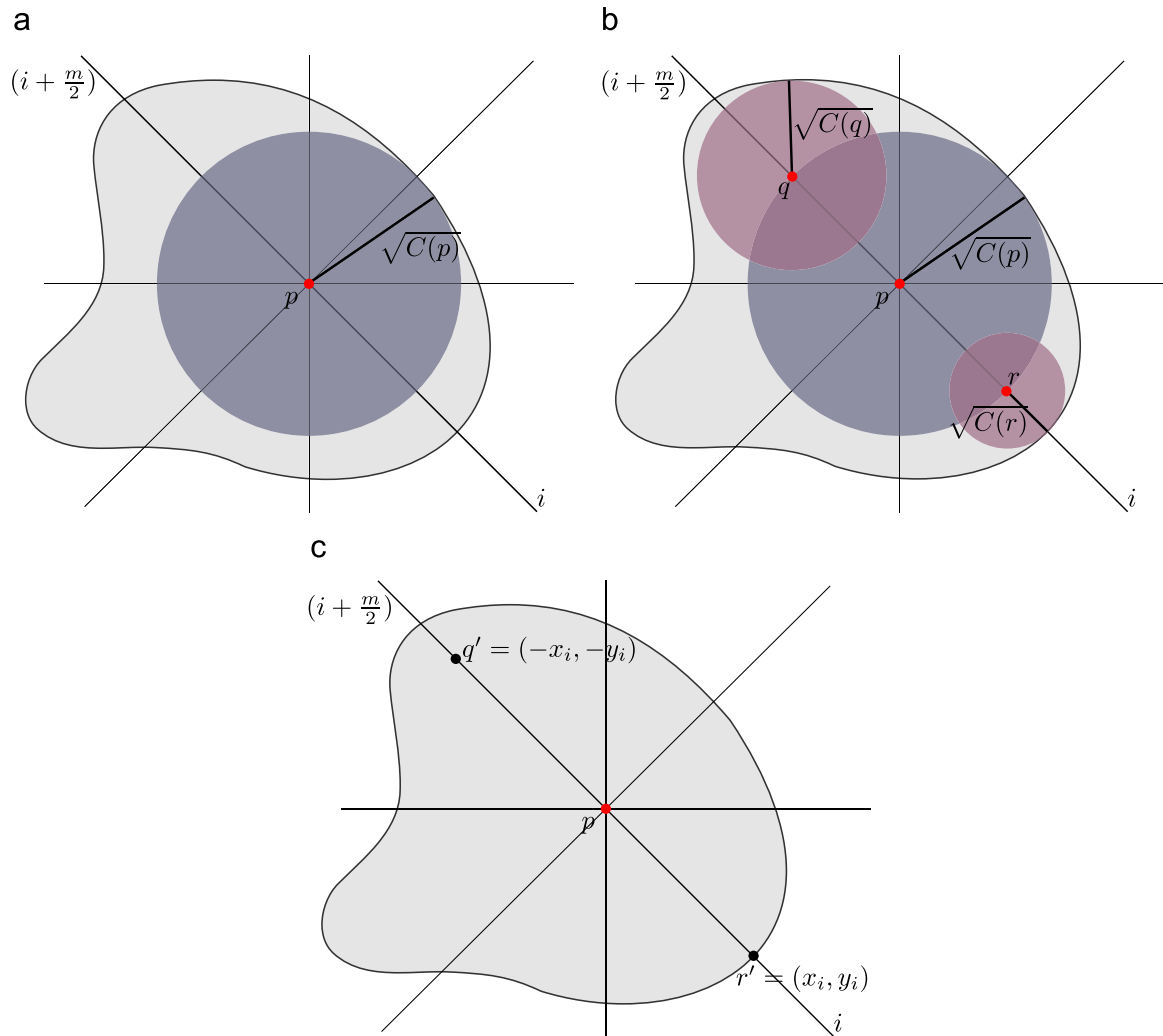


Fig. 5. Example of optimization by using Euclidean IFT.

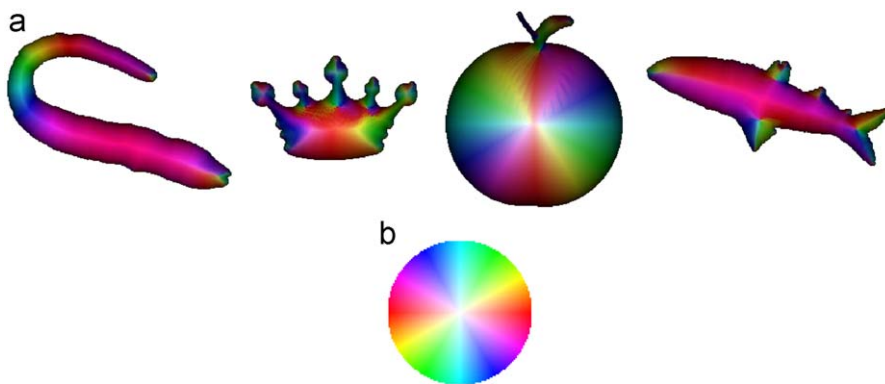


Fig. 6. New tensor scale computation method for binary images: (a) Tensor scale examples; (b) HSI color coding.

we map the center of each tensor scale to HSI color space: orientation to hue, anisotropy to saturation, and thickness to intensity. The circle of the HSI color coding (Fig. 6(b)) represents the hue for every possible orientation that an ellipse can assume.

The next section presents orientation mapping procedure based on Euclidean IFT.

3. Orientation mapping

For this step, it is necessary to compute the discrete Voronoi regions (influence zones) inside the object by Euclidean IFT (Algorithm 1), and summarize the tensor scale orientation information contained in each of these regions.

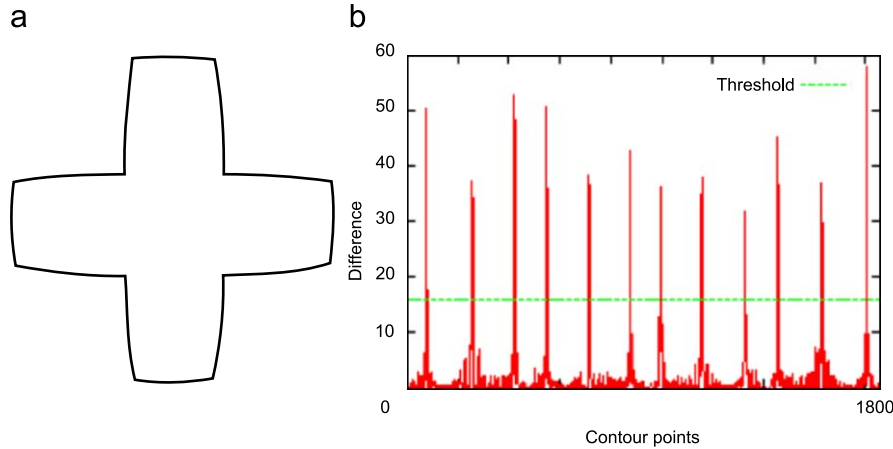


Fig. 7. Example of difference values along a contour: (a) object; (b) difference values for (a).

The saliency detector requires the influence zones of the contour points, then it uses λ_1 for obtaining the label map L . TSDIZ requires the influence zones of contour segments, then it uses λ_2 for obtaining the label map L .

The label map L groups the object's pixels into influence zones. Considering the tensor scale already computed for these pixels, we can also group the information of the ellipses in such zones. For each influence zone Z_i , with n_i pixels and label i , we have a set T_{Z_i} of tensor scale information defined as follows:

$$T_{Z_i} = \bigcup_{p=1 \dots n_i} (A(p), O(p)),$$

where $A(p)$ and $O(p)$ are the anisotropy and orientation computed for the ellipse centered at p .

For the saliency detector, the mapped orientation $M_1(s_i)$ to a given contour point s_i , with label i , is $M_1(s_i) = Summary_1(T_{Z_i})$. The function $Summary_1(T_{Z_i})$ returns the orientation $O(p)$, where $p \in Z_i$ is the center point of the tensor scale ellipse with maximum anisotropy inside the correspondent influence zone Z_i :

$$Summary_1(T_{Z_i}) = O(p) \mid (A(p), O(p)) \in T_{Z_i} \quad \text{and} \quad \forall (x, y) \in T_{Z_i} : x \leq A(p). \quad (2)$$

For the TSDIZ, the mapped orientation $M_2(s_j)$ to a given contour segment s_j , with label j , is $M_2(s_j) = Summary_2(T_{Z_j})$. The function $Summary_2(T_{Z_j})$ returns the weighted angular mean [15] of the ellipses' orientation contained in the influence zone Z_j , considering the anisotropies as the weights:

$$Summary_2(T_{Z_j}) = \arctan \left(\frac{\sum_{(O(p), A(p)) \in T_{Z_j}} A(p) * \sin(2O(p))}{\sum_{(O(p), A(p)) \in T_{Z_j}} A(p) * \cos(2O(p))} \right). \quad (3)$$

The orientation mapping step is responsible for constructing the orientation maps M_1 and M_2 for the saliency detector and the TSDIZ method, respectively.

4. Saliency point detector

The saliences of a shape are defined as the higher curvature points along the shape contour [11], or vertex points along the contour with first derivative discontinuity [12].

These saliences can be well localized by detectors that satisfy important criteria [19], such as: all true saliences should be detected; no false saliences should be detected; salience points should be well localized; robustness with respect to noise (e.g., rounded corners or peaks on the object's contour); and efficient computation.

To continue explaining the last step of our saliency point detector (the detection itself), we give a review of the necessary steps:

- (1) Computation of tensor scale for all pixels inside the object, using the approach described in Section 2.
- (2) Orientation mapping: Computation of the influence zone for each contour point, using the Euclidean IFT (Algorithm 1) with labeling function λ_1 ; and constructing the orientation map M_1 with the function $Summary_1$ (Eq. (2)).
- (3) Detection of the contour saliences, using the orientation map M_1 , as described in this section.

Note that a contour point s_i with no influence zone inside the object does not have a mapped orientation. In this case, the method considers that $M_1(s_i) = M_1(s_k)$, where s_k is the closest contour point to s_i with a mapped orientation.

In order to locate the saliency points, the method computes the differences between adjacent mapped orientations in M_1 . The difference value at $p \in M_1$ is

$$Difference(p) = AngDist(M_1(p-1), M_1(p+1)),$$

where the function $AngDist(\alpha, \beta)$ gives the smallest angle between the orientations α and β .

Now, the method uses a threshold value to eliminate low values of difference along the contour. Fig. 7(b) shows the difference values computed for every contour point of the shape illustrated in Fig. 7(a). The same graphic also shows the adopted threshold to obtain the result in Fig. 8(a).

Fig. 8 shows the detected saliences (dots) using threshold 16, i.e., saliences related to angle differences lower than 16° were not represented.

5. Tensor Scale Descriptor with Influence Zones—TSDIZ

A descriptor can be characterized by two functions: a feature vector extraction function and a similarity function. The feature vector represents the properties extracted from the image and the similarity function computes the similarity between images based on their feature vectors [13]. In this section, the feature vector extraction and similarity functions of the proposed shape descriptor are presented.

Before explaining the last step of the TSDIZ descriptor, we give a review of the method's steps:

- (1) Computation of tensor scale for all pixels inside the object, using the approach described in Section 2.

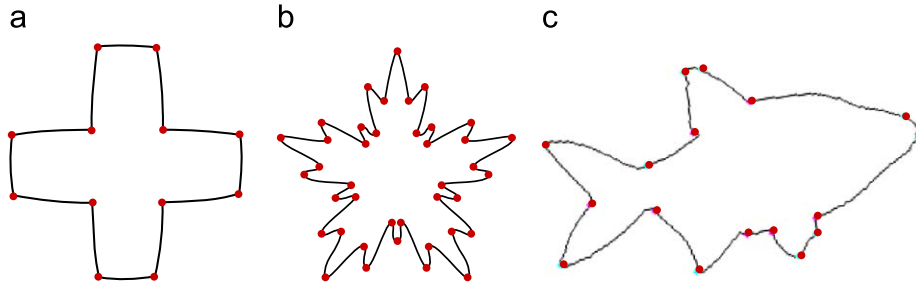


Fig. 8. Visualization of salience points.

- (2) Orientation mapping: Division of the object's contour into a few segments; computation of the influence zone for each contour segment using the Euclidean IFT (Algorithm 1) with labeling function λ_2 ; constructing the orientation map M_2 with the function $Summary_2$ (Eq. (3)).
- (3) Construction of the descriptor, as we describe in this section.

The orientation map M_2 with n_s values, where n_s is the number of contour segments, is used directly as the TSDIZ feature vector.

The similarity function has to determine the rotation difference of the orientations between two TSDIZ vectors. This function also has to determine the position (the segment) in which the feature vectors must be lined up to obtain the best matching between the underlying shapes.

The exhaustive algorithm consists of the registration between the orientation feature vectors. For this purpose, the algorithm computes, for each rotation α , where $\alpha = 0^\circ, \dots, 179^\circ$, and for each shift j in the feature vector, where $j = 1, \dots, n_s$ and n_s is the size of the vectors, the difference between the vectors, after rotating all orientations of one vector by α and circular shifting the same vector by j . The minimum difference obtained corresponds to the distance between the vectors. The distance between two TSDIZ feature vectors F_a and F_b is

$$dist_{a,b} = \min_{\substack{0 < j \leq n_s \\ 0 \leq \alpha < 180^\circ}} \left\{ \sum_{0 < i \leq n_s} AngDist(\{F_b((j-i) \bmod n_s) + \alpha\} \bmod 180^\circ, F_a(i)) \right\}.$$

The complexity of the similarity algorithm is $O(c * n_s^2)$, where c is a constant (in this case, 179°). Although it is an exhaustive search, small values of n_s (e.g., $n_s < 70$) makes it still fast. Fig. 9 illustrates the registration between two TSDIZ vectors. An orientation curve is computed for each object and then, applying the matching algorithm, these curves can be matched.

6. Experimental results

In this section, we present and discuss the results of the conducted evaluation experiments.

6.1. Image database

Experiments were conducted using two databases (Fish-shape and MPEG-7 CE-shape-1 part B).

The Fish-shape¹ database consists of 1100 fish shapes. The classes were formed by 10 variations of each original image with rotation and scaling. Then, the whole database resulted in 1100 classes with 10 images each one.

The MPEG-7² part B database is the main part of the Core Experiment CE-Shape-1. The database consists of 1400 images, categorized in 70 classes (20 images on each class). It is composed of objects silhouettes, like fruits and animals.

6.2. Results

The experiments were divided into two parts: salience detection and shape description by TSDIZ.

Our salience detector is evaluated with respect to the quality of the estimated salience points by means of their impact in shape description (Section 6.2.1). For this purpose, we replaced the salience point detector module of the Contour Saliency (CS) descriptor [16] with our detection method, and computed the same shape description functions of CS. Then, we compared both methods using the multiscale separability effectiveness measure [16].

The TSDIZ experiments (Section 6.2.2) consist in comparing the descriptor to other shape descriptors with respect to two effectiveness measures used in CBIR: precision vs. recall [17] and multiscale separability [16].

6.2.1. Saliency detector

The experiments are based on comparisons between the proposed method and the Contour Saliency (CS) method [16]. The CS approach of detecting saliences begins with the calculation of multiscale internal and external skeletons by label propagation. Then, the saliences are detected by matching each salience point of the internal skeleton to one convex point of the contour and each salience point of the external skeleton to one concave point of the contour.

For sake of simplicity, we call the CS salience detector as skeleton-based approach and the proposed salience detector as tensor scale-based approach, or just ts-based approach.

Before comparing the different approaches, we need to find the best threshold for our method. For this purpose, we constructed a database consisting of 112 shapes of the MPEG-7 part B database, resulting in 2144 saliences. The images were chosen by taking into account the obviousness of the contour salience points location. Then, a set of ground truth images were constructed with the location of the salience points.

This experiment relies on counting the true positive saliences (T_+) and false positive saliences (F_+) for the ground truth images. After this counting, three effectiveness measures were calculated: recall, precision, and accuracy. Recall (Rec) and precision (Prec) are computed as

$$Rec = \frac{T_+}{T_+ + T_-}$$

¹ <http://www.ee.surrey.ac.uk/research/vssp/imagedb/demo.html>

² <http://www.chiariglione.org/mpeg/>

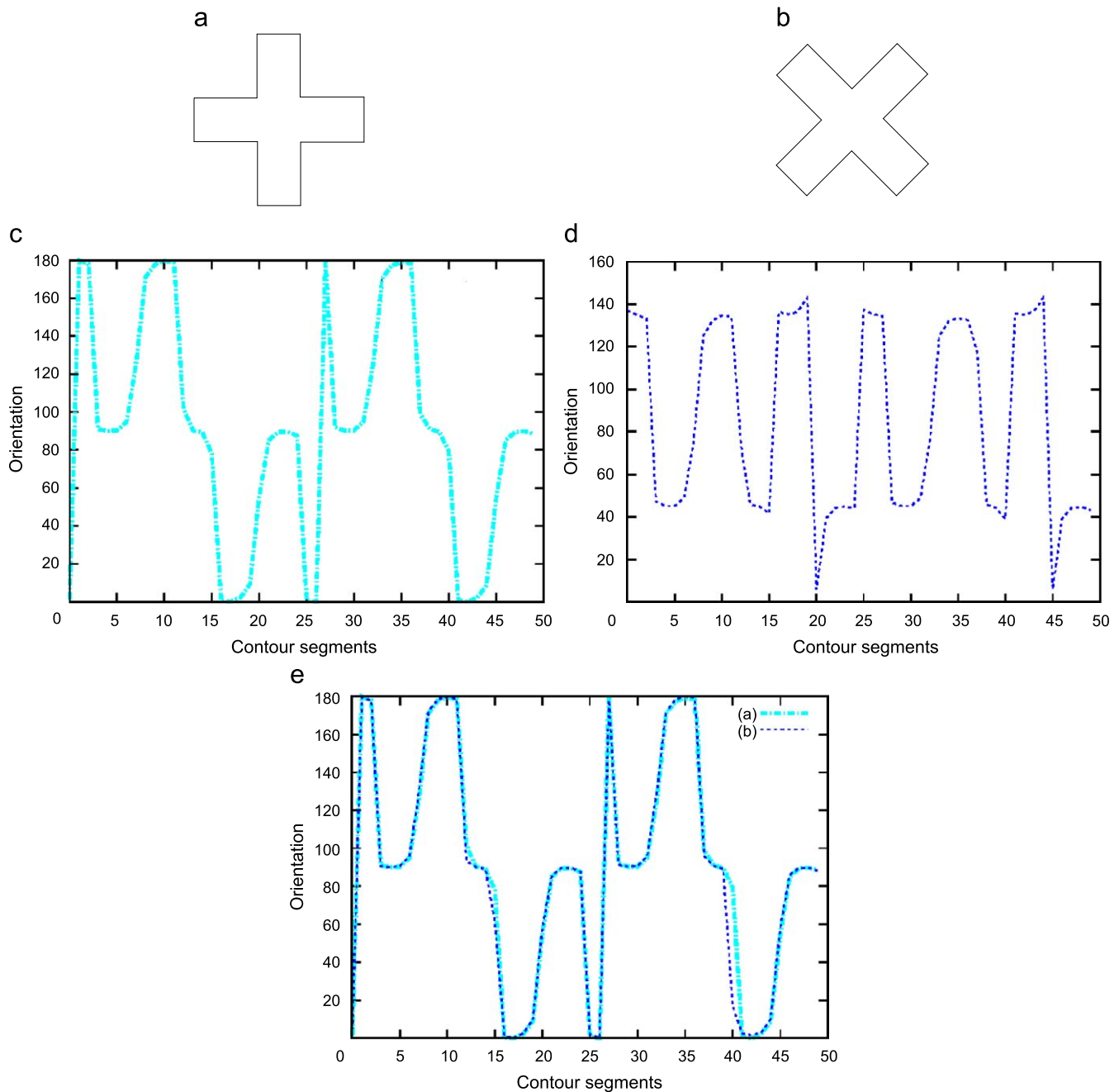


Fig. 9. Examples of TSDIZ curves and registration. (c) Orientation curve for (a). (d) Orientation curve for (b). (e) Curve matching.

and

$$Prec = \frac{T_+}{T_+ + F_+},$$

where T_- is the number of true negatives, and $(T_+ + T_-)$ represents the total number of points. The accuracy (Acc) is calculated as

$$Acc = \frac{T_+ + T_-}{T_+ + T_- + F_+ + F_-},$$

where F_- (false negatives) represents the number of miss-detections.

The results with different threshold values (10–18) for ts-based approach are presented in Table 1. Note that the method is robust to the choice of the threshold. However, the accuracy was maximized

Table 1

Effectiveness measures for tensor-scale based approach considering different threshold values (10–18).

Measures	10	12	14	16	18
Recall	0.954	0.954	0.953	0.954	0.953
Precision	0.913	0.940	0.960	0.976	0.980
Accuracy	0.881	0.901	0.914	0.943	0.919

with threshold value 16 and this is the value adopted for this method in further experiments.

The first consideration made between the approaches was related to performance issues. The ts-based approach was twice faster (speedup of 2.04), on average, than the skeleton-based approach

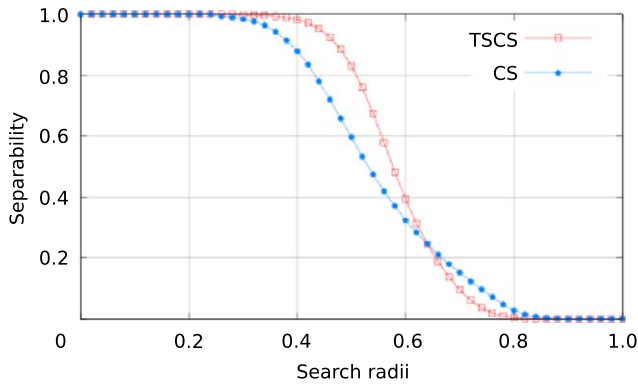


Fig. 10. Multiscale separability curve for Fish database.

used in CS [16], when executed for the entire Fish database. Experiments considered that the methods were executed on a AMD 64 3000+ Processor, with 1 GB of RAM memory.

The second consideration is that the ts-based method is computed locally, looking for each mapped orientation and for its neighbors along the contour. The skeleton-based method is more global, because it uses the internal and external skeletons of the whole shape for saliency detection. This difference in granularity also makes the detection of saliencies less robust in the skeleton-based approach, because the multiscale skeletons have to be thresholded to obtain saliency points. This threshold represents a smoothing of the contour and, consequently, loss of some important saliencies. In order to detect these saliencies, we would have to reduce the threshold. The ts-based method is also dependent of a threshold, but it is much easier to fix a single threshold for the entire database, which is the case of ts-based approach, than to find the best threshold for every single image in the database, which is the case of the skeleton-based approach.

The last consideration is about the impact of a better saliency estimation in shape description. Corners and high curvature points concentrate more information than other points of the shape [20]. For this reason, it is intuitive to conceive that curvature is an important key for the identification of many geometric aspects. Based on this, we use the saliencies as key points for shape description.

In order to compare the quality of the saliencies obtained with ts- and skeleton-based approaches, we replaced the saliency detector module of the Contour Saliency (CS) descriptor [16] with our detection method, and computed the same shape description functions of CS. We call this new descriptor as Tensor Scale Contour Saliency (TSCS). The difference between TSCS and the original CS descriptor is only the saliency detector algorithm and, consequently, the quality of the saliency points detected along the contour.

We compared both descriptors using the multiscale separability effectiveness measure. Separability indicates the discriminatory ability between objects that belong to distinct classes. This concept was introduced for CBIR in [16].

The TSCS and CS descriptors were computed for Fish-shape database and the multiscale separability curves for the descriptors are shown in Fig. 10. Higher is the curve, better is the method.

The TSCS and CS descriptors have equivalent performance for search radii less than 25% of their maximum distance. From this point to 65%, the TSCS is more robust and effective than CS. By analyzing Fig. 10, we observe that TSCS is more effective or equal to CS in up to 80% of the search radii.

6.2.2. TSDIZ experiments

In [16], Torres et al. showed that multiscale separability (MS separability) represents better than precision vs. recall (PR) curves the

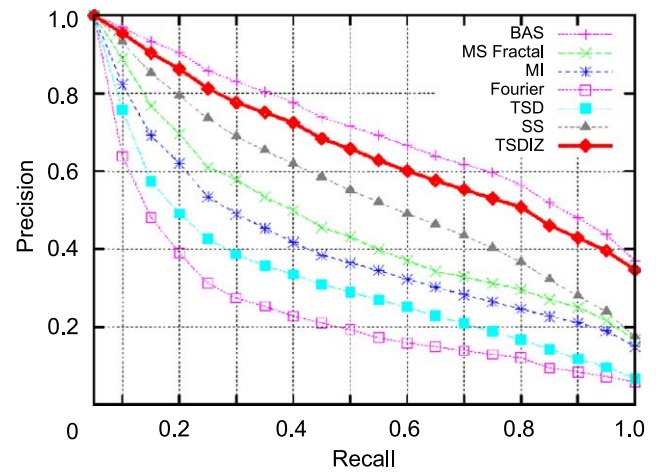


Fig. 11. PR curves for several descriptors.

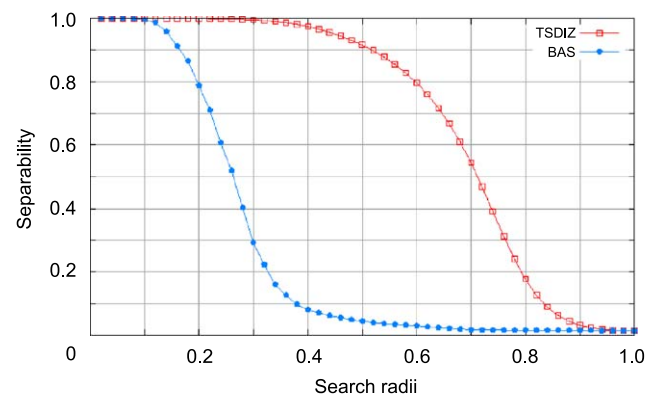


Fig. 12. Multiscale separability for BAS and TSDIZ descriptors.

separation among clusters (groups of relevant images) in the feature space. This separation is strongly related to the effectiveness of CBIR systems because the search methods rely on the clusters. However, PR is still the most popular effectiveness measure in CBIR. For this reason, we present the results with both measures.

Precision is defined as the fraction of retrieved images that are relevant to the query. In contrast, recall measures the proportion of relevant images among the retrieved images. The precision vs. recall curve, or PR curve, indicates the commitment between the two measures and, generally, the highest curve in the graph indicates better effectiveness.

In this experiment, TSDIZ is compared with the following shape descriptors: Beam Angle Statistics [18] (BAS), Multiscale Fractal Dimension [11] (MS Fractal), Moment Invariants [21] (MI), Fourier Descriptor [22] (Fourier), Tensor Scale Descriptor [8] (TSD), and Segment Saliencies [16] (SS).






















Fig. 11 presents the PR curves for the evaluated descriptors and TSDIZ with 60 contour segments, computed for MPEG-7 part B database. The method is quite robust to the number of segments. We have tested different number of segments (30–120) and the final results are consistent.

TSDIZ descriptor has the second better PR curve among the tested descriptors. BAS descriptor presented the best effectiveness according to PR.

TSDIZ has outperformed all other descriptors for MS separability, including BAS. Fig. 12 shows the MS separability curves of TSDIZ and BAS only.

TSDIZ and BAS present equivalent effectiveness for search radii less than 10% of their maximum distance. From this point on, the BAS

Table 2
Visual CBIR example.

Query	Desc.	1	2	3	4	5	6	7	8	9	10
	TSDIZ										
	BAS										

separability curve decreases quickly, indicating that this descriptor is neither robust nor effective for search radii greater than 20%.

Table 2 shows a visual CBIR example for a query image. The images that are not in the same class of the query image and should not be returned by the query are shown with a border around them.

7. Conclusions

This article introduces a faster algorithm for tensor scale computation in binary images using Image Foresting Transform (IFT), a saliency detector and a shape descriptor, both based on tensor scale. For the saliency detector, the experimental results showed that the method is faster and more robust than the saliencies detection approach proposed in [16]. The experiments done with the new version of CS descriptor (TSCS), using our saliency detector, indicate that the new approach is more effective than CS up to 80% of the search radii, according to multiscale separability measure.

In the TSDIZ method, the partition of the contour aims at efficiency in encoding contour information and tensor scale orientation mapping aims at storing spatial information into the feature vector. These TSDIZ characteristics make the descriptor compact, fast and effective for CBIR.

The experiments conducted with MPEG-7 CE-shape-1 part B database indicate that TSDIZ has better PR curve than all relevant shape descriptors (except BAS) and the best separability among them, making it the most robust and effective, according to this metric.

Future work will consider extensions of the proposed methods. The tensor scale algorithm via IFT can be extended for gray-scale images, considering each image layer (gray-scale) as a binary image itself and applying the same algorithm; or applying a segmentation method to define homogeneous regions. This idea can also be used to extend the algorithm for colored images.

Another possible extension to the tensor scale algorithm via IFT is its application to 3D shapes. In this case, the tensor scale would be represented by an ellipsoid rather than an ellipse.

Acknowledgments

Authors are grateful to FAEPEX, CAPES, CNPq, and Microsoft for financial support.

References

- [1] S. Loncaric, A survey of shape analysis techniques, *Pattern Recognition* 31 (8) (1998) 983–1001.
- [2] D. Zhang, G. Lu, Review of shape representation and description techniques, *Pattern Recognition* 37 (1) (2004) 1–19.
- [3] I. Dryden, K. Mardia, *Statistical Shape Analysis*, Wiley, London, 1998.
- [4] L.daF. Costa, J.R.M. Cesar, *Shape Analysis and Classification: Theory and Practice*, CRC Press, Florida, USA, 2001.
- [5] P. Saha, Tensor Scale: a local morphometric parameter with applications to computer vision and image processing, *Computer Vision and Image Understanding* 99 (2005) 384–413.
- [6] P. Saha, J. Udupa, Tensor scale-based fuzzy connectedness image segmentation, in: *Proceedings of the SPIE: Medical Imaging*, vol. 5032, 2003, pp. 1580–1590.
- [7] P. Saha, J. Gee, Z. Xie, J. Udupa, Tensor scale-based image registration, in: *Proceedings of the SPIE: Medical Imaging*, vol. 5032, 2003, pp. 314–324.
- [8] P. Miranda, R. da S. Torres, A. Falcão, TSD: a shape descriptor based on a distribution of tensor scale local orientation, in: *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing*, 2005, pp. 139–146.
- [9] P. Saha, F. Wehrli, A robust method for measuring trabecular bone orientation anisotropy at in vivo resolution using tensor scale, *Pattern Recognition* 37 (2004) 1935–1944.
- [10] P. Saha, F. Wehrli, In vivo assessment of trabecular bone architecture via three-dimensional tensor scale, in: *Proceedings of the SPIE: Medical Imaging*, vol. 5069, 2004, pp. 750–760.
- [11] R. da S. Torres, A. Falcão, L.da F. Costa, A graph-based approach for multiscale shape analysis, *Pattern Recognition* 37 (6) (2004) 1163–1174.
- [12] L.da F. Costa, A.G. Campos, E.T.M. Manoel, An integrated approach to shape analysis: results and perspectives, in: *Proceedings of the International Conference on Quality Control by Artificial Vision*, vol. 1, 2001, pp. 23–24.
- [13] R. da S. Torres, A.X. Falcão, Content-based image retrieval: theory and applications, *Revista de Informática Teórica e Aplicada* 13 (2) (2006) 161–185.
- [14] A.X. Falcão, J. Stolfi, R.A. Lotufo, The image foresting transform: theory, algorithms, and applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (1) (2004) 19–29.
- [15] K.V. Mardia, P.E. Jupp, *Directional statistics*, in: *Wiley Series in Probability and Statistics*, Wiley, New York, 1999.
- [16] R. da S. Torres, A.X. Falcão, Contour saliency descriptors for effective image retrieval and analysis, *Image and Vision Computing* 25 (1) (2007) 3–13.
- [17] H. Müller, W. Müller, D.M. Squire, S. Marchand-Maillet, T. Pun, Some informational aspects of visual perception, *Pattern Recognition Letters* 22 (5) (2001) 593–601.
- [18] N. Arica, F.T.Y. Vural, BAS: a perceptual shape descriptor based on the beam angle statistics, *Pattern Recognition Letters* 24 (9–10) (2003) 1627–1639.
- [19] F. Mokhtarian, R. Suomela, Robust image corner detection through curvature scale space, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (12) (1998) 1376–1381.
- [20] F. Attneave, Some informational aspects of visual perception, *Psychological Review* 61 (3) (1954) 183–193.
- [21] M.K. Hu, Visual pattern recognition by moment invariants, *IEEE Transactions on Information Theory* 8 (2) (1962) 1163–1174.
- [22] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, second ed., Addison-Wesley Longman Publishing Co. Inc., Reading, MA, 2001.

About the Author—FERNANDA ALCÂNTARA ANDALÓ received a B.Sc. in Computer Science from the University of Brasília, Brazil in 2004. In 2007, she received a M.Sc. in Computer Science from the University of Campinas, Brazil. She is currently a Ph.D. student at Institute of Computing, University of Campinas, and her research interests include image processing, shape analysis, content-based image retrieval, and computer vision.

About the Author—PAULO ANDRÉ VECHIATTO MIRANDA received the Computer Engineering degree (2003) and the M.Sc. degree in Computer Science (2006) from the State University of Campinas (Brazil). Since 2006, he is pursuing a Ph.D. in Computer Science at the State University of Campinas. His research involves image analysis, segmentation, medical image analysis, object recognition, and content-based image retrieval.

About the Author—RICARDO DA SILVA TORRES received a B.Sc. in Computer Engineering from the University of Campinas, Brazil, in 2000. He got his doctorate in Computer Science at the same university in 2004. He has been Professor at the Institute of Computing, University of Campinas, since 2005, and his research interests include image analysis, content-based image retrieval, image databases, and geographic information systems.

About the Author—ALEXANDRE XAVIER FALCÃO received a B.Sc. in Electrical Engineering from the University of Pernambuco, PE, Brazil, in 1988. In 1993, he received a M.Sc. in Electrical Engineering from the University of Campinas, SP, Brazil. During 1994–1996, he worked at University of Pennsylvania, PA, USA, on interactive image segmentation for his doctorate. He got his doctorate in Electrical Engineering from the University of Campinas in 1996. He has been Professor at the Institute of Computing, University of Campinas, since 1998, and his research interests include image segmentation and analysis, volume visualization, content-based image retrieval, mathematical morphology, digital TV and medical imaging applications.