Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 141–152.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.18.31

A new shape descriptor based on tensor scale

Fernanda A. Andaló, Paulo A. V. Miranda,
Ricardo da S. Torres and Alexandre X. Falcão

Instituto de Computação (IC), Universidade Estadual de Campinas (Unicamp), SP, Brazil
{feandalo,paulo.miranda,rtorres,afalcao}@ic.unicamp.br

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 paper, we exploit this concept for binary images and propose a shape descriptor that encodes region and contour properties in a very efficient way. Experimental results are provided, showing the effectiveness of the proposed descriptor, when compared to other relevant shape descriptors, with regard to their use in content-based image retrieval systems.

Keywords: image processing, shape description, image retrieval.

1. Introduction

The recent growth of the World Wide Web and the new technologies that became available for image acquisition and storage have increased the demand for image retrieval systems based on image properties.

In content-based image retrieval (CBIR) systems, image processing techniques are used to describe the image content, encoding image properties that are relevant to the query. Usually, these properties are represented by shape, color, and texture descriptors of objects or regions within the image. 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 [10].

The shape of an object is an important and basic visual feature for describing image content [14]. Shape representation generally aims at effective and perceptually important shape features based on boundary information – *contour-based methods* – and/or on interior content – *region-based methods*. Each class can be further broken into *structural* and *global approaches*, depending on whether the shape is represented as a whole or by segments or sections [14].

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 141–152.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.18.31

In this work, we propose a new descriptor based on *tensor scale* that exploits region and contour information. Tensor scale [9] is a morphometric parameter yielding a unique representation of local structure thickness, orientation, and anisotropy. That is, at any image point, its tensor scale is represented by the largest ellipse (2D), or ellipsoid (3D), centered at that point and within the same homogeneous region.

We exploit the tensor scale concept for objects and, for sake of simplicity, we only consider objects with a single contour. The descriptor computes the tensor scale ellipse for every object point, divides the object's contour into a predefined number of segments, computes the *influence zone* of each segment and assigns, to each segment, the weighted angular mean orientation [5] of the ellipses within its influence zone. The influence zone of a segment consists of the object pixels that are closest to pixels of that segment than to any other pixel along the contour.

By dividing the contour into a small number of ordered segments, we are aiming at efficiency in encoding contour information. By mapping tensor scale orientation onto the segments, we are exploiting region information. As we will show, this makes the proposed descriptor compact, efficient, and effective for CBIR.

A previous shape descriptor based on tensor scale – Tensor Scale Descriptor [6] (TSD) – was proposed based on the histogram of the tensor scale orientation of the ellipses. Our descriptor introduces a totally new way of exploiting tensor scale orientation, which includes spatial information. We also present a much faster computation of the ellipses by exploiting the Image Foresting Transform (IFT) [2].

2. Background

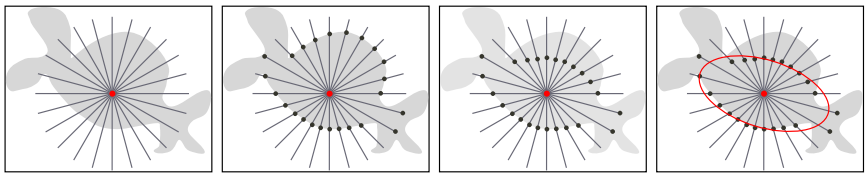
In [9], Punam introduced a local method for gray-scale images – tensor scale – represented by the largest ellipse within the same homogeneous region, centered at a given pixel p. This method defines the ellipse by three factors:

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

where $|t_1(p)|$ and $|t_2(p)|$, with $|t_1(p)| \geq |t_2(p)|$, denote the length of the two semi-axis of the ellipse centered at p.

A tensor scale ellipse is calculated from sample lines that are traced around a given pixel, from 0 to 179 degrees (Figure 1(a)). The axes of the ellipse are determined by computing the image intensities along each of the sample lines and the location of two opposite edge points on these lines

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 141–152.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.18.31



(a) (b) (c) (d)

Figure 1. Tensor scale computation.

(Figure 1(b)). The next step consists of repositioning the edge locations to points equidistant to that given pixel, following the axial symmetry of the ellipse (Figure 1(c)). The computation of the best-fit ellipse to the repositioned edge locations is done by Principal Component Analysis (PCA) (Figure 1(d)).

These computations are performed for every pixel of the image. A critical drawback of Punam's approach is that the computational cost of the algorithm makes his method quite prohibitive for CBIR systems. For this reason, Miranda et al. [6] proposed an efficient implementation of the original method, which differs in the following aspects.

The first change was in the edge location phase, in which Miranda's approach is to go along each pair of opposite segments, alternately and at the same time, instead of going along one entire segment by turn. By doing this, the reposition phase is no longer necessary. The second change was the use of two connected thresholds to improve and simplify the original method of detecting edges. The third and final change was the improvement of the ellipse computation phase. Miranda et al. proposed a function g (Equation 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.

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

where $x_{i_\gamma} = x_i \cos(\gamma) - y_i \sin(\gamma)$, $y_{i_\gamma} = x_i \sin(\gamma) + y_i \cos(\gamma)$, (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. [6] 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 shape (this is also valid for texture, i.e., their descriptor could be easily extended for colored images). The TSD descriptor computes the tensor scale parameters for the original image and then computes the local orientation histogram, used

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 141–152.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.18.31

as feature vector. The matching of two given images by TSD is made by taking the absolute difference of the area between their orientation histograms, after correcting the displacement by correlation (i.e., object rotations cause shifts in the histogram).

In the next sections, we propose a new shape descriptor based on tensor scale – Tensor Scale Descriptor with Influence Zones (TSDIZ) – and show how it can provide considerable improvements in CBIR. We also provide a faster computation of tensor scale, as compared to previous approaches [6,9], by exploiting the Euclidean Image-Foresting Transform [12].

3. The TSDIZ descriptor

The key idea of the proposed descriptor is to map the tensor scale orientations inside an object onto a few segments of its contour, and use this information for shape description.

Orientation mapping is done by exploiting the discrete Voronoi regions (influence zones) of contour segments inside the object. The discrete Voronoi regions can be efficiently obtained by label propagation using the Image Foresting Transform (IFT) [2].

The IFT is a graph-based approach to the design of image processing operators based on connectivity, 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. For a given seed set, each seed defines an influence zone consisting of the pixels that are "more closely connected" to that seed than to any other, according to a path-cost function [2]. We use a path-cost function that assigns the closest Euclidean distance between object pixels and contour pixels to each pixel inside the object (Euclidean IFT – Euclidean distance transform via IFT).

The TSDIZ approach divides the contour into segments, labels each contour segment with a distinct number, and propagates these labels inside the object via Euclidean IFT. It is assigned to each segment, a weighted angular mean orientation of the ellipses inside its influence zone, using their anisotropies as weights.

In the next section, we present the Euclidean IFT that is used for tensor scale computation and tensor scale mapping.

3.1 Euclidean IFT

The Euclidean IFT is used for two purposes in TSDIZ: faster tensor scale computation (Section 3.2) and tensor scale orientation mapping (Section 3.3). The advantages of calculating the Euclidean Distance Transform via IFT is that label propagation is executed on-the-fly and in linear time.

In the Euclidean IFT (Algorithm 1), the path-cost function is such that the cost of a path from a seed s to a pixel t in the forest is the Euclidean

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 141–152.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.18.31

distance between s and t. The algorithm also needs an Euclidean relation A that is defined as

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

where ρ is the adjacency radius and (x_i, y_i) are the coordinates of a pixel i in the image.

Our Euclidean IFT assigns three attributes to each object 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 $L(p) = L(s)$ of the segment that contains s (forming a label map).

Algorithm 1 Euclidean Distance Transform via IFT.

Input: A binary image I, a set S of contour pixels in I (seeds), an Euclidean adjacency relation A, and a labeling function $\lambda(p)$ that assigns a segment label to each pixel p in S.
Output: The cost map C, the root map R, and the label map L.
Auxiliary data structure: A priority queue Q.

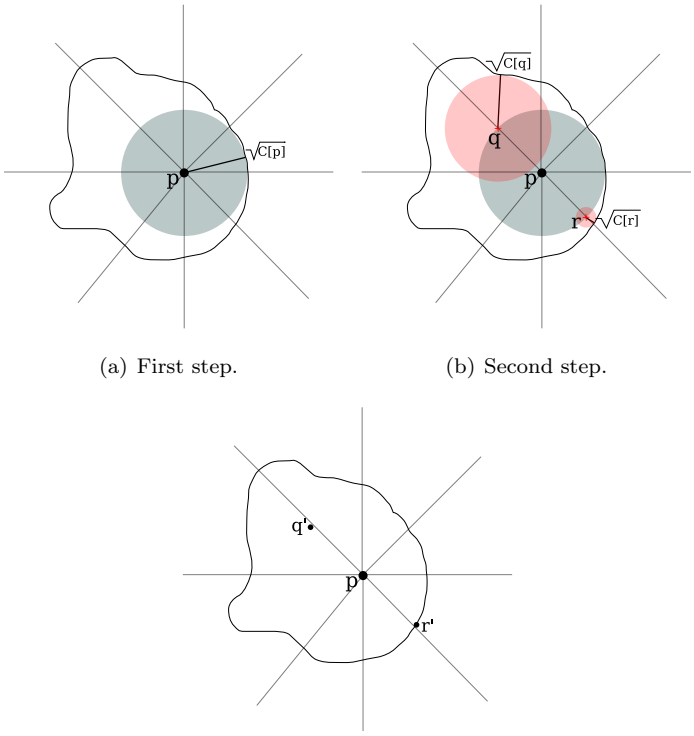
```
begin
    foreach p ∈ I do
        C(p) ← +∞; R(p) ← NIL; L(p) ← NIL
    foreach p ∈ S do
        C(p) ← 0; R(p) ← p; L(p) ← λ(p)  insert p in Q
    while Q is not empty do
        remove from Q a pixel p = (x_p, y_p) such that C(p) is minimum
        foreach q = (x_q, y_q) such that q ∈ A(p) and C(q) > C(p) do
            C' ← (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) ≠ +∞ then
                    remove q from Q
                C(q) ← C'; R(q) ← R(p); L(q) ← L(p)  insert q in Q
end
```

3.2 Faster tensor scale computation for binary images

A considerable speedup in the computation of the tensor scale for binary images is possible by exploiting the following aspect: if we have the shortest distance between a pixel p and the contour, there is no need to search for edge points inside the circle with radius $\sqrt{C(p)}$ (Figure 2(a)).

According to Miranda's algorithm, edge points are searched along opposite sample lines, alternately. In our approach, the algorithm jumps along the lines and visits the pixels q and r at the same time (Figure 2(b)).

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 141–152.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.18.31



(a) First step. (b) Second step.

(c) Third step.

Figure 2. Example of optimization made by using Euclidean IFT.

The searching for edge points continues outside the area defined by the cost $\sqrt{C(p)}$ in Figure 2(b), and the minimum between $\sqrt{C(r)}$ and $\sqrt{C(q)}$ indicates the location for the next jump. These jumps may continue iteratively until the closest edge point along the sample line is found.

In the example, the edge is found at the pixel $R(r)$ (i.e., at the contour point r' nearest to r). The algorithm defines that the two edge points in this sample line are at r' (coordinate of $R(r)$ relative to p) and at q' (coordinate of the point diametrically opposite to r', relative to p), as shown in Figure 2(c).

By performing this procedure for all sample lines, the algorithm defines all edge points and uses the same formula defined by Miranda et al. (Equation 1) for finding the orientation of the ellipse.

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

The next section presents orientation mapping based on Euclidean IFT.

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 141–152.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.18.31

Algorithm 2 Edge points localization for ellipse centered at pixel p.

Input: A pixel $p = (x_p, y_p)$, the number m of sample lines, and the cost map C returned by Algorithm 1.

Output: The vector *edge* that contains m pairs of edge points localized at the sample lines.

begin

  for $\theta \leftarrow 0°$ **to** $179°$, *with increments* $\frac{180}{m}$ **do**

    $v \leftarrow \sqrt{C(p)}$ $p_1 \leftarrow NIL$; $p_2 \leftarrow NIL$; $q_1 \leftarrow 0$; $q_2 \leftarrow 0$ **while** $p_1 \neq 0$ *and* $p_2 \neq 0$ **do**

      $x \leftarrow v * \cos(\theta)$; $y \leftarrow v * \sin(\theta)$ **if** $q_1 = 0$ **then**

        $temp \leftarrow (x_p + x, y_p + y)$; $p_1 \leftarrow \sqrt{C(temp)}$;

      if $q_2 = 0$ **then**

        $temp \leftarrow (x_p - x, y_p - y)$; $p_2 \leftarrow \sqrt{C(temp)}$;

      $d \leftarrow min(p_1, p_2)$; $v \leftarrow v + d$ $q_1 \leftarrow p_1 - d$; $q_2 \leftarrow p_2 - d$

    $edge[\theta] \leftarrow ((x, y), (x', y'))$, where (x', y') is the coordinate of the point diametrically opposite to (x, y), relative to p

end

3.3 Feature vector of TSDIZ by orientation mapping onto contour segments

Prior to tensor scale orientation mapping, TSDIZ approach has two stages: tensor scale computation for all pixels inside the object and partition of the object contour into segments.

The orientation mapping uses the label map L returned by the Euclidean IFT (Algorithm 1). The map L groups pixels and their ellipses in the influence zone of each segment. TSDIZ uses as feature vector F the weighted mean of the ellipses orientations in each influence zone of segment. Therefore, F can be indexed by L. The TSDIZ feature vector is formed by the mapped tensor scale orientations (F) and has size equal to the number n_s of segments. Algorithm 3 shows the feature vector computation for TSDIZ.

The function $WeightedAngularMean(V[i])$ returns the weighted angular mean of the ellipses orientations contained in influence zone i, $i = 1, 2, \ldots, n_s$, considering the anisotropies as the weights. The mean $\overline{\theta}$ for angular data [5] is calculated as

$$\overline{\theta} = arctan\left(\frac{\sum_{p=1}^{n} Ani[p] * \sin(2 * Ori[p])}{\sum_{p=1}^{n} Ani[p] * \cos(2 * Ori[p])}\right). \quad (3)$$

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 141–152.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.18.31

Algorithm 3 Feature vector computation for TSDIZ by tensor scale orientation mapping.

Input: A binary image I containing an object O, the number n_s of contour segments, the label map L returned by Algorithm 1, and the vectors Ani and Ori that contain the anisotropies and the orientations of the tensor scale ellipses computed for all pixels of object O, respectively.

Output: A feature vector F that contains the mapped orientation for each contour segment.

Auxiliary data structure: A vector V of n_s lists to store ellipse information in each influence zone.

begin
  foreach $p \in O$ **do**
    insert $(Ani[p], Ori[p])$ in list $V[L(p)]$, where $L(p)$ is the label of the influence zone in which p is contained
  foreach $i \in [1, \ldots, n_s]$ **do**
    $F[i] = WeightedAngularMean(V[i])$
end

3.4 TSDIZ similarity function

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 (Algorithm 4) consists of the registration between the orientation feature vectors. Considering $\alpha = 0°, \ldots, 179°$ and $j = 1, \ldots, n_s$, where n_s is the size of the vectors, the algorithm computes, for each rotation α and for each shift j in the feature vector, the difference between the vectors, after rotating all orientations of one vector by α and circular shifting the same vector by j. The minimun difference obtained corresponds to the distance between the vectors.

In Algorithm 4, the function $AngularDistance(\alpha, \beta)$ gives the smallest angle between the orientations α and β.

The complexity of this 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. Figure 3 illustrates the registration between two TSDIZ vectors.

4. Experimental results

In this section, the results of the experiments in CBIR are presented.

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 141–152.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.18.31

Algorithm 4 Similarity between two TSDIZ vectors.

Input: Two feature vectors F_A and F_B.
Output: Distance $dist$ between F_A and F_B.
begin
  $dist \leftarrow \infty$ **foreach** $j \in [1, \ldots, n_s]$ **do**
    foreach $\alpha \in [0, \ldots, 179]$ **do**
      foreach $i \in [1, \ldots, n_s]$ **do**
        $dist_{aux} \leftarrow AngularDistance(\{F_B[(j - i) \mod n_s] + \alpha\} \mod 180, F_A[i])$ **if** $dist_{aux} < dist$ **then**
          $dist \leftarrow dist_{aux}$
end

4.1 Image database

Experiments were conducted using MPEG-7 CE-shape-1 part B [7] database, which consists of 1400 images, categorized in 70 classes (20 images on each class). It is composed by objects silhouettes, like fruits and animals.

4.2 Results

The experiments consist in comparing the TSDIZ and other shape descriptors, with respect to two effectiveness measures used in CBIR – precision versus recall [8] (PR) and multiscale separability [11] (MS separability).

In [11], Torres et al. showed that MS separability represents better than PR curves the separation among clusters (groups of relevant images) in the feature space. This separation is strongly related to the performance of CBIR systems because the search methods rely on them. However, PR is still the most popular effectiveness measure in CBIR. For this reason, we present the results with both measures.

Precision versus recall

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 versus 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 [1] (BAS), Multiscale Fractal Dimension [13] (MS Fractal), Moment Invariants [4] (MI), Fourier Descriptor [3] (Fourier), Tensor Scale Descriptor [6] (TSD) and Segment Saliences [11] (SS).

Figure 4(a) presents the PR curves for the evaluated descriptors and TSDIZ with 60 contour segments. The number of segments is a parameter

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 141–152.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.18.31

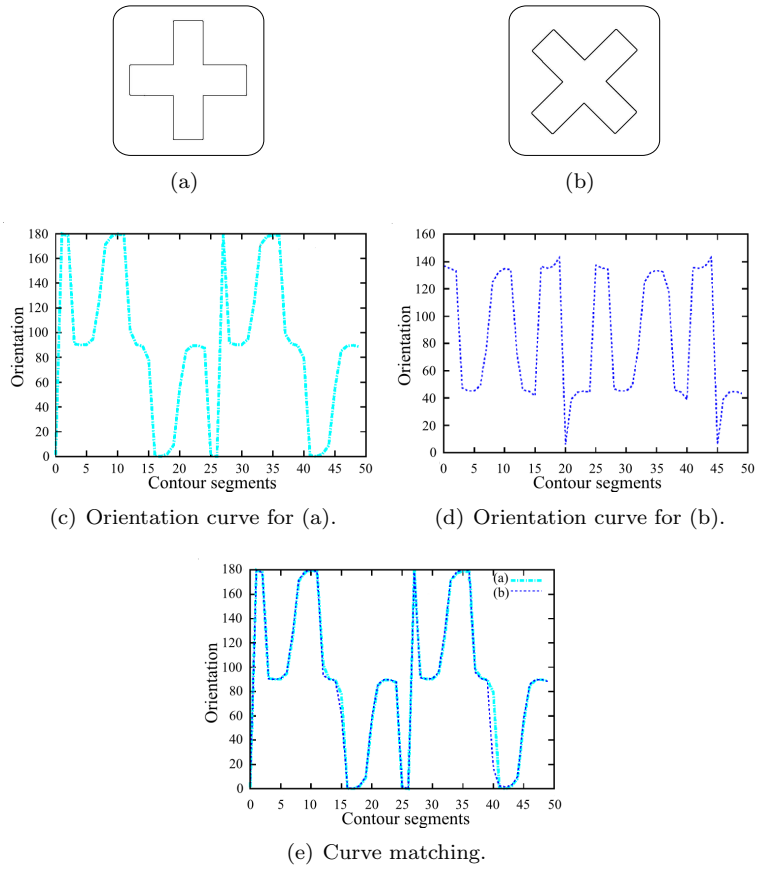


Figure 3. Examples of TSDIZ curves and registration.

that is tuned for each database and can be learned by training.

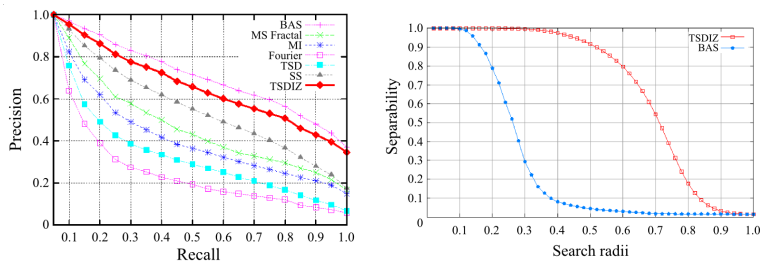
TSDIZ descriptor has the second best PR curve among the tested descriptors. BAS descriptor presented the best performance according to PR, as expected for this database [1].

Multiscale separability

A good effectiveness measure should capture the concept of separability. Separability indicates the discriminatory ability between objects that belong to distinct classes. This concept is widely used in cluster analysis, and it was introduced for CBIR by Torres et al. [11].

As TSDIZ has outperformed all other descriptors for MS separability as well, we show in Figure 4(b) the MS separability curves of TSDIZ and

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 141–152.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.18.31



(a) PR curves for several descriptors. (b) Multiscale separability curves for TSDIZ and BAS descriptors.

Figure 4. Effectiveness measures experiments conducted in MPEG-7 CE-shape-1 part B database.

BAS only. TSDIZ and BAS present equivalent performance for search radii less than 10% of their maximum distance. From this point on, the BAS separability curve decreases quickly, indicating that this descriptor is neither robust nor effective for search radii greater than 20%.

Figure 5 shows a visual CBIR example for a query image. The images with a gray background are not in the same class of the query image and should not be returned by the query.

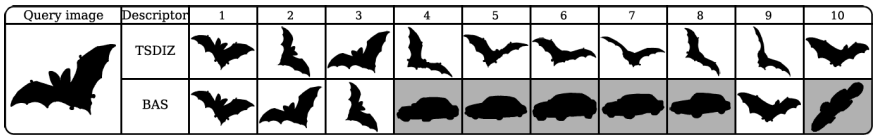


Figure 5. Visual CBIR example.

5. Conclusions and future work

This paper introduced a new shape descriptor based on tensor scale (TSDIZ). It also provided a faster algorithm for tensor scale computation using Image Foresting Transform (IFT).

The feature vector consists of the tensor scale orientations computed for all pixels of a given object and mapped onto contour segments. The partition of the contour aims at efficiency in encoding contour information and tensor scale orientation mapping aims at storing spatial information into TSDIZ feature vector. These TSDIZ characteristics make the descriptor compact, fast and effective for CBIR.

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 1, p. 141–152.
http://urlib.net/dpi.inpe.br/ismm@80/2007/03.19.18.31

The experiments done 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.

The TSDIZ feature extraction function only computes tensor scale ellipses inside objects. Future works will be directed towards incorporating information from ellipses outside the object as well. The TSDIZ descriptor will also be evaluated with other shape databases.

6. Acknowledgments

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

References

[1] N. Arica and F. T. Y. Vural, *BAS: a perceptual shape descriptor based on the beam angle statistics*, Pattern Recognition Letters **24** (2003), no. 9–10, 1627–1639.

[2] A. X. Falcão, J. Stolfi, and R. de A. Lotufo, *The Image Foresting Transform: theory, algorithms, and applications*, IEEE Transactions on Pattern Analysis and Machine Intelligence **26** (2004), no. 1, 19–29.

[3] R. C. Gonzalez and R. E. Woods, *Digital image processing*, 2nd, Addison-Wesley Longman Publishing Co. Inc., 2001.

[4] M.-K. Hu, *Visual pattern recognition by moment invariants*, IEEE Transactions on Information Theory **8** (1962), no. 2, 1962.

[5] K. V. Mardia and P. E. Jupp, *Directional statistics*, John Wiley and Sons, 1999.

[6] P. A. V. Miranda, R. da S. Torres, and A. X. Falcão, *TSD: a shape descriptor based on a distribution of tensor scale local orientation*, Proceedings of Brazilian Symposium on Computer Graphics and Image Processing, 2005, pp. 139–146.

[7] *The MPEG Project*. <http://www.chiariglione.org/mpeg>. Access in: Jan,2007.

[8] H. Müller, W. Müller, D. McG. Squire, S. Marchand-Maillet, and T. Pun, *Performance evaluation in content-based image retrieval: overview and proposals*, Pattern Recognition Letters **22** (2001), no. 5, 593–601.

[9] P. K. Saha, *Tensor scale: a local morphometric parameter with applications to computer vision and image processing*, Computer Vision and Image Understanding **99** (2005), no. 3, 384–413.

[10] R. da S. Torres and A. X. Falcão, *Content-based image retrieval: theory and applications*, Revista de Informática Teórica e Aplicada **13** (2006), no. 2, 161–185.

[11] ———, *Contour salience descriptors for effective image retrieval and analysis*, Image and Vision Computing **25** (2007), no. 1, 3–13.

[12] R. da S. Torres, A. X. Falcão, and L. da F. Costa, *Shape Description by Image Foresting Transform*, Proceedings of International Conference on Digital Signal Processing, 2002, pp. 1089–1092.

[13] ———, *A graph-based approach for multiscale shape analysis*, Pattern Recognition **37** (2004), no. 6, 1163–1174.

[14] D. Zhang and G. Lu, *Review of shape representation and description techniques*, Pattern Recognition **37** (2004), no. 1, 1–19.