

Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

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

Textual description of shapes

Slimane Larabi *

Computer Science Department, USTHB University, BP 32 EL ALIA, Algiers 16000, Algeria

ARTICLE INFO

Article history:

Received 14 July 2007

Accepted 20 August 2009

Available online 2 September 2009

Keywords:

Shape

Outline shape

Part

Description

XML language

Similarity

Shape retrieval

Image coding

ABSTRACT

We propose in this paper a method for textual description of shape. In the first, we propose a part-based method for the decomposition of outline shape into parts and separating lines. These elements are described geometrically and written with a structured text. In the next we propose the representation of shape by a tree structure where nodes are the internal regions, and arcs correspond to the inclusion relation between these regions. The tree structure is translated to a structured text where is associated for each node a set of attributes such as the description of the outline region, its color and its position.

A set of applications of this descriptor are proposed. We show how to decode and visualize the shape from its descriptor and how to retrieve a shape query using shapes database. To do this, a shape similarity is defined and extracted directly from the descriptor.

Experiments conducted over real images of various silhouettes and shapes are presented and discussed.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

The representation of shape remains one of the most difficult problems in computer vision. Various methods have been developed in order to represent shape in an abstract and efficient way while still preserving important shape feature. The most interesting methods may be classified as follows:

- Part-based methods where silhouette is decomposed into parts. Different criterions have been proposed: limb and neck [40], optimization of the convexity of all parts [34] and morphology criteria [26,31]. In [45] shape is decomposed into union of meaningful convex subparts by a recursive scheme. The shape of each subpart is approximated by a morphological dilatation of basic structuring elements. Constrained morphological which recursively performs decomposition scheme for shapes is used in [16]. In [23], morphological set operations are used to represent and encode a discrete binary image by parts of its skeleton.
- Aspect-graph methods are viewer-centered representations of a three-dimensional object. The underlying theory was introduced by Koenderink and Doorn [18] who observed that while for most views a small change in the vantage point of an object results in small change in the shape of the projection of that object, for certain views the change in projected object shape is dramatic. These unstable views represent a singularity in the visual mapping or a transition. They suggested that a derivation of such transition boundaries is a good representation of the object.

The stable views, also called general views, are what define an aspect. In [10], Cyr and Kimia propose an aspect graph representation of an object where edges of the graph are the transitions between two neighboring stable views and a change between aspects in called a visual event.

- Methods that use the medial axis of silhouettes. Zhu and Yuille in [47] propose flexible object recognition and modeling system where shape representation is based on medial axis extraction and part segmentation using deformable circles. In [35], Ruberto proposes to represent the medial axis characteristic points as an attributed skeletal graph to model the shape. Shape has been also represented by an axis trees (S-A-trees) [14].
- Methods based on the shock graph that is an emerging shape representation for object recognition, in which a 2D silhouette is decomposed into a set of qualitative parts, captured in a directed acyclic graph [41]. Sebastian et al. in [36] use a graph structure for shape representation where shocks with isolated point topology are nodes and those with curve topology are links.
- Methods using graph for shape representation: concavity graph that is a directed graph with a unique root for representing single as well multi-object images. For Badawy and Kamel in [3] five types of nodes in the graph representing five types of conceptual or logical regions: objects, concavities, holes, multiple objects and nodes representing multiple holes. In [22], images as well as stored models are represented as graphs whose vertices correspond to object corners and whose edges correspond to outlines connecting corners in the image.
- Approximation of outline shape by 2D features. Grosky and Mehrota in [15] have approximated shapes as polygonal curves: for each vertex, a local feature is defined by considering the

* Fax: +213 21 24 76 07.

E-mail addresses: slarabi@usthb.dz, slimane.larabi@lifl.fr

internal angle at the vertex, the distance from the adjacent vertex, and the vertex coordinates. Petrakis et al. in [30] have approximated shapes as a sequence of convex/concave segments between two consecutive curvature points. Concave and convex sections are used in [28] that use a set of segmented contours fragments broken at points of high curvature. The longest curves are selected as keys curves, and a fixed-size template is constructed. The template represents not just the keying fragment but all portions of other curves that intersect the square. Cronin in [9] present a new method based on concavity code to partition a digital contour into concave and convex sections and labels each point with a string.

- Methods based on the reference points of outline shape. Mokhtarian and Mackworth present in [25,26] propose a multiscale curvature-based shape representation technique for planar curves. In [39] set of reference points such as corners and bi-tangent points which are stable under the various transformations are used to represent shapes. Shape is partitioned at points where the shape curvature is minima. These points identify tokens that correspond to protrusions of the curve and can be used as signature [6].
- Methods based on the attributes of outline shape. In [2], silhouette is described with one dimensional descriptor, which preserves the perceptual structure of its shape. The proposed descriptor is based on the moments of the angles between the bearings of a point on the boundary, in a set of neighborhood systems. At each point of the boundary, the angle between a pair of bearings is calculated to extract the topological information of the boundary in a given locality. In [5], shape is represented directly by its contour in term of the angle from the centroid and normalized radial length. This method produces a numeric result and preserves the shape information. Shape is also represented by invariant points obtained using properties that are preserved under projection, such as tangency [29]. Sethi et al. in [38] propose a method for detecting the silhouette curve, computing its pedal curve (defined as the set of its tangent lines) and constructing its signature.
- Method based on the shape context introduced to describe the coarse distribution of the rest of the shape, with respect to a given point of the shape [4].
- Appearance-based methods proposed initially by Murase and Nayer [27].

A review of shape representation methods can be found in [7,46]

Even if each one of the proposed method for shape representation verifies some or all properties required in computer vision applications such as structure preserving, invariance to rotation, translation, scale change, easiness in computation . . . , nevertheless, the proposed silhouette descriptors are numerical value organized as a list or graph structure.

In order to homogenous all databases of shapes models and to offer accessibility to all users, a common and compact format of representation is recommended. In addition to known criterions of any representation of shapes, this format must be easy to index, easy to compare and efficient for computation and storage.

Our aim in this paper is to propose a shape representation method verifying the above recommendations. Firstly, we propose a part-based method for silhouette representation that decompose silhouette into parts and partitioning lines and describe these elements geometrically. An XML version of the language LWDOS (Language for Writing Descriptors of Outline Shapes) published in [19] is proposed allowing a textual writing of silhouettes description with XML syntax.

Silhouette associated to the image of object is not sufficient in many cases for recognition of object. Internal regions to the main silhouette are fundamental and must be included in shape repre-

sentation. Felzenszwalb and Huttenlocher in [13] consider that an object is modeled by a collection of parts arranged in a deformable configuration. Each part encodes local visual properties of the object, and the deformable configuration is characterized by spring-like connections between certain pairs of parts. This method allows finding features such as eyes, nose and mouth, and the spring-like connections allow for variation the relative location of these features. For people, the parts are the limbs, torso and head, and the spring-like connections allows for articulation at the joints.

A new method for image representation based is proposed in [43] where Tu et al. presented a Bayesian framework for parsing images into their constituent visual patterns. Image parsing is defined as the task of decomposing an image into its constituent visual patterns. The output is represented by a hierarchical graph. The shape is first divided into many regions at a coarse level. These regions considered as shapes are further decomposed into different regions in the second level and so.

Our second contribution in this paper is the proposition of new method for shape representation and description using both the outer contour and the interior regions. The basic principle of image representation of Tu et al. [43] is also used in our method.

We assume that shape may contain internal regions. These regions may also contain internal regions, and so. Shape is decomposed at a coarse level into its constituent internal regions. Each one is considered as a shape and recursively is decomposed into regions in the second level and so. The output is represented by a tree structure translated using a new textual marked language noted XLWDS (XML Language for Writing Descriptors of Shapes) into XML description.

Compared with existing part-based representations, the proposed scheme can provide the following properties:

- Uniqueness, invariance to translation, rotation and scale change.
- Coding of shapes with text following XML language.
- Reduced size of computed description.
- Easiness for index extraction and comparison.
- Preservation of perceptual structure with loss of information.
- Coding and visualization of images for multimedia applications.

This paper is organized as follows: In Section 2 are presented the part-based method for representing objects from their silhouettes and the XML language XLWDOS allowing writing their descriptors. Section 3 introduces a new description method of shapes tacking into account the internal regions and followed by the XML language noted XLWDS (XML Language for Writing Descriptors of Shapes). In Sections 4 and 5 are presented the properties of proposed descriptors and their applications such as shape reconstruction for image visualizing and shape retrieval using shapes database. In Section 6, the results of experiments conducted over artificial and real images are presented and discussed. Finally, Section 7 concludes this paper.

2. A part-based method for outline shapes description

2.1. Basic principle of the method

Let:

- (S) be a silhouette that is assumed extracted from the background of the image.
- (RM) be the rectangle of minimum area encompassing the silhouette (S) (see Fig. 1).
- (OXY) be the coordinates system attached to (RM) so as (O) corresponds to the highest left corner of (RM) , the X -axis and Y -axis are associated, respectively, to the width and the length of (RM) .

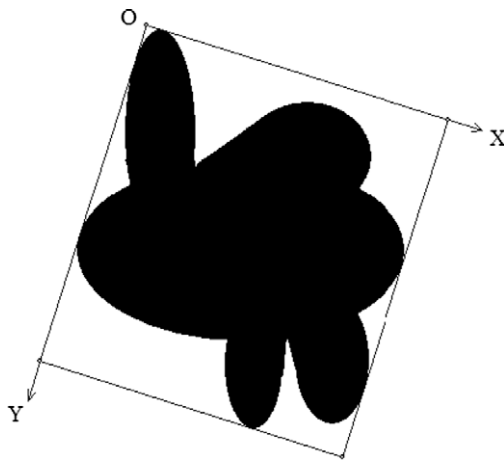


Fig. 1. The rectangle of minimum area encompassing the silhouette.

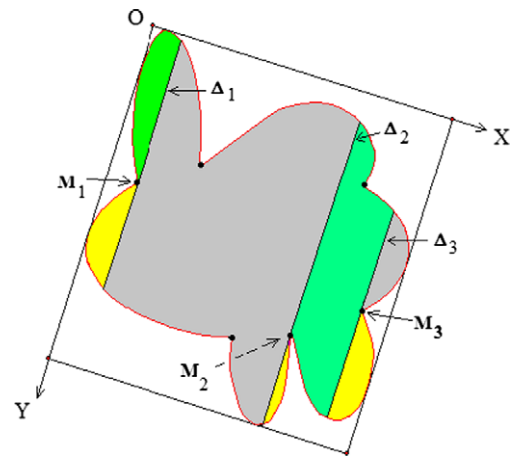


Fig. 3. X-Decomposition of (S).

- (X_Q, Y_Q) be the X- and Y-coordinates of any point Q of the outline contour of the silhouette relatively to the system (OXY).
- f_X and f_Y be two functions so as: $f_X(Q) = X_Q$ and $f_Y(Q) = Y_Q$.

Four kind of concave points may be located onto the contour boundary of any silhouette (S) (see Fig. 2):

- Concave points Q_i so as $f'_X(Q_i) = 0$, these points will be noted M_i .
- Concave points Q_i so as $f'_Y(Q_i) = 0$, these points will be noted N_i .
- Concave points Q_i so as $f'_X(Q_i) = 0$ and $f'_Y(Q_i) = 0$, these points will be noted M_i and N_i .
- Concave points Q_i so as $f'_X(Q_i) \neq 0$ and $f'_Y(Q_i) \neq 0$.

We define:

- The separating lines (Δ_i) as the line passing by the concave point M_i , parallel to the Y-axis, and appertaining only to the silhouette (S).
- The separating line (Δ'_i) as the line passing by the concave point N_i , parallel to the X-axis, and appertaining only to the silhouette (S).

From this definition, the set of lines (Δ_i) decompose the silhouette (S) into parts following the X-direction (see Fig. 3) and the set of lines (Δ'_i) decompose the silhouette (S) into parts following the Y-direction (see Fig. 4).

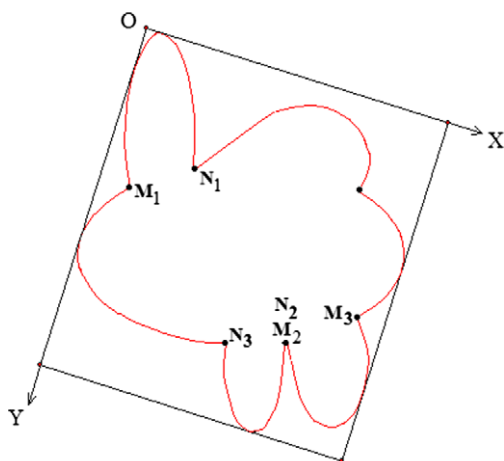


Fig. 2. The set of concave points.

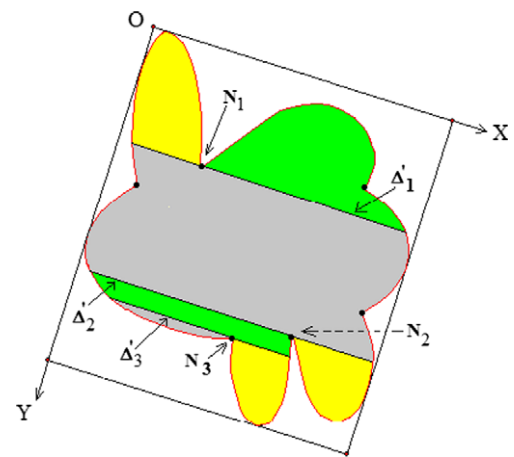


Fig. 4. Y-Decomposition of (S).

When many concave points M_i (or N_i) appertain to the same separating line, the separating process is identical. Fig. 5 illustrates an example of this case.

We call junction line the separating line followed by a single part and preceded by many parts and we call disjunction line the separating line followed by many parts and preceded by a single part.

The result of silhouette decomposition is then a set of parts, junction and disjunction lines. The uniqueness of the rectangle of minimum area encompassing the silhouette implies that the uniqueness of the proposed separating pattern.

However, often silhouettes may have the same decomposition; the difference is then performed using the geometry of external contours of parts and the disposition of parts relatively to the partitioning lines. We show in the next subsection how these elements are described.

2.2. Geometric description of parts

The boundary of any part is decomposed into left and right boundary. Depending on the part position, the two extremities of each boundary are given in Table 1 and illustrated on the example of Fig. 6.

The part is then described by its reference number (apparition order) and the geometry of its two boundaries.

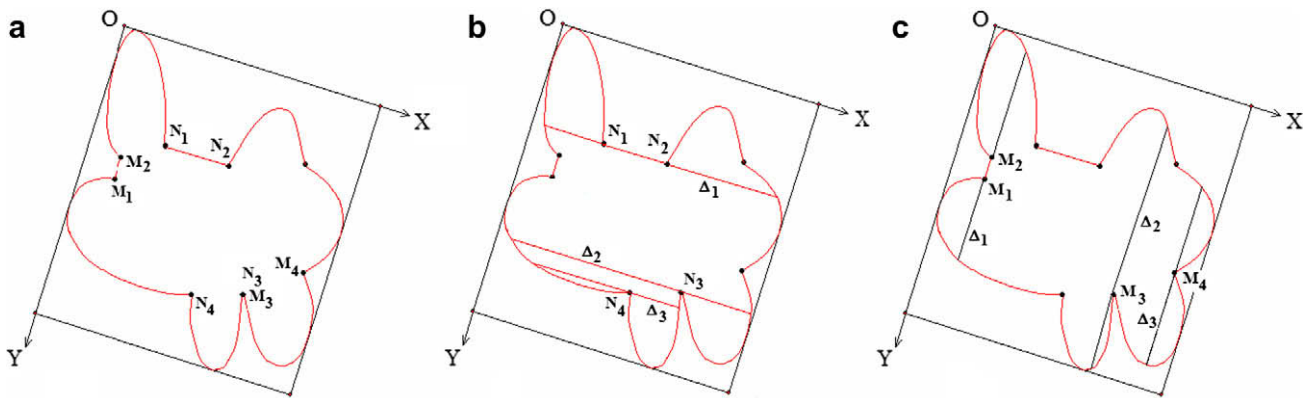


Fig. 5. Case where many concave points appertains to the same separating line (a) the silhouette (S), (b) the X-decomposition of (S), and (c) the Y-decomposition of (S).

Table 1
Extremities of boundaries part.

Position of the part	Left boundary		Right boundary	
	Beginning point	End point	Beginning point	End point
A separating line follows the part	The highest left point	The ultimate point of the left boundary before the separating line	The highest left point	The ultimate point of the right boundary before the separating line
A separating line precedes the part	The first point of the left boundary after the separating line	The Lowest right point of boundary part	The first point of the right boundary after the separating line	The Lowest right point of boundary part
A first separating line precedes the part and a second separating line follows it	The first point of the left boundary after the first separating line	The ultimate point of the left boundary before the second separating line	The first point of the right boundary after the first separating line	The ultimate point of the right boundary before the second separating line

2.2.1. Boundary decomposition into elementary contours

To segment the boundaries of parts into elementary contours, we use the points of high curvature.

Many algorithms have been proposed in the literature. Each one inputs a chain-coded curve that is converted into a connected sequence of grid points $p_i = (x_i, y_i), i = 1, 2, \dots, N$.

A measure of corner strength is assigned to each point, and then corner points are selected based on this measure [21].

The results of boundaries decomposition depends of the selected curvature algorithm.

Chetverikof's algorithm [8] is selected in our implementation because it verifies better the performance evaluation criteria of corner detectors than other algorithms (see Appendices A–C). These criteria are:

- *Selectivity*: The rate of the correct detections should be high; the rate of the wrong ones should be low.
- *Single response*: Each corner should be detected only once.
- *Precision*: The positions of the detected corners should be precise.
- *Robustness to noise*.
- *Easy setting of parameters*: Given a new task, it should be easy to tune the parameters to this task. Ideally, different categories of shapes should be correctly processed with no need to significantly modify the parameters.
- *Robustness to parameters*: Minor changes in parameters should not cause drastic changes in performance.
- *Speed*.

An elementary contour is described by the following parameters (see Fig. 7):

- *Type* that may be: line, convex curve or concave curve.
- *Degree of concavity or convexity* of the curve. The degree of a curve (C) is computed as the ratio of d and the distance of the correspondent chord of (C), where d is the maximum of distances from points on the curve to associated chord
- *Angle of inclination* of the line or the curve. This angle is defined by the line joining the two extremities of elementary contour and the X-axis in case of Y-decomposition or the Y-axis in case of X-decomposition.
- *Length* of the line or the curve computed as the number of rows (or columns in case of horizontal primitive) between its two extremities.

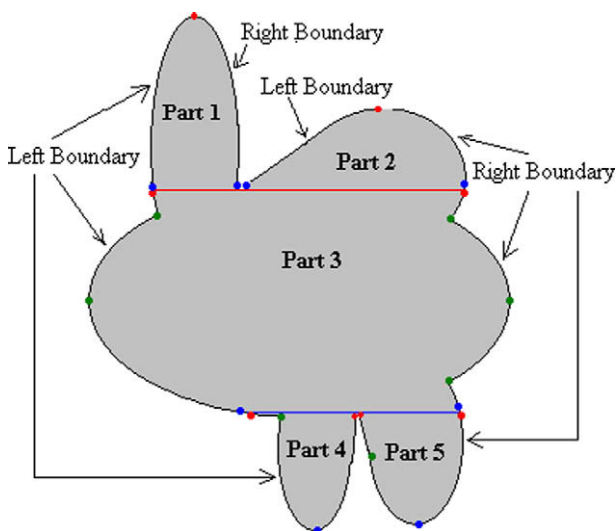


Fig. 6. Boundaries of parts.

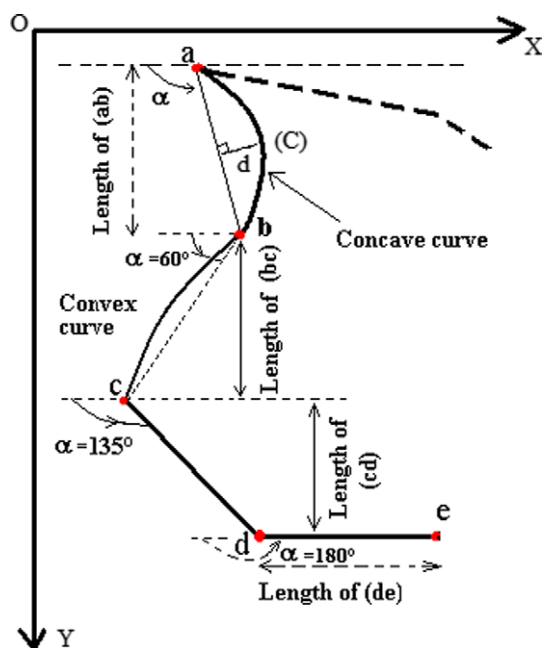


Fig. 7. Description of elementary contours.

The use of these parameters allows a geometric description of the part. The accuracy of this description depends on the good location of curvature points.

2.3. Geometric description of separating lines

In order to describe the position of each part onto the separating line, we decompose this line into segments. Each one of these segments is described using three parameters (see Fig. 7):

- *Type of the segment*: it may be **Shared** if it joins two parts, **Free-High** if it follows only the high part or **Free-Low** if it appears only before the low part.
- *The reference numbers* of parts that it joins.
- *The length*.

Each separating line (junction line (J_i) or disjunction line (D_i)) will have the following description:

$$((\text{Type}, \text{ReferenceNumber1}, \text{ReferenceNumber2}^*, \text{Length})_1, (\text{Type}, \text{ReferenceNumber1}, \text{ReferenceNumber2}^*, \text{Length})_2, \dots, (\text{Type}, \text{ReferenceNumber1}, \text{ReferenceNumber2}^*, \text{Length})_k),$$

where k is the number of segments, **ReferenceNumber2** appears only in case where type of segment is shared.

We describe for example the junction line of Fig. 8 as follows: ((Free-High, Part2, Value1), (Shared, Part2, Part3, Value2), (Shared, Part1, Part3, Value3), (Free-Low, Part3, Value4)), where Value1, Value2, Value3, Value4 are, respectively, the length of the four segments of the junction line.

2.4. An XML language to write the description of outline shapes

In the literature, the use of XML language for coding shapes has been proposed but only for describing particular shapes. In [44], Yu et al. propose a set of coordinates (x, y) written following XML format to represent drawn shapes. The XML language is also used for the description of specific shapes (square, rectangle, etc.) in the SVG format of images.

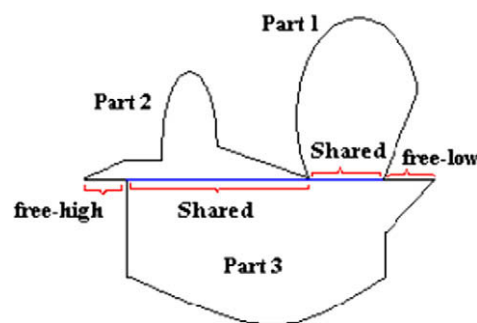


Fig. 8. Description of partitioning line.

The writing with XML language of images descriptions facilitates to database of images the communication with different systems. It permits also the easy extraction of image information from the XML file for data base indexation.

In [19], Larabi et al. have proposed a marked language LWDOS writing descriptors of outline shapes. We present in this section the XML syntax of this language. We give firstly the set of rules allowing the writing of description of parts and separating lines. We introduce after the notion of composed part that allows expressing all components of the silhouette.

2.4.1. Description of silhouette elements

2.4.1.1. *Description of parts*. The part is defined by its reference number and the geometry of elementary contours of the left and right boundaries. Its description is written as follows:

Part \rightarrow **<P PartNumber>** Left boundary Right boundary **</P PartNumber>**
 LeftBoundary \rightarrow **<L>** Contour₁ ... Contour_n **</L>**
 RightBoundary \rightarrow **<R>** Contour₁ ... Contour_m **</R>**
 Contour_i \rightarrow **cv** Convexity-degree Inclination-angle Length/
cc Convexity-degree Inclination-angle Length/**r** Inclination-angle Length

where

- **PartNumber** is a set of characters indicating the reference number of the part.
- **<P PartNumber>**, **</P PartNumber>** are the marks indicating the beginning and the end of the part whose number is PartNumber.
- **<L>**, **</L>** and **<R>**, **</R>** are the marks indicating the beginning and the end of, respectively, the left and right boundary.
- **cv**, **cc**, **r** indicate, respectively, convex, concave, and right contour.

For example, the text: **<P1><L>cc 25 90 40</L> <R>r 180 40 r 90 40</R></P1>** corresponds to the description of part 1 where its left boundary is composed by a concave contour with 25% as degree of convexity, 90° of inclination and 40 pixels as length. Its right boundary is composed by an horizontal contour oriented to right (180° of inclination), 40 pixels as length followed by a vertical contour (90° of inclination) with 40 pixels as length.

2.4.1.2. *Description of separating lines*. The separating line that may be a junction or disjunction line is described as follows:

Junction line \rightarrow **<J JunctionNumber>** Segment₁ ... Segment_n **</J JunctionNumber >**
 Disjunction line \rightarrow **<D DisjunctionNumber >** Segment₁ ... Segment_m **</D DisjunctionNumber >**
 Segment_i \rightarrow **s** High-Part-Number Low-Part-Number Length/
w Low-Part-Number Length/**h** High-Part-Number Length
 where

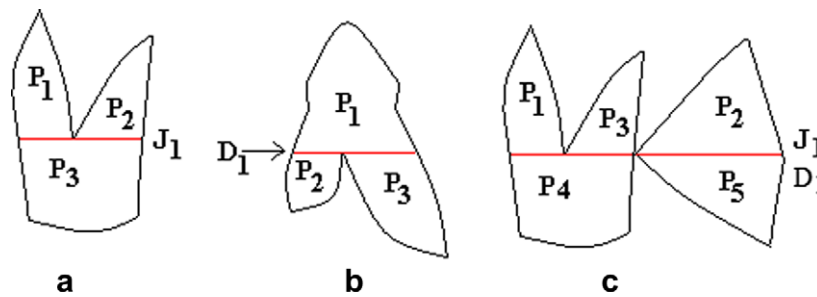


Fig. 9. The possibilities of composed.

- $\langle J \rangle$, $\langle /J \rangle$ and $\langle D \rangle$, $\langle /D \rangle$ are the marks indicating, respectively, the beginning, the end of junction and disjunction line.
- **JunctionNumber** is the order number given to the junction line.
- **DisjunctionNumber** is the order number given to the disjunction line.
- **s, w, h** denote **Shared**, **Free-Low** and **Free-High** attributes for the segments.

For example, the text: $\langle J1 \rangle w P5 40 s P1 P5 40 w P5 40 s P2 P5 40 w P5 40 \langle /J1 \rangle$ corresponds to the description of the first junction line and means that is composed by five segments. The first, the third, and the fifth appertain only to the fifth part and have as length 40 pixels, whereas the second appertains to the first and fifth parts and have 40 pixels as length, the fourth appertains to the second and fifth part with 40 pixels as length.

2.4.1.3. *Description of composed part.* We define a **composed part** as the set of two (or more) parts joined to another part using junction or disjunction line. We write the composed part as follows:

- *Composed Part* $\rightarrow \langle CP \rangle P_1 P_2 \dots P_n J_k P_{n+1} \langle /CP \rangle \langle CP \rangle P_1 D_l P_{i1} P_{i2} \dots P_{in} \langle /CP \rangle$

where

- $\langle CP \rangle$ and $\langle /CP \rangle$ are the marks indicating the beginning and the end of the composed part.
- P_i refers to the description of the part number i .
- J_k and D_l refer to the description of partitioning lines.

The composed part may be also the grouping of parts and composed parts. This allows the writing of all components of the silhouette.

For example, $\langle CP \rangle \langle CP \rangle P1 P2 J1 P3 \langle /CP \rangle P4 J2 P5 \langle /CP \rangle$ is a composed part that groups the composed part $\langle CP \rangle P1 P2 J1 P3 \langle /CP \rangle$, the parts $P4$, $P5$ and the junction line $J2$.

2.4.1.4. *Description of the silhouette.* To write the descriptor of silhouettes we use following syntax:

<DXLWDOS>
<Name>Object name**</Name>**
 Description of the composed part associated to the silhouette
</DXLWDOS>

The grammar of the language XLWDOS is given in [Appendix B](#).

2.4.2. *How to write the descriptor of any outline shape?*

The first step of this process is the writing of each one of separating lines the correspondent composed part which will have one of the following expressions:

- $\langle CP \rangle P_{i1} \dots P_{in} J_i P_1 \langle /CP \rangle$ (see Fig. 9a).
- $\langle CP \rangle P_j D_j P_{j1} \dots P_{jn} \langle /CP \rangle$ (see Fig. 9b).

- $\langle CP \rangle P_{k1} \dots P_{kn} J_k P_\epsilon \langle /CP \rangle$ and $\langle CP \rangle P_\epsilon D_m P_{m1} \dots P_{mn} \langle /CP \rangle$ (see Fig. 9c).

This means that a set of parts $P_{k1} \dots P_{kn}$ are joined through the junction line J_k to many parts $P_{m1} \dots P_{mn}$. The separating line is considered in the two written composed parts as junction and disjunction line (J_k and D_m) and the left and right boundaries of the part (P_ϵ) are empties.

The second step consists to apply a substituting process on all written composed parts. To do this, we define four rules:

Let CP_j ($i = 1, \dots, n$) be a set of written composed parts with junction lines and a CPD_j ($j = 1, \dots, m$) be a set of composed parts with disjunction lines.

- **Rule 1:** If P_i is a main part in composed part CP_j and a secondary part in another composed part CP_k then P_i in CP_k is replaced by the expression of CP_j (see Fig. 10a, where P_i refers to P_4).
- **Rule 2:** If P_i appears as a main part in two composed parts: CP_j and CP_k then P_i in CPD_k is replaced by the expression of CP_j (see Fig. 10b, where P_i refers to P_3).
- **Rule 3:** If P_i appears as a secondary part in CP_j and CPD_k , then P_i in CP_j is replaced by the expression of CPD_k (see Fig. 10c, where P_i refers to P_4).
- **Rule 4:** If P_i appears as a main part in CPD_j and as a secondary part in another CPD_k , then P_i in CPD_k is replaced by the expression of CPD_j (see Fig. 10d, where P_i refers to P_2).

These rules concern all possible combinations, any other combination not may occur.

After the partitioning of the shape of Fig. 11 into parts and separating lines, the outline of each one of the parts are partitioned into elementary contours and their computed description are:

```

<P1> <L> r 33 4 cv 8 80 112 </L> <R> r 171 7 cv 9 102 115 </R>
</P1>
<P2> <L> r 8 7 cv 3 32 53 </L> <R> r 176 18 cv 14 138 35 r 95 10
r 82 8 </R> </P2>
<P3> <L> r 98 14 cv 42 97 121 r 165 10 </L>
<R> r 62 17 cv 36 89 108 r 113 9 r 105 11 </R> </P3>
<P4> <L> cv 8 98 68 r 139 6 r 171 7 </L> <R> r 84 30 cv 6 70 39 r
33 6 </R> </P4>
<P5> <L> r 105 26 r 100 21 r 124 20 r 158 4 r 180 4 </L>
<R> cv 8 86 47 cv 7 50 22 r 18 6 </R> </P5>
    
```

Also, the separating lines are partitioned into segments and their computed description are:

```

<J1> s P1 P3 57 w P3 2 s P2 P3 147 </J1>
<D1> h P3 27 s P3 P4 50 h P3 2 j P3 P5 66 </D1>
    
```

We write all composed part:

```

CPJ1  $\rightarrow \langle CP \rangle P1 P2 J1 P3 \langle /CP \rangle$ 
CPD1  $\rightarrow \langle CP \rangle P3 D1 P4 P5 \langle /CP \rangle$ 
    
```

As, $P3$ is the main part in the two composed parts, we applicate then the rule 2 and we obtain the descriptor of the shape:

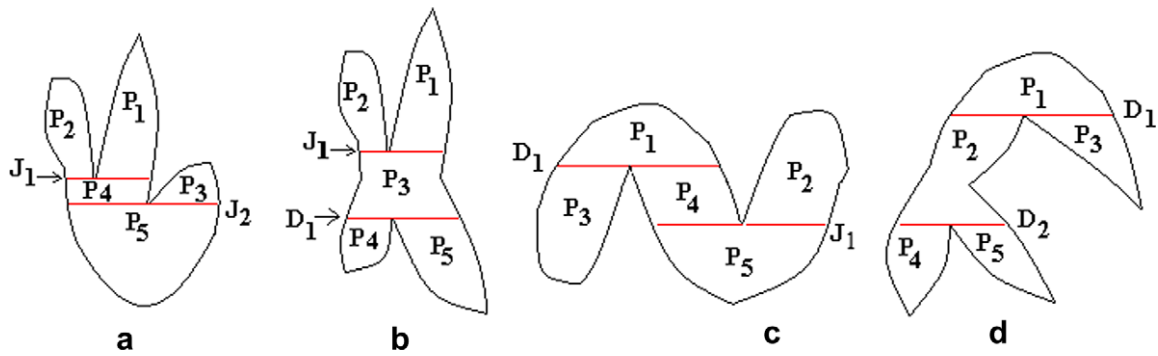


Fig. 10. The four cases of composed part.

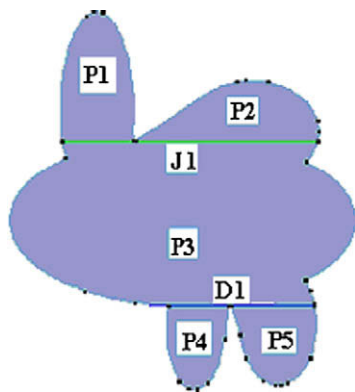


Fig. 11. Result of the decomposition of a silhouette.

```
<DXLWDOS><Name>Silhouette 1</Name>
<CP> <CP> P1 P2 J1 P3 </CP> D1 P4 P5 </CP>
</DXLWDOS>
```

The detailed descriptor is obtained replacing each part and separating line by its description.

3. Description of shapes

3.1. Basic principle of the method

A correspondent shape to a 3D object is generally composed by many regions obtained after image segmentation. In addition to the geometry of its outline, we propose the taking into account of the geometry, the color and the position of its internal regions. This additional information for shape description is very useful

for shape comparison in the recognition process of objects from their 2D images.

We define a shape as the set of internal regions; each one may contain recursively internal regions and so on.

Any shape is then characterized by the list of encompassed regions, the inclusion relation between regions, the position of each region in the image and the description of each region (the geometry of its outline contour and its color).

Having these attributes, we can represent the shape with a tree structure where the node root corresponds to the outline shape, nodes correspond to internal regions and children nodes of any node correspond to the set of encompassed regions (see Fig. 12).

If we consider i as the level of a region which corresponds to its position in the tree beginning from the bottom ($i = 0$) to the up, and j as the order number of regions of same level sweeping top-bottom and left-right the image, any region of the shape will be note S_{ij} .

Thus, the level zero (0) is then given to regions that not contain internal regions. In general case, the level (n) is associated to regions encompassing regions of level less or equal to ($n - 1$). For example, the image of Fig. 12 contains for example four regions of level 0 ($S_{0,0}, S_{0,1}, S_{0,2}, S_{0,3}$) and two regions of level 1 ($S_{1,0}, S_{1,1}$) encompassing $S_{0,0}, S_{0,1}$ and one region $S_{2,0}$ of level 2 containing $S_{1,0}, S_{1,1}, S_{0,2}$ and $S_{0,3}$.

The tree structure can be automatically achieved starting from a segmented image, by the location on the image all elementary regions which not encompass any internal regions; the level zero is associated for them. The next step is the location of regions of level 1 which encompass regions of level 0. We repeat this treatment in order to locate regions of level i which correspond to regions that encompass regions of level less or equal to ($i - 1$). An order number is given for regions of the same level starting from zero by the sweep of the image from the top to bottom and left to the right.

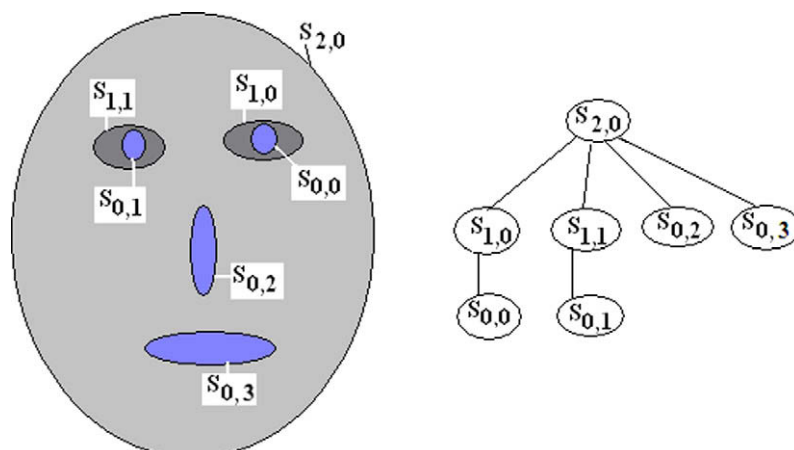


Fig. 12. Shape parsing into internal regions and the associated tree structure.

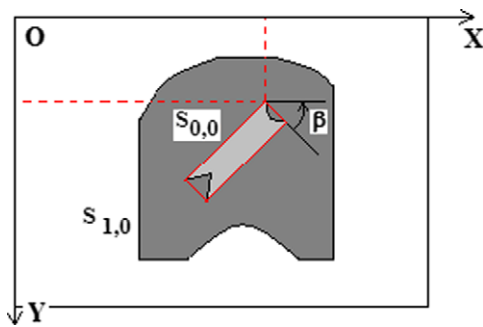


Fig. 13. Initial shape and the minimum rectangle of internal region.

In case where regions of level less or equal to $(k - 1)$ are neighboring one to other, a new region of level (k) is created and added in the tree as the union of these regions.

At each node of the tree structure we must add a set of attributes describing the geometry of its outline, the position of highest left corner of the rectangle of minimum area (RM) encompassing it and its orientation computed as the inclination angle β aligning its width with image rows (see Fig. 13).

The following algorithm summarizes all steps of shape decomposition and representation:

Algorithm 1.

```

Begin
1. Locate all regions of the shape
2. For Each one elementary region  $S_{i,j}$  (which not encompasses any region other)
Do
- Compute the associated minimum rectangle (RM)
- Compute the textual descriptor
- Compute the average color of the region
- Calculate the X- and Y-coordinates of the highest left corner of (RM)
- Calculate the rotation angle  $\beta$  aligning the width of (RM) with image rows
-  $i = 0$ 
EndDo
3.  $i = 1$ 
Repeat
For Each one region encompassing regions whose level is less than Current level ( $i$ )
Do
- Compute the associated minimum rectangle (RM)
- Compute the XLWDOS descriptor
- Calculate the X- and Y-coordinates of the highest left corner of (RM)
- Calculate the rotation angle  $\beta$  aligning the width of (RM) with image rows
- RegionLevel =  $i$ 
EndDo
 $i = i + 1$ 
Until all regions should be processed
End
    
```

3.2. A text language for writing descriptors of shapes

An XML language noted XLWDS (XML Language for Writing Descriptors of Shapes) is proposed in this paper for writing descriptors of shapes.

Shape may be a region without internal silhouettes or contains at different levels internal silhouettes. The following syntax is used to describe recursively this inclusion where the marks **<Internal Shape></Internal Shape>** are used in order to cite internal silhouettes.

```

<Shape>
  Position of the highest left corner of the minimum rectangle encompassing the outline shape
  Orientation of the associated minimum rectangle
  Color of region
  XLWDOS Descriptor of the outline shape
<Internal Shape>
  Position of the highest left corner of associated minimum rectangle
  Orientation of the associated minimum rectangle
  XLWDOS Description of region 0
  Color of the region 0
</Internal Shape>
<Internal Shape> ...
</Internal Shape>
</Shape>
    
```

In addition to the rules of the language XLWDOS, a set of rules are added to express the inclusion of regions. The grammar of the XLWDS language is given in Appendix C.

4. Shape descriptor properties

The method presented in this paper for shape description verifies the following properties:

4.1. Uniqueness and preservation of perceptual structure

In the proposed method for outline shape description, all elementary contours and positions of parts on separating lines are geometrically well estimated. The reconstruction of the silhouette is then the inverse process and it accuracy depends on the accuracy of the boundary description.

We demonstrate in follow how the outline is reconstructed from its description without ambiguity.

Let $(a_i, b_i, \dots, x_i, y_i, z_i)$ and $(a'_i, b'_i, \dots, x'_i, y'_i, z'_i)$ be the set of curvature points located, respectively, on the left and right boundary of the part P_i which begin by the points (a_i, a_i) and terminate by the points (z_i, z'_i) .

Fixing the position of one extremity of left boundary, the end point (z_i) for example, the contour $(z_i y_i)$ is drawn knowing its description (length, inclination angle and concavity or convexity degree in case of curve contour). The neighbor contour $(y_i x_i)$ is drawn knowing its description and the position of (y_i) . The repetition of this process allows the drawing of all elementary contours. The same principle is used to draw the right boundary by fixing the position of one extremity.

When the two boundaries of the (RM) of the part have one common extremity, it will be used as starting point and the drawn part corresponds exactly to the described part (see Fig. 14).

However, when there is not common extremity, this part is then neighbor to two separating lines. In this case, these extremities are positioned after the positioning of the separating line. Fig. 14 illustrates an example of reconstruction process of parts and partitioning lines of the described outline shape. There are two parts of case 2, two parts of case 3 and one part of case 4.

Once the parts are drawn, they will be positioned exactly on the separating line using the length of each segment and the parts that are linked.

4.2. Invariance to rotation

The result of the decomposition process described above depends on the orientation of the silhouette in the image. The minimum bounding box is firstly computed. A rotation of the box is performed with an angle β around the highest corner point M so

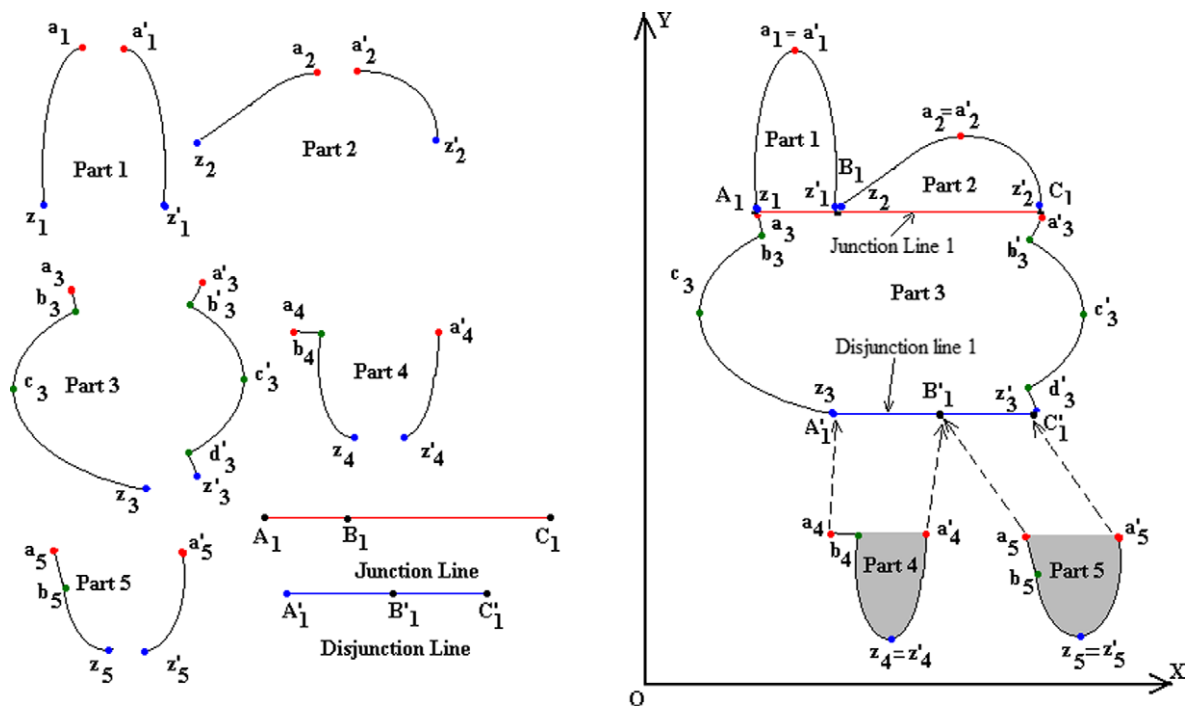


Fig. 14. Different steps of the drawing process.

as its width coincides with the rows. The coordinates system OXY is then attached to the box so as OX coincides with rows and OY with columns (see Fig. 15).

The sweeping of the silhouette relatively to one direction of the rectangle of minimum area encompassing it guarantees the invariance to rotation of this description because of the uniqueness of this rectangle.

4.3. Invariance to scale change

To guaranty the property of scalability, the lengths in outline shape description are relative and computed relatively to the length of the minimum bounding box.

4.4. Easiness for indexation

The outline shape descriptor is textual information structured into elements (parts, junction and disjunction lines) and relations between these elements. From the XML descriptor it is easy to extract much information so as the number of parts, the numbers

of junction and disjunction lines, the number of parts neighboring to each one separating line, etc.

We give in shape comparison subsection how are used these elements for the computation of the index which permits to select a group of shapes from the database of shape models.

4.5. Computation of outline shape descriptor in the multi-scale space

Over the last few years, multi-scale approaches in image analysis have proved to be useful in terms of describing images at varying levels of resolution [17]. Scale-space theory, as a relatively new field, has been established as a well founded, general and promising multi-resolution technique for image structure analysis, both for 2D, 3D and time series [33]. It provides more information about the object, increases discrimination power and immunity to noise [12]. The underlying image can be represented by a family of images on various levels of inner spatial scale. These are obtained by convolution with the Gaussian kernel as the lowest order, rescaling operator, and its linear partial derivatives. A continuous scale space is constructed to enable local image analysis in a robust

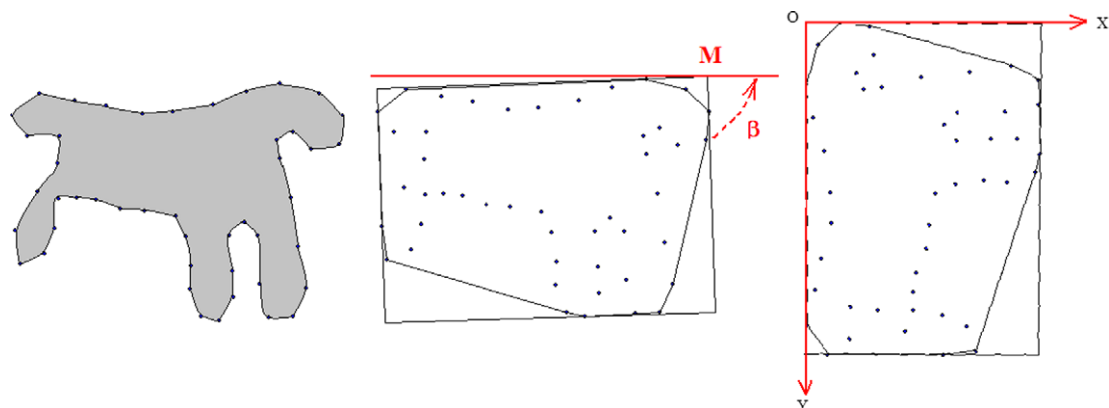


Fig. 15. Outline shape (left), the convex hull and the minimum bounding box (center), rotated bounding box (right).

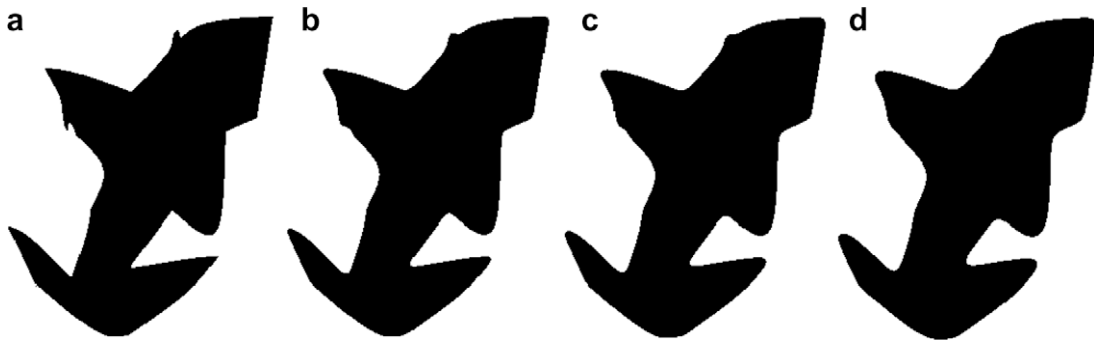


Fig. 16. (a) Initial silhouette, (b) the smoothed silhouette with $\sigma = 14.94$, (c) with $\sigma = 29.88$, and (d) with $\sigma = 59.76$.

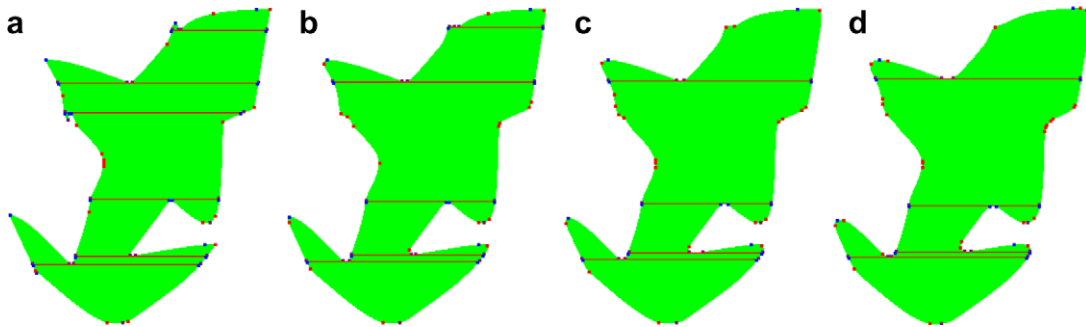


Fig. 17. (a) Result of silhouette decomposition of initial silhouette, (b) of smoothed silhouette with $\sigma = 14.94$, (c) with $\sigma = 29.88$, and (d) with $\sigma = 59.76$.

way, while at the same time global features are captured through the extra scale degree of freedom [37].

Curvature Scale Space is a technique where shape is represented along a range of scales spanning from coarse to fine. If a boundary exhibits small false oscillations, then excessive curvature points will be found during separating process. It was selected as a contour shape descriptor for MPEG-7 after substantial and comprehensive testing, which demonstrated the superior performance of the CSS-based descriptor [24].

Our proposed descriptor for outline shapes may be computed in the multiscale space. Descriptors of silhouette at different scales may be computed varying the value of σ (Gaussian filter). At the coarse scale, there are numerous curvature points, separating lines and parts; this number is low at the low scale. Figs. 16 and 17 illustrate an example of silhouette and the result of its decomposition after it's smoothing with different values of σ and its thresholding by ISODATA technique [32]. We can see that in low scale (high value of σ) the number of parts and curvature points decreases considerably.

4.6. Facility in shape recognition

The presence of descriptors of internal regions in the structured shape descriptor allows having more information for shape indexing and comparison.

5. Applications

5.1. Image coding and visualization

In this subsection we explain how the proposed descriptor of shape is decoded in order to visualize the correspondent image. We discuss also the quality of reconstructed image.

5.1.1. Outline shape reconstruction from XLWDOS descriptor

A referential system (OXY) is chosen so as the OX -axis coincides with the rows, OY -axis coincides with the columns and the origin O coincides with the left extremity of any one of separating lines of the outlines shape (see Fig. 18).

Knowing the description of this line, the X -coordinates of extremities of its segments which correspond to those of neighboring parts boundaries will be computed.

A propagation process is then performed which consists to calculate the positions of all curvature points of boundaries of linked parts from their description. This allows obtaining the position of extremities of segments of others separating lines.

This process is repeated recursively and allows reconstructing the outline entirely.

Fig. 18 illustrates this process where the first step is the use of the first separating line for the positioning of the extremities m_1, m_2, m_3, m_4 and m_5 . The second step is the positioning of extremities of elementary contours of parts P_1, P_2 and P_3 . The outline of these parts is then drawn. As the part P_3 is linked to disjunction line and

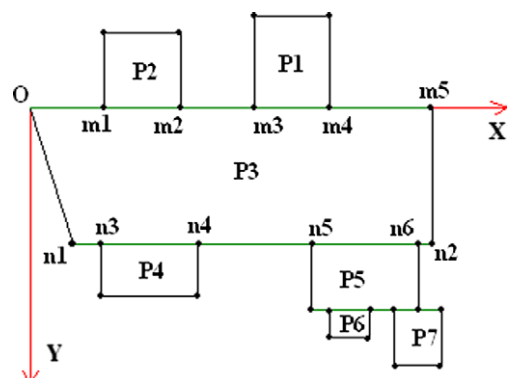


Fig. 18. Outline shape reconstruction.

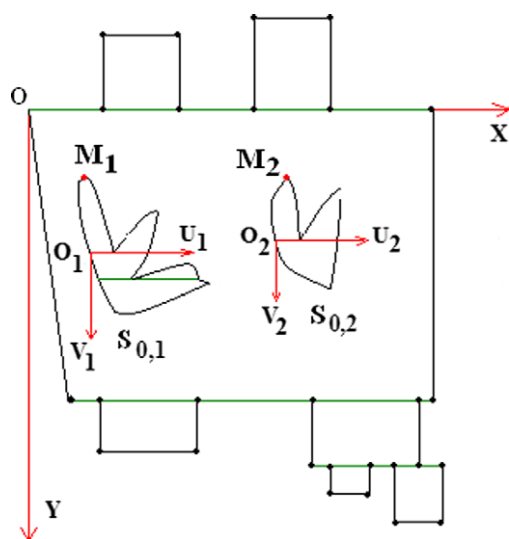


Fig. 19. Shape reconstruction.

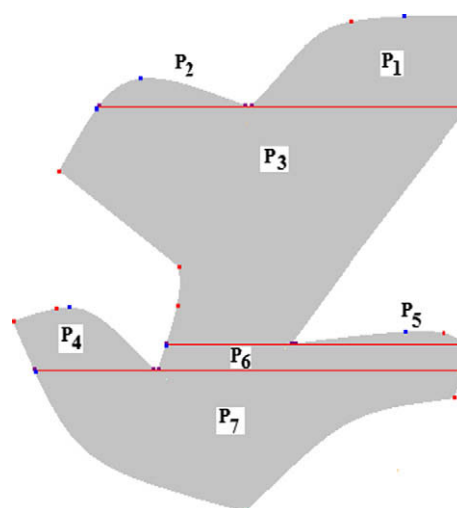


Fig. 20. Example of outline shape.

the positions of the end points n_1 and n_2 of its boundaries are known, the next step is the use of XML description of this separating line for the computation of the position of n_3, n_4, n_5 and n_6 which correspond to the beginning points of P_4 and P_5 boundaries. We can then draw the boundaries of these two parts, and so on.

In case where the shape descriptor contains only one part, the highest left point of this part is considered in this case as the origin O of the system (OXY). The coordinates of extremities of all elementary contours relatively to the starting point are extracted from the descriptor and therefore all elementary contours are drawn.

5.1.2. Shape reconstruction from XLWDS descriptor

The system OXY is fixed as described in Section 5.1.1 relatively to more external outline shape. Each one of internal outline shapes is reconstructed and the use the position of their highest left points extracted from the correspondent descriptions permit to position them relatively to (OXY) system.

Fig. 19 illustrates an example of shape reconstruction where the external outline shape is firstly reconstructed and internal outlines $S_{0,1}$ and $S_{0,2}$ are reconstructed relatively to the systems $O_1U_1V_1$ and $O_2U_2V_2$. The knowing of the coordinates of points M_1 and M_2 of these two regions permits to compute the new coordinates of all points relatively to OXY system.

5.1.3. Quality Assessment of the Drawn Shape

The quality of the drawn shape depends on the quality of boundaries parts description.

If the curvature points are well located, the elementary contours should be well described, and the reconstruction process should draw a similar contour.

Knowing that the separating lines of the two shapes (initial and reconstructed) are identical, the quantification of this quality of drawn shape is measured as the sum of rates of missing and additional pixels computed after the superposition of these two shapes through one of the separating. We note that any pair of correspondent separating line should be well superposed. We will give in experimentation examples of shape reconstruction and their quality.

5.2. Shape comparison

The high quality of any shape descriptor is measured by its performance to resolve the shape comparison problem (algorithm complexity, space memory, similarity measure).

We explain in this subsection firstly how to index shape descriptors and we propose after a similarity measure for outline shapes and shapes comparison.

5.2.1. Indexing of Outline Shape descriptor

The index of outline shape descriptor (OS) query or model is defined by:

- Number of parts (N).
- Types of separating lines: 0 for junction line and 1 for disjunction line.
- Attributes of segments of separating lines.
- Number of parts joined to each one of separating line.

This information is easily extracted from the textual descriptor and will be written:

$$I(D_{LWDSOS}(OS)) = N((0/1)(j/w/h)^+ n_i)^+$$

As example, the index of the textual descriptor of the shape illustrated by Fig. 20 is (7 0 j w j 3 0 3 j j 0 3 j w j) which means that there seven parts and three separating lines (junctions), each one is joined with three parts.

When two shapes have the same index; the difference between them should be computed using the geometry of their parts outlines and their disposition relatively to different separating lines.

5.2.2. Indexing of Shape descriptor

Regions of shape may be represented by a tree structure where the root corresponds to the level value of more external outline shape; nodes correspond to the region levels, and arcs indicate the inclusion of regions. This structured information is extracted from the descriptor and constitutes the first part of shape index which is obtained reading the values of the tree in depth. The second part of the shape descriptor index is a set of pointers to indexes of outlines shapes descriptors which contains.

For example, the index of shape descriptor of Fig. 21 is: 3210010010 @₁ . . . @₁₀ indicating that the more internal region is of level 3 which contains three internal regions:

- Region of level 2 which contains:
- Region of level 1, this region contains two regions of level 0.
- Region of level 1, this region contains two regions of level 0.
- Region of level 1, this region contains one region of level 0.

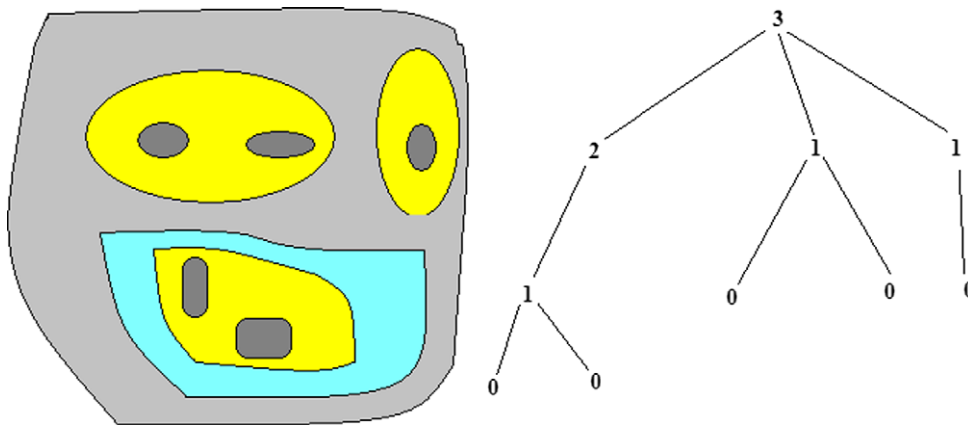


Fig. 21. Example of shape and the associated tree.

The indexes of outline shapes are indicated in the order after this information.

5.2.3. Similarity measure between outline shapes

A necessary condition so that outline shape query matches one of outline shape models is that their textual descriptors must have the same index.

In real applications, outline of extracted shapes may be different of outline shapes models but their appearances are similar. These shapes may have different indexes due to some additional or missing pixels on the outline shape query (see Fig. 22).

To retrieve outline shape query in the database of outline shapes models, we propose the smoothing of the outline with Gaussian filter using different values of σ and its thresholding with ISODATA clustering algorithm [32]. This process should produce outlines shapes at different scales. For each one of these new outlines, if the index of textual descriptor corresponds to one of the database models, the comparison of boundaries parts and separating lines should be done.

The similarity between outline shapes is measured by three factors. The first and second ones measure the distortion between the boundaries of parts, while the third one measures the positioning of parts on the separating lines.

Let $(P_Q), (P_M)$ two parts of the shape query and shape model (see Fig. 23). Similarity between $(P_Q), (P_M)$ is measured in relation to the geometry of their two boundaries.

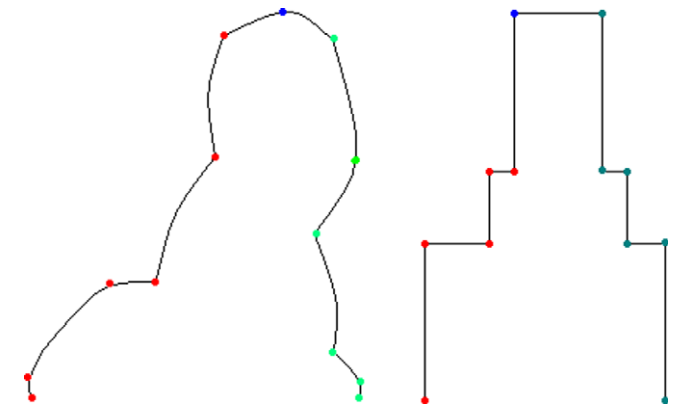


Fig. 23. Two parts: query and model.

The first factor of similarity measures the difference between the areas $A(P_Q), A(P_M)$ of the two parts. To do this, all elementary contours of are considered as segment lines and $A(P_Q)$ and $A(P_M)$ are estimated as the sum of areas of all trapezoids and triangles as illustrated by Fig. 24.

Let (OXY) be a system fixed so as the origin O coincides with the starting point C_Q^1 of the two boundaries of the part query. Using the

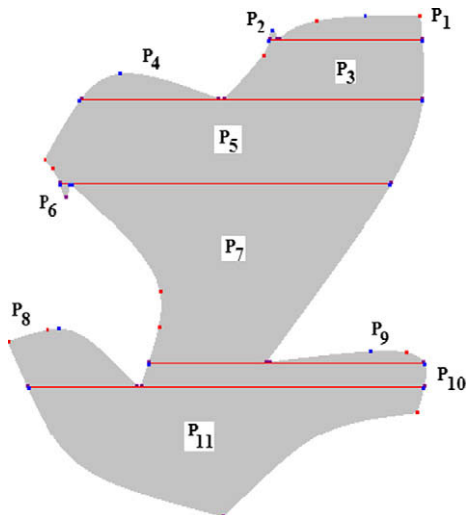


Fig. 22. Two additional parts (P_2 and P_6) dues to the outline distortion.

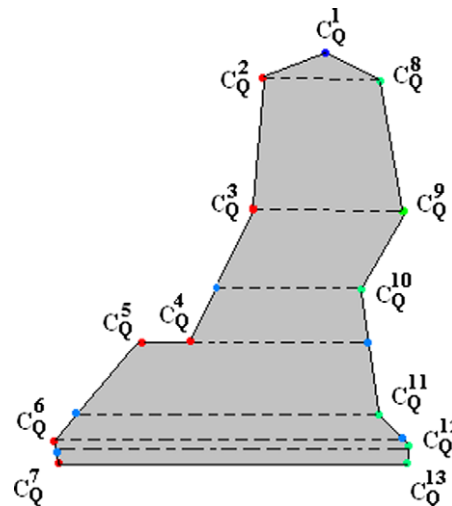


Fig. 24. All trapezoids of the part query for area estimation.

part description, the 2D coordinates of curvature points C_Q^i of the part P_Q are computed relatively to (OXY) .

In the same way is estimated the area $A(P_M)$ of model part. The accuracy of the computed area depends on the convexity and concavity degree of elementary contours. The parameters of the used algorithm for locating curvature points must be chosen so as the degree of concavity and convexity is always low.

We define the first similarity factor SF_1 between parts as:

$$SF_1(OS^Q, OS^M) = \frac{1}{NP} \sum_{i=1}^{NP} \left(\left| \frac{A(P_i^Q)}{A(MR^Q)} - \frac{A(P_i^M)}{A(MR^M)} \right| \right)$$

where $A(MR^Q)$, $A(MR^M)$ are, respectively, the areas of the minimum rectangle encompassing the outline shape query and model, and NP is their number of parts.

The second one factor measures the difference between the geometry of boundaries parts query and model. We define this similarity as follows:

$$SF_2(OS^Q, OS^M) = \frac{1}{NP} \sum_{i=1}^{NP} \left(\frac{1}{NC_i} \sum_{j=1}^{NC_i} (\Delta T_j + \Delta D_j / 180) \right)$$

where ΔD_j is the difference (in degree) of the orientation of compared contours number (j) ; $\Delta T_j = 0$ if the two compared contours have the same type and $\Delta T_j = 1$ otherwise; NC_i is the number of elementary contours of the part number i ; NP is the number of parts.

As the two outline shapes query and model have the same index, there is a correspondence between segments of separating lines of outline shapes query and model. The third similarity factor is measured as the sum of difference between the relative lengths of matched segments of all pairs of separating lines. The length of each one segment is computed relatively to the length of its separating line.

We define the third similarity factor as:

$$SF_3(OS^Q, OS^M) = \frac{1}{NSL} \left(\sum_{j=1}^{NSL} \sum_{k=1}^{NSeg_j} \left(\left| \frac{s_{j,k}^Q}{L_j^Q} - \frac{s_{j,k}^M}{L_j^M} \right| \right) \right)$$

where $NSeg_j$ is the number of segments of the separating line number j ; NSL is the number of separating lines; $s_{j,k}^Q, s_{j,k}^M$ are the segments number k of the separating line j , respectively, of query and model outline shape; L_j^Q, L_j^M are, respectively, the lengths of separating lines number j of outline shapes query and model.

When segments of matched separating lines have similar relative lengths, the value of SF_3 decreases towards zero. For example, the similarity factor is equal to 0.311 for the first outline shape query and 0.179 for the second one (see Fig. 25).

The similarity measure SM of two outline shapes is then computed as the sum of the three factors:

$$SM(OS^Q, OS^M) = \sum_{i=1}^3 SF_i(OS^Q, OS^M)$$

5.2.4. Similarity measure of shapes

Assuming that the shapes to be compared have the same index, the similarity measure is then computed as the sum of similarity measures between correspondent outline shapes correlated by a factor which indicates the similarity between their positions.

Let $(X_Q, Y_Q), (X_M, Y_M)$ be the position of the highest left corners of the minimum rectangles encompassing them whose lengths are L_Q and L_M , the similarity between the positions of two matched outline shapes (SP) is measured as:

$$SP(OS_i^Q, OS_i^M) = |(X_Q/L_Q) - (X_M/L_M)| + |(Y_Q/L_Q) - (Y_M/L_M)|$$

$$SM(S^Q, S^M) = \frac{1}{NOS} \sum_{i=1}^{NOS} SM(OS_i^Q, OS_i^M) \times SP(OS_i^Q, OS_i^M)$$

where NOS is the number of outline shapes.

6. Experimentation

6.1. Computation of XLWDOS descriptors

Once silhouettes are extracted from image using image one of segmentation techniques [11,13] or adaptive background modeling using a Gaussian Mixture Model (GMM) in case of moving object [42], the following steps are performed for each one:

- Computation of the rectangle of minimum area (RM) encompassing it.
- Rotation of the silhouette around the highest left corner of (RM) in order to align the width or the length of (RM) with image rows. The first rotation of (RM) with angle β_w aligning its width, allows the sweep of the silhouette in the length direction of (RM). The second rotation with angle β_l aligning its length, allows the sweep of the silhouette in the width direction of (RM).
- Decomposition of rotated silhouette into parts, separating lines and the boundaries of all parts into elementary contours by means of curvature points.
- Computation of attributes of different contours and the writing of the textual descriptor.

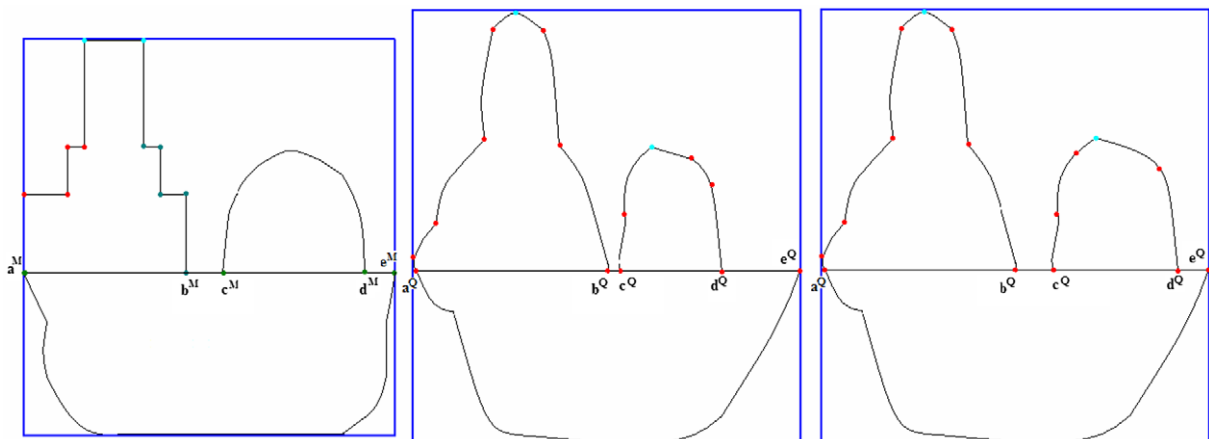


Fig. 25. (left) Outline shape model, (center) first outline shape query (right) the second one.

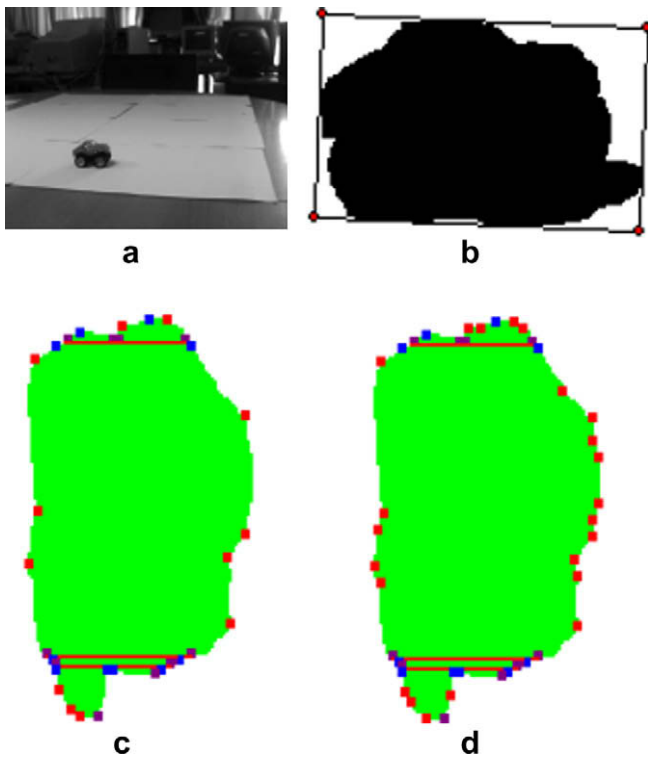


Fig. 26. Image of moving car (a), the minimum rectangle associated for the extracted silhouette (b), after rotation, its decomposition for $d = 5$ pixels, $\alpha = 150^\circ$ (c), and for $d = 5$ pixels, $\alpha = 160^\circ$ (d).

Fig. 26 illustrates image of a moving car (toy) and the rectangle of minimum area (RM) encompassing the extracted silhouette. A rotation of the silhouette around the highest left corner of (RM) is performed with $\beta = +84.67$ (Fig. 26b). Fig. 26c and 26d illustrate the results of its decomposition using different values of the parameters α and d of Chetverikov's algorithm [8]. The size of computed textual descriptors is 673 bytes for $\alpha = 150^\circ$ and 745 bytes for $\alpha = 160^\circ$.

Fig. 27 illustrates the extracted silhouettes using JSEG method for image segmentation [11] as well as the obtained results applying the same steps described above for the computation of descriptors where the values 5, 160° for d and α are used for the detection of curvature points.

Fig. 28 illustrates the result of the segmentation of image of a hand applying the method of FELZENSZWALB and HUTTENLOCHER [13]. The more internal region is selected and the result of its decomposition into parts is illustrated by Fig. 29.

As the outline is much distorted, the computation of the descriptor at different scale is very useful for the comparison process. Indeed, there are many curvature points and many elements (parts, separating lines) not useful for shapes comparison because they are generated by the distortion of the outline.

The smoothing with a Gaussian at different scales is then very important. We illustrate by Fig. 30 the result obtained where the number of parts is 12 parts. This number not changes if the value σ of the Gaussian filter increases.

6.2. Computation of XLWDS descriptors

To locate regions of shapes and all internal regions, we used the method (JSEG) [11]. Once the image is segmented, the following steps are performed:



Fig. 27. Different steps applied for the computation of XLWDS descriptors for the two extracted silhouettes (glasses and scissors).

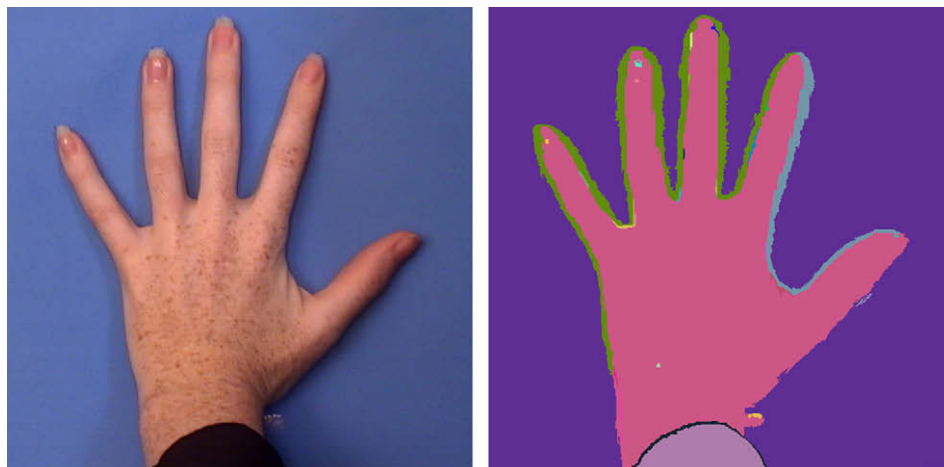


Fig. 28. Image of a hand and its segmentation.

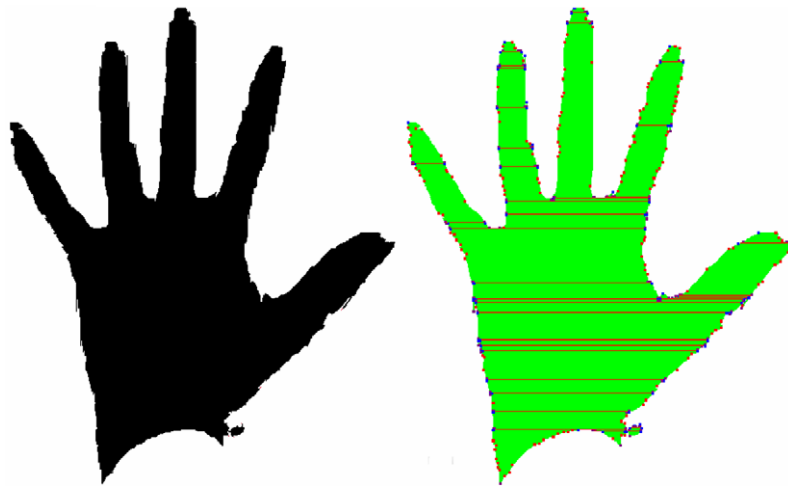


Fig. 29. The more internal region and its decomposition into parts.

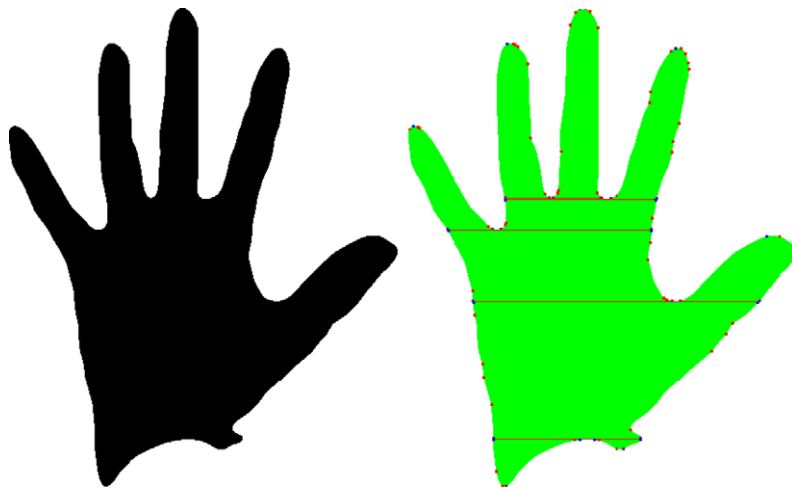


Fig. 30. The smoothed more internal region and the result of its decomposition.

- For each one region, computation of the level value and the order number obtained sweeping image in the direction top-bottom and left-right.
- A tree structure is created for the representation of different regions.
- For each region, the rectangle of minimum area encompassing it (RM) is computed so as the X- and Y-coordinates of the highest left corner, and the rotation angle β which permit the alignment of its width with the rows of the image.
- Computation of the XLWDOS descriptor of rotated region with the angle β .
- Writing of all computed data of the tree following the syntax of XLWDS language.

Fig. 31 illustrates an example of drawn 2D shape with internal regions and the segmented image using the JSEG method [11]. Software is developed which compute and visualizes:

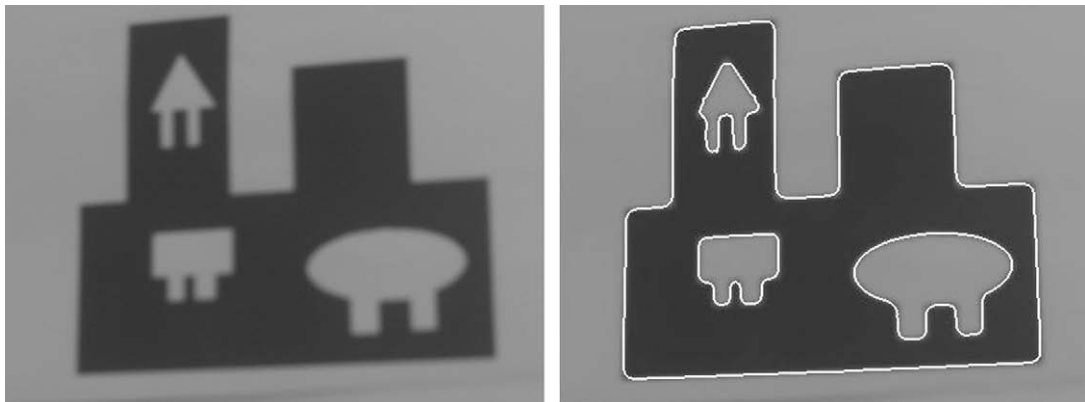


Fig. 31. Example of shape, and the result of its segmentation.

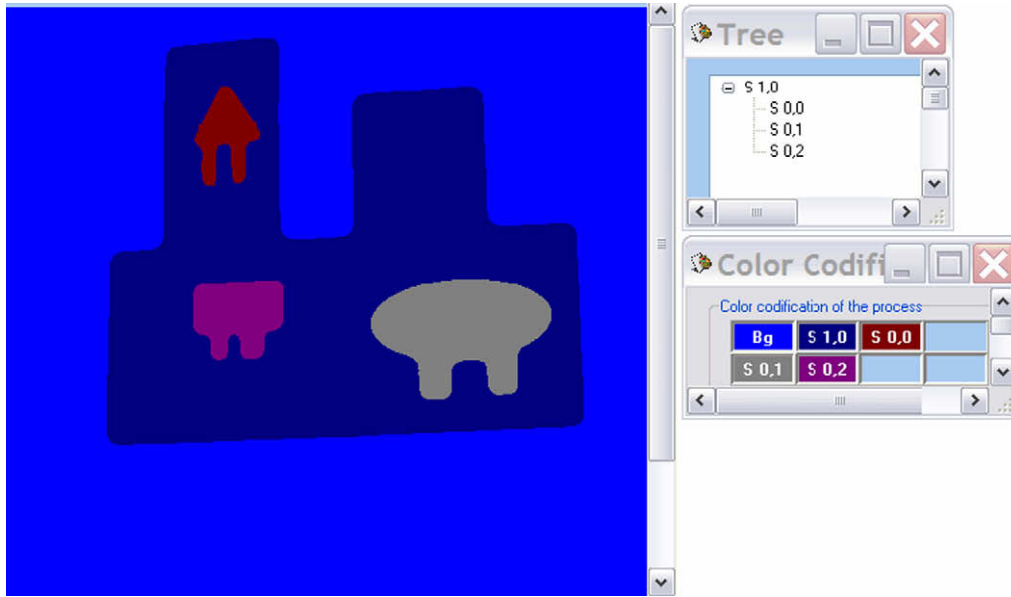


Fig. 32. Result of XLWDS descriptor computation for the drawn shape.

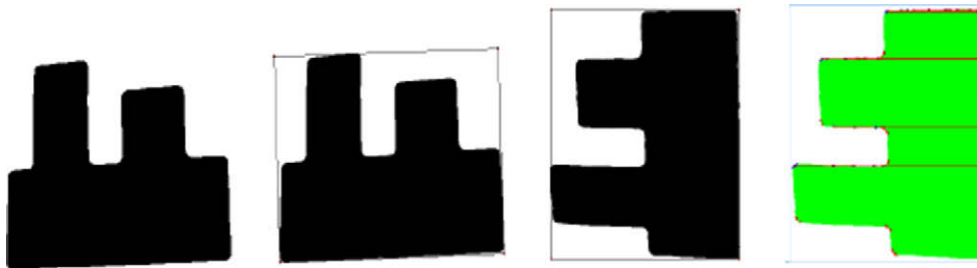


Fig. 33. For the region $S_{1,0}$, the minimum rectangle, rotated region, the result of XLWDS description.

- The different regions using different colors (the background (Bg) is also colored).
- The codification of different colors. For two neighboring regions, their union is considered as a new region and the black color is associated.
- The correspondent tree structure.
- The XLWDS descriptor of each region using the two parameters of Chetverikov's algorithm [8]: the angle α and the distance d .
- The XLWDS descriptor of the shape.

The XLWDS descriptors are computed for all rotated regions and will constitutes the XML descriptor of the shape.

Fig. 32 illustrates $S_{1,0}$ of level 0 as example of regions, the minimum rectangle (RM) encompassing it and the rotated region and the result of its decomposition Fig. 33.

We give in the next the global structure of XLWDS descriptor where each one of the textual descriptors of regions is preceded

by the coordinates of highest left corner of the minimum rectangle encompassing it, the amount of its rotation and the color of the region (three values).

```

<DXLWDS>
<Name> im1 </Name>
<Shape> <Name> S1,0 </Name> 447 16 -88.28 0 0 128 XLWDS
descriptor of S1,0
<Internal Shape> <Name> S0,0 </Name> 181 51 -19.23 128 0 0
XLWDS descriptor of S0,0 </Internal Shape>
<Internal Shape> <Name> S0,1 </Name> 429 198 -86.10 128
128 128 XLWDS descriptor of S0,1 </Internal Shape>
<Internal Shape> <Name> S0,2 </Name> 232 200 -88.02 128 0
128 XLWDS descriptor of S0,2 </Internal Shape>
</Shape>
</DXLWDS>
    
```

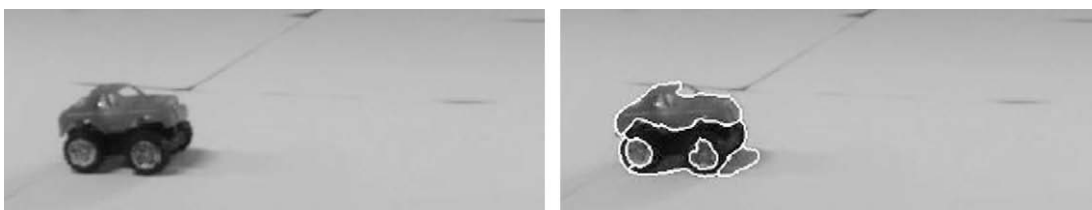


Fig. 34. Image of moving car and the segmented image.

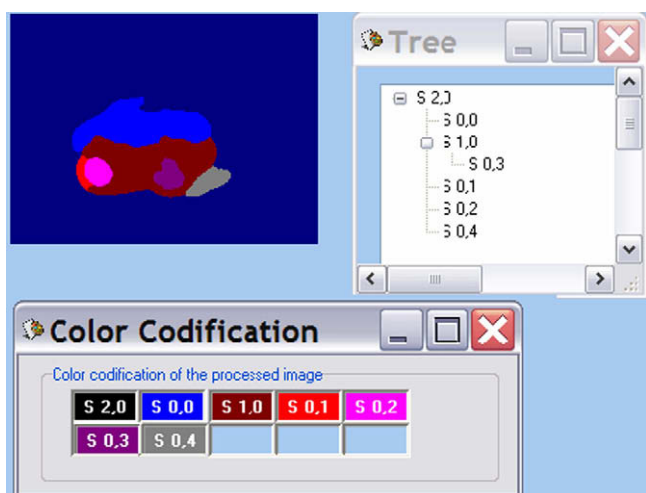


Fig. 35. Results of XLWDS descriptor computation for the car image.

Fig. 34 illustrates image of moving car (toy) and the correspondent shape is extracted using background subtraction and JSEG method [11] for its segmentation. Fig. 35 illustrates the representation of all regions with a tree structure. The shape $S_{2,0}$ of level two is constituted by three regions of level zero ($S_{0,0}, S_{0,1}, S_{0,2}, S_{0,4}$) and a region of level one ($S_{1,0}$) encompassing the region ($S_{0,3}$) of level zero. The global XLWDS descriptor is written as follow where to each one region the X- and Y-coordinates of highest point of the minimum rectangle encompassing it and the rotation angle performed in order to align it with the row of images.



Fig. 36. Image of a logo and the result of its segmentation.

As the region $S_{2,0}$ is the union of many regions, the black color is associated.

```
<DXLWDS>
<Shape> <Name> S2,0 </Name> 46 57 87.58 0 0 0 XLWDS
  descriptor of S2,0
<Internal Shape> <Name> S0,0 </Name> 141 55 -85.10 0 0
  255XLWDS descriptor of S0,0
</Internal Shape>
<Internal Shape>
<Name> S1,0 </Name> 52 87 90 128 0 0 XLWDS descriptor of
  S1,0
<Internal Shape> <Name> S0,3 </Name>111 102 57.09 128 0
  128 XLWDS descriptor of S0,3
</Internal Shape>
</Internal Shape>
<Internal Shape>
<Name> S0,1 </Name> 56 101 6.34 255 0 0 XLWDS descriptor
  of S0,1
</Internal Shape>
<Internal Shape>
<Name> S0,2 </Name> 62 100 42.71 255 0 255 XLWDS
  descriptor of S0,2
</Internal Shape>
<Internal Shape>
  <Name> S0,4 </Name> 153 106 -63.43 128 128 128 XLWDS
  descriptor of S0,4
</Internal Shape>
</Shape>
</DXLWDS>
```

Fig. 36 illustrates the image of a logo. As this image is synthetic, the regions are well separated. In Fig. 37 we can see the structure tree associated to the set of regions.

The shape is composed by three regions: the region $S_{0,0}, S_{0,3}$ and a region of level one $S_{1,0}$ which is composed by two neighboring regions $S_{0,1}$ and $S_{0,2}$. The XLWDS descriptor is then the grouping of XLWDS of all regions into XML structured text as explained in Section 3.

Fig. 38 illustrates the image of another logo where regions are less well separated. In Fig. 39 we can see the tree structure associated and the distortion of regions $S_{0,0}$ and $S_{1,1}$ in relation to initial image.

6.3. Shape reconstruction

Fig. 40 illustrates the result of the decomposition into parts and elementary contours of Cow outline shape of the database of Leibe and Shiele [20]. This decomposition is done using Chetverikov's algorithm [8] with two different values of (α, d) : $\alpha = 150^\circ, d = 5$ pix-

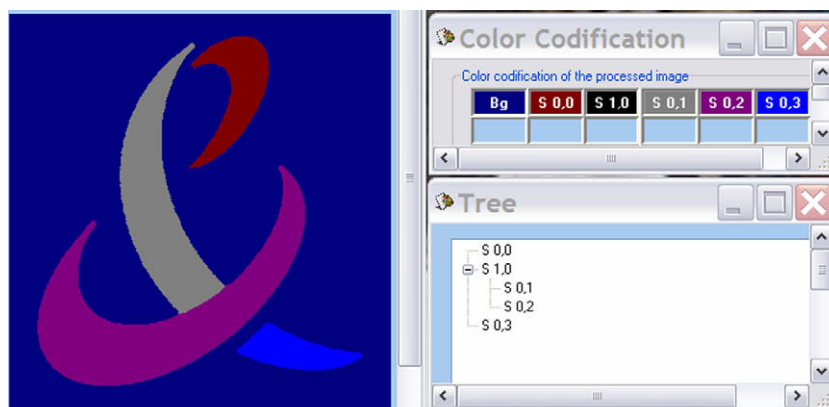


Fig. 37. Regions of the shape and the correspondent tree.

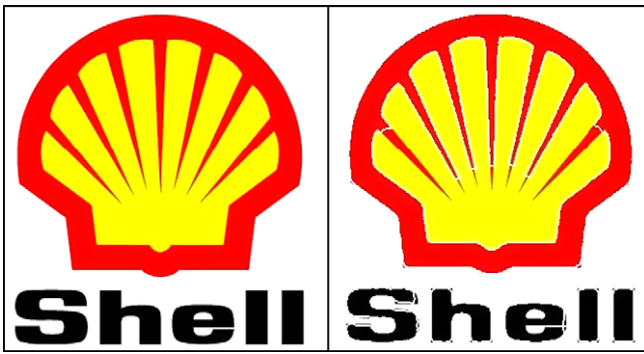


Fig. 38. Image of industrial logo and the result of its segmentation.

els and $\alpha = 160^\circ$, $d = 5$ pixels. We can see that the increase of α permits to get more curvature points.

Applying the decoding process on the XML descriptor, the drawn shapes are illustrated in same figure. Fig. 41 illustrates the results obtained applying the same process using outline shape of a Horse.

The visual quality of reconstructed shape approximates the initial shape when the outline is well decomposed and well described. To quantify this quality, we used the fact that the separating lines of the two shapes (initial and reconstructed) are identical. The superposition of these shapes through one of the separating lines involves the superposition of all lines. The defect of shape reconstruction concerns the outline shape. We calculate therefore the loss of pixels RL as the sum of rate of additional pixels and the rate of missing pixels. For example the drawn shape “cow” RL decreases from 5.3% to 0.7% when the angle α increases from 150° to 170° . This is also the case for the drawn shape “horse” for which the RL decreases from 6.8% to 0.9% when the angle α increases from 150° to 170° .

In addition to image quality, there are other important aspects which are the time needed to decode an image and the size of the image descriptor. With PC Pentium IV 2, 66 GHz, the execution time depends on the used parameters α , d in shape coding. More α is greater, more is the execution time. This time varies from 110 to 188 ms for the reconstructed shape “cow” and from 125 to 172 ms for the reconstructed shape “horse”.

For shape reconstruction, the same algorithm was applied to all regions. Fig. 42 illustrates the initial image (logo) and the reconstructed images starting from the textual descriptor.

The quality of reconstruction depends only on the quality of the decomposition of outlines of parts into elementary contours. The good reconstruction is due to the values of the parameters d and the angle α equal, respectively, to 5 pixels and 160° .

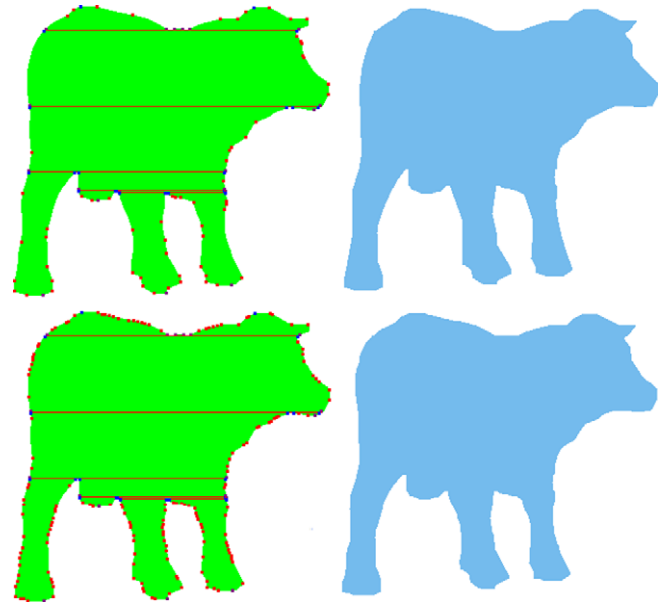


Fig. 40. Cow decomposition with $d = 5$ pixels, $\alpha = 150^\circ$ (above) and $\alpha = 170^\circ$ (below) and the result of its reconstruction.

6.4. Outline shape retrieval based on the textual descriptor

We used for the validation of the proposed similarity measure a subset of shapes from fish-shape database [1] (see Table 2). Fig. 43 and Table 3 illustrate the shape query and the result of the decomposition of smoothed shapes with a Gaussian filter ($\sigma = 29.88$).

To retrieve the most similar to shape query, we select from the subset of shapes models those which have the same index and we compute for each one the three similarity factors and then the similarity measure.

Table 3 illustrates the obtained results and shows that the proposed measure produces a good order of similarity between shapes.

7. Conclusion

In this paper, we proposed in the first a method for describing outline shape using parts, junction line and disjunction line. An XML language is proposed for writing descriptors of outline shapes. We proposed in the next a shape description method based on the use of the main silhouette and its internal regions. Due to the nature of the decomposition of the shape, the obtained description is

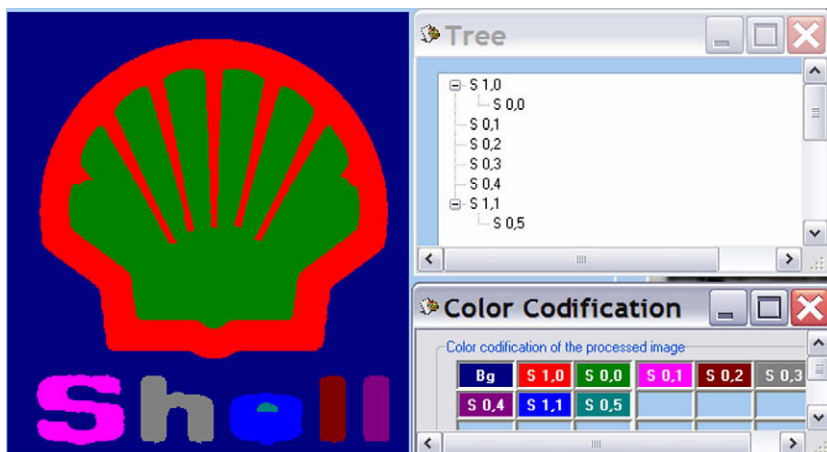


Fig. 39. Regions of the shape and the correspondent tree.

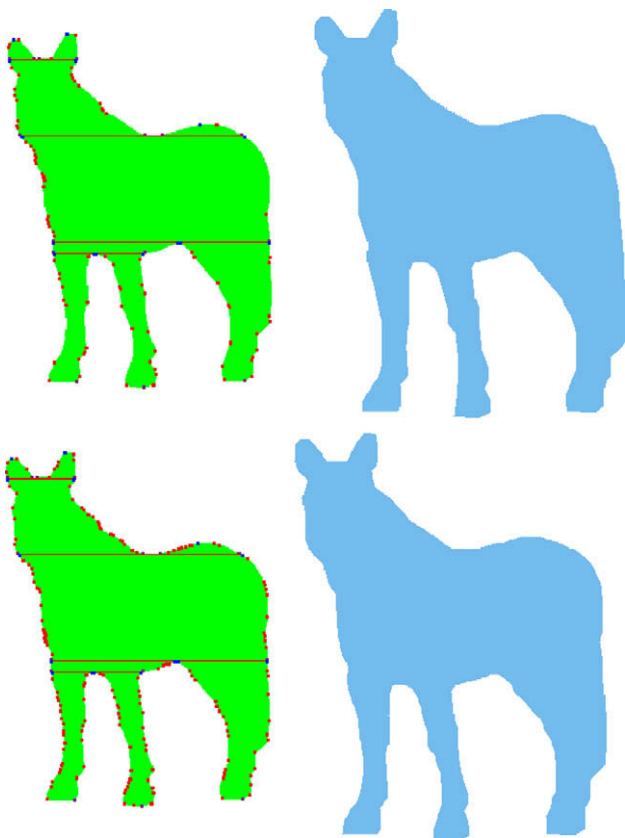


Fig. 41. Horse decomposition with $d = 5$ pixels, $\alpha = 150^\circ$ (above) and $\alpha = 170^\circ$ (below) and the result of its reconstruction.

invariant for scale change and rotation. We presented also the XML language noted XLWDS that permit to write textually these descriptors.

A set of applications of the textual descriptor are proposed:

- A method for shape comparison and similarity measure which is computed directly from the textual descriptor.

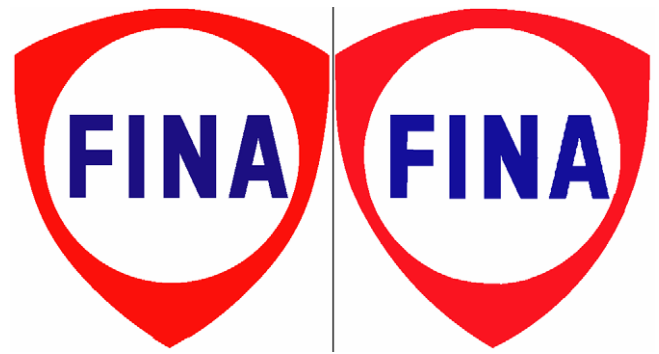


Fig. 42. Initial and reconstructed shape.

- A method for the shape descriptor decoding and visualization of correspondent image.

Results obtained applying the proposed methods over real images are presented demonstrating the useful of the textual descriptor.

Appendix A. Chetverikof's algorithm [8]

The proposed two-pass algorithm defines a corner in a simple and intuitively appealing way, as a location where a triangle of specified size and opening angle can be inscribed in a curve.

A curve is represented by a sequence of points in the image plane. The ordered points are densely sampled along the curve, no regular spacing between them is assumed. A chain-coded curve can also be handled if converted to a sequence of grid points.

In the first pass the algorithm scans the sequence and selects candidate corner points.

The second pass is post-processing to remove superfluous candidates.

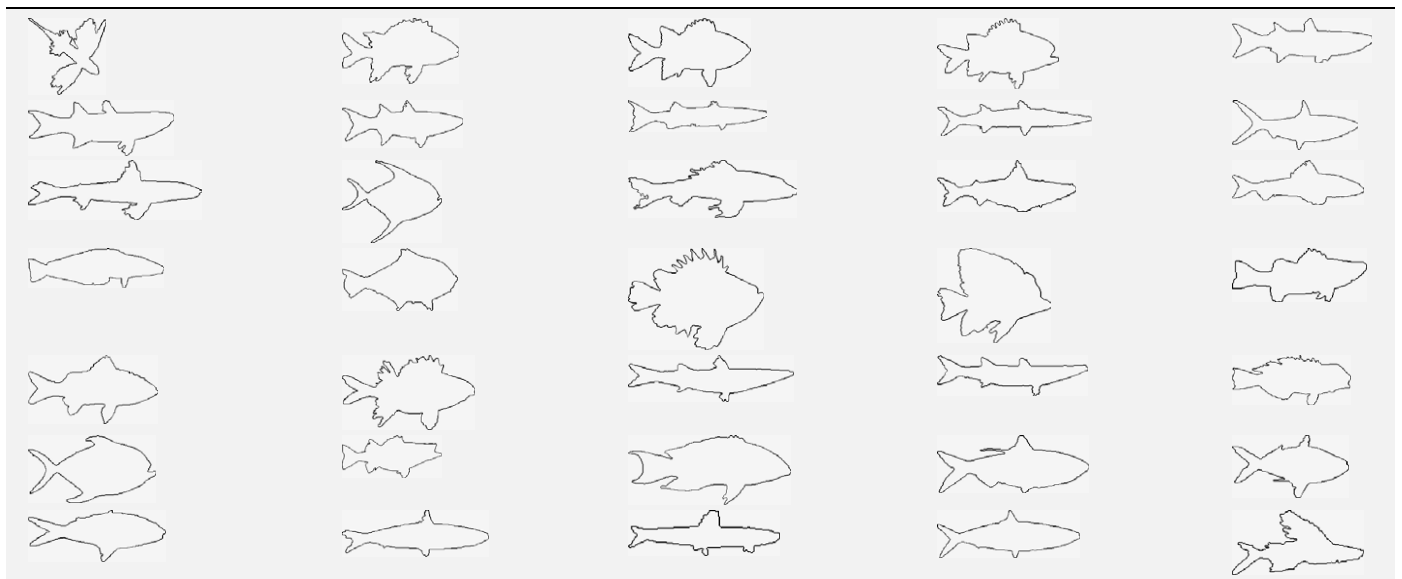
A.1. First pass

In each curve point the detector tries to inscribe in the curve a variable triangle (p^-, p, p^+) constrained by a set of rules:

$$d_{\min}^2 \leq |p - p^+|^2 \leq d_{\max}^2, d_{\min}^2 \leq |p - p^-|^2 \leq d_{\max}^2, \alpha \leq \alpha_{\max}, \dots \quad (1)$$

Table 2

Some of fish shapes from the database.



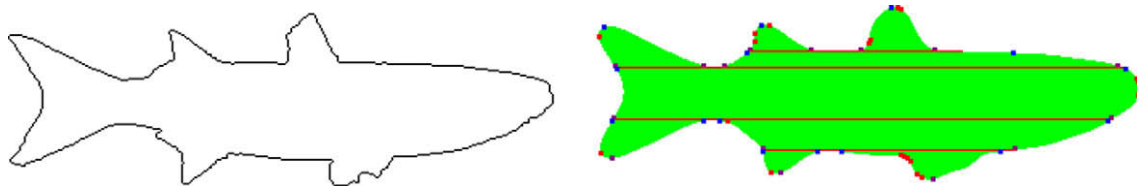


Fig. 43. Outline shape query and its decomposition.

Table 3
Similarity measure for outline shape with same index.

Outline shape	Result of decomposition	Similarity measure
		SF1 = 0.096 SF2 = 0.027 SF3 = 0.105 SM = 0.228
		SF1 = 0.154 SF2 = 0.051 SF3 = 0.251 SM = 0.231
		SF1 = 0.185 SF2 = 0.027 SF3 = 0.275 SM = 0.487
		SF1 = 0.235 SF2 = 0.143 SF3 = 0.276 SM = 0.654
		SF1 = 0.166 SF2 = 0.095 SF3 = 0.402 SM = 0.663
		SF1 = 0.249 SF2 = 0.134 SF3 = 0.357 SM = 0.740

where $|p - p^+| = |a| = a$ is the distance between p and p^+ , $|p - p^-| = |b| = b$ is the distance between p and p^- and $\alpha \in [-\Pi, \Pi]$ the opening angle of the triangle. The latter is computed as:

$$\alpha = \arccos \frac{a^2 + b^2 - c^2}{2ab}$$

Variations of the triangle that satisfy the conditions (1) are called admissible. Search for the admissible variations starts from outwards and stops if any of the conditions (1) is violated (that is, a limited number of neighboring points is only considered). Among the admissible variations, the least opening angle $\alpha(p)$ is se-

lected. $\Pi - |\alpha(p)|$ is assigned to p as the sharpness of the candidate. If no admissible triangle can be inscribed, p is rejected and no sharpness is assigned.

Considering the vector product of $b = (b_x, b_y)$ and $c = (c_x, c_y)$, it is easy to see that the corner is convex if $b_x c_y - b_y c_x \geq 0$, otherwise it is concave.

A.2. Second pass

The sharpness based non-maxima suppression procedure is illustrated by Fig. 44. A corner detector can respond to the same corner

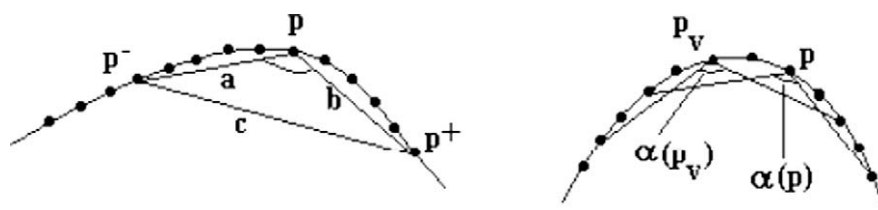


Fig. 44. Detecting high curvature points. (a) Determining if is a candidate point. (b) Testing for sharpness non-maxima suppression.

in a few consecutive points. Similarly to edge detection, a post-processing step is needed to select the strongest response by discarding the no maxima points. A candidate point p is discarded if it has a sharper valid neighbor $p_v: \alpha(p) > \alpha(p_v)$.

Appendix B. Grammar of XLWDOS language

The grammar G of the language XLWDOS (XML Language for Writing Descriptors of Outline Shapes) is defined as $G = (V_N, V_T, P, S_0)$ where: V_T, V_N are, respectively, the finite set of terminal vocabulary and the finite set of non-terminal vocabulary, $S_0 \in V_N$ is the starting symbol (in our case, it corresponds to Silhouette), and P is a finite set of production rules of the type $\alpha \rightarrow \beta$, where $\alpha \in V_N$ and $\beta \in (V_N \cup V_T)^*$ of all string. The non-terminal vocabulary of XLWDOS language is written as:

- $V_N = \{iSilhouette_i, iComposed-Part_i, iPart_i, iList-Parts_i, iLeft-Boundary_i,$
- $iRight-Boundary_i, iContour-Descriptor_i, iType_i, iLength_i, iJunction-Line_i,$
- $iInclination-Angle_i, iDisjunction-Line_i, iConvexity-Degree_i,$
- $iConcavity-Degree_i,$
- $iSegment-Description_i, iHigh-Part-Number_i, iLow-Part-Number_i, iNumPart_i,$
- $iNumJunction_i, iNumDisjunction_i, iSegments_i\}$

The terminal vocabulary of XLWDOS language is:

$V_T = \{s, w, h, <L>, </L>, <R>, </R>, <CP>, </CP>, <P>, </P>, <J>, </J>, <D>, </D>\}$

The set P of production rules are:

$iSilhouette_i \rightarrow <DXLWDOS>$
<Name> “Silhouette name” **</Name>**
<CP> $iPart_i$ **</CP>**
</DXLWDOS>
 $iPart_i \rightarrow iComposed-Part_i/$
<P $iNumPart_i <L> iLeft-Boundary_i </L> <R> iRight-Boundary_i </R> </P $iNumPart_i >$
 $iComposed-Part_i \rightarrow iPart_i iList-Parts_i iJunction-Line_i iPart_i/$
 $iPart_i iDisjunction-Line_i iPart_i iList-Parts_i$
 $iList-Parts_i \rightarrow iPart_i iList-Parts_i/iPart_i$
 $iLeft-Boundary_i \rightarrow iContour-Descriptor_i iLeft-Boundary_i/$
 $iContour-Descriptor_i$
 $iRight-Boundary_i \rightarrow iContour-Descriptor_i iRight-Boundary_i/$
 $iContour-Descriptor_i$
 $iContour-Descriptor_i \rightarrow iType_i iInclination-Angle_i iLength_i$
 $iType_i \rightarrow cv$ $iConvexity-Degree_i/cc$ $iConcavity-Degree_i/r$
 $iInclination-Angle_i \rightarrow$ INTEGER VALUE
 $iLength_i \rightarrow$ INTEGER VALUE
 $iJunction-Line_i \rightarrow <J$ $iNumJunction_i >$ $iSegments_i </J$
 $iNumJunction_i >$
 $iDisjunction-Line_i \rightarrow <D$ $iNumDisjunction_i >$ $iSegments_i </D$
 $iNumDisjunction_i >$
 $iSegments_i \rightarrow iSegment-Description_i iSegments_i/iSegment-Description_i$
 $iSegment-Description_i \rightarrow s$ $iHigh-Part-Number_i iLow-Part-Number_i iLength_i/$
 w $iLow-Part-Number_i iLength_i/h$ $iHigh-Part-Number_i$
 $iLength_i$
 $iHigh-Part-Number_i \rightarrow$ INTEGER VALUE
 $iLow-Part-Number_i \rightarrow$ INTEGER VALUE
 $iConvexity-Degree_i \rightarrow$ INTEGER VALUE
 $iConcavity-Degree_i \rightarrow$ INTEGER VALUE
 $iNumJunction_i \rightarrow$ INTEGER VALUE$

$iNumDisjunction_i \rightarrow$ INTEGER VALUE
 $iNumPart_i \rightarrow$ INTEGER VALUE

Appendix C. Additional rules for XLWDS Language

$S \rightarrow <DXLWDS> <Name> \text{“name of the shape”} </Name>$
 $iShape_i </DXLWDS>$
 $iShape_i \rightarrow <Shape> iPosition_i iOrientation_i iColor_i iSilhouette_i$
 $iList-Internal-Shapes_i </Shape>$
 $iList-Internal-Shapes_i \rightarrow iInternal-Shape_i iList-Internal-Shapes_i/iInternal-Shape_i$
 $iInternal-Shape_i \rightarrow <Internal Shape> <Name> \text{“name of internal shape”} </Name>$
 $iPosition_i iOrientation_i iColor_i iSilhouette_i$
</Internal Shape>/
<Internal Shape> **<Name>** “name of internal shape” **</Name>**
 $iPosition_i iOrientation_i iColor_i iSilhouette_i$
 $iList-Internal-Shapes_i$
</Internal Shape>
 $iPosition_i \rightarrow iX-Position_i iY-Position_i$
 $iX-Position_i \rightarrow$ INTEGER VALUE
 $iY-Position_i \rightarrow$ INTEGER VALUE
 $iOrientation_i \rightarrow$ INTEGER VALUE
 $iColor_i \rightarrow <C> iR_i iG_i iB_i </C>$
 $iR_i \rightarrow$ INTEGER VALUE
 $iG_i \rightarrow$ INTEGER VALUE
 $iB_i \rightarrow$ INTEGER VALUE

References

- [1] S. Abbasi, F. Mokhtarian, J. Kittler, Search for similar shapes in the SQUID system: shape queries using image databases, 1997. Available from: <http://www.ee.surrey.ac.uk/CVSSP/demos/css/demo.html>.
- [2] N. Arica, F.T.Y. Vural, BAS: a perceptual shape descriptor based on the beam angle statistics, *Pattern Recognition Letters* 24 (9) (2003) 1627–1639.
- [3] O.E. Badawy, M. Kamel, Shape representation using concavity graphs, in: *Proceeding of the 16th International Conference on Pattern Recognition (ICPR'02)*, vol. 3, 2002, pp. 461–464.
- [4] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (4) (2002) 509–522.
- [5] T. Bernier, J.A. Landry, A new method for representing and matching shapes of natural objects, *Pattern Recognition* 36 (8) (2003) 1711–1723.
- [6] S. Berretti, A. Bimbo, Retrieval by shape similarity with perceptual distance and effective indexing, *Transactions on Multimedia* 2 (4) (2000).
- [7] R.J. Campbell, P.J. Flynn, A survey of free-form object representation and recognition techniques, *Computer Vision and Image Understanding* 81 (2001) 166–210.
- [8] D. Chetverikov, A simple and efficient algorithm for detection of high curvature points in planar curves, in: *Proceedings of the 10th International Conference, CAIP 2003*, Groningen, The Netherlands, August 25–27, 2003.
- [9] T.M. Cronin, Visualizing concave and convex partitioning of 2D contours, *Pattern Recognition Letters* 24 (2003) 429–443.
- [10] C.M. Cyr, B.B. Kimia, A similarity-based aspect-graph approach to 3D object recognition, *International Journal of Computer Vision* 57 (1) (2004) 5–22.
- [11] Y. Deng, B.S. Manjunath, Unsupervised segmentation of color-texture regions in images and video, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23 (8) (2001) 800–810.
- [12] C. Direkoglu, M.S. Nixon, Image-based multiscale shape description using gaussian filter, in: *IEEE Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP-2008)*, Bhubaneswar, India, December 2008.
- [13] P.F. Felzenszwalb, D.P. Huttenlocher, Pictorial structures for object recognition, *International Journal of Computer Vision* 61 (1) (2005) 55–79.
- [14] D. Geiger, T. Liu, R.V. Kohn, Representation and self-similarity of shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (1) (2003) 86–99.
- [15] W.I. Grosky, R. Mehrota, Index-based recognition in pictorial data management, *Computer Vision Graphics and Image Processing* 52 (1990) 416–436.
- [16] D.H. Kim, I.D. Yun, S.U. Lee, A new shape decomposition scheme for graph-based representation, *Pattern Recognition* 38 (2005) 673–689.

- [17] J.J. Koenderink, The structure of images, *Biological Cybernetics* 50 (1984) 363–370.
- [18] J.J. Koenderink, V. Doorn, The internal representation of solid shape with respect to vision, *Biological Cybernetics* 32 (1976) 211–216.
- [19] S. Larabi, S. Bouagar, F.M. Trespadarne, E.L. Fuente, LWDOS: language for writing descriptors of outline shapes, in: *Proceeding of Scandinavian Conference on Image Analysis*, June 29–July 02, Goteborg, Sweden, 2003, *Lecture Notes in Computer Science*, vol. 2749, pp. 1014–1021.
- [20] B. Leibe, B. Schiele, Analyzing appearance and contour based methods for object categorization, in: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003, pp. 409–415.
- [21] H.-C. Liu, M.D. Srinath, Corner detection from chain-code, *Pattern Recognition* 23 (1–2) (1990) 51–68.
- [22] T. Lourens, R.P. Wurtz, Object recognition by matching symbolic edge graphs, in: *Proceeding of the ACCV'98*, 1998, pp. 193–200.
- [23] P.A. Maragos, R.W. Schafer, Morphological skeleton representation and coding of binary images, *IEEE transactions on Acoustic, Speech Signal Processing* 34 (5) (1986) 1228–1244.
- [24] F. Mokhtarian, M. Bober, *Curvature Scale Space Representation: Theory, Applications and MPEG-7 Standardization*, Kluwer Academic Publisher, Dordrecht, 2003.
- [25] F. Mokhtarian, A.K. Mackworth, A theory of multiscale, curvature-based shape representation for planar curves, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (8) (1992) 789–805.
- [26] F. Mokhtarian, Silhouette-based isolated object recognition through curvature scale space, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17 (5) (1995) 539–544.
- [27] H. Murase, S.K. Nayer, Visual learning and recognition of 3D objects from appearance, *International Journal of Computer Vision* 14 (1) (1995) 5–24.
- [28] R.C. Nelson, A. Selinger, Large-scale tests of a keyed, appearance-based 3D object recognition system, *Vision Research* 38 (1998) 2469–2488.
- [29] C. Orrite, J.E. Herreo, Shape matching of partially occluded curves invariant under projective transformation, *Computer Vision and Image Understanding* 93 (1) (2004) 34–64.
- [30] E.G.M. Petrakis, A. Diplaros, E. Milios, Matching and retrieval of distorted and occluded shapes using dynamic programming, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (11) (2002) 1501–1516.
- [31] I. Pitas, A.N. Venetsanopoulos, Morphological shape decomposition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12 (1) (1990) 38–45.
- [32] T.W. Ridler, S. Calvard, Picture thresholding using an iterative selection method, *IEEE Transactions on Systems, Man and Cybernetics* 8 (1978) 630–632.
- [33] B.M.H. Romeny, Introduction to scale-space theory: multiscale geometric image analysis, in: *Fourth International Conference on Visualization in Biomedical Computing*, Tutorial VBC'96, Hamburg, Germany.
- [34] P.L. Rosin, Shape partitioning by convexity, in: *Proceeding of British Machine Vision Conference*, Nottingham, 1999, pp. 633–642.
- [35] C. Ruberto, Recognition of shapes by attributed skeletal graphs, *Pattern Recognition* 37 (2004) 21–31.
- [36] T.B. Sebastian, P.N. Klein, B.B. Kimia, Recognition of shapes by editing their shock graphs, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (5) (2004) 550–571.
- [37] J.A. Schnabel, S.R. Arridge, Active contour models for shape description using multiscale differential invariants, in: *Proceedings of the 1995 British Conference on Machine Vision*, vol. 1, Birmingham, United Kingdom, 1995, pp. 197–206.
- [38] A. Sethi, D. Renaudie, D. Kriegman, J. Ponce, Curve and surface duals and the recognition of curved 3D objects from their silhouette, *International Journal of Computer Vision* 58 (1) (2004) 73–86.
- [39] Z. Shao, J. Kittler, Shape representation and recognition based on invariant unary and binary relations, *Image and Vision Computing* 17 (5–6) (1999) 429–444.
- [40] K. Siddiqi, B.B. Kimia, Parts of visual form computational aspects, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17 (3) (1995) 239–251.
- [41] K. Siddiqi, B.B. Kimia, A shock grammar for recognition, in: *Conference of Computer Vision and Pattern Recognition*, June 18–20, 1996, pp. 507–513.
- [42] C. Stauffer, W.E.L. Grimson, Learning patterns of activity using real-time tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (8) (2000) 747–757.
- [43] Z. Tu, X. Chen, A.L. Yuille, S. Zhu, Image parsing: unifying segmentation, detection, and recognition, *International Journal of Computer Vision* 63 (2) (2005) 113–140.
- [44] M.H. Yu, C.C. Lim, J.S. Jin, Shape similarity search using XML and portal technology, in: *Proceeding of Visual Information Processing Workshop (VIP-2003)*, Sydney, Australia, 2003, pp. 109–112.
- [45] L. Yu, R. Wang, Shape representation based on mathematical morphology, *Pattern Recognition Letters* 26 (9) (2004) 1354–1362.
- [46] D. Zhang, G. Lu, Review of shape representation and description techniques, *Pattern Recognition* 37 (2004) 1–19.
- [47] S.C. Zhu, A.L. Yuille, FORMS: a flexible object recognition and modeling system, in: *Proceedings of the Fifth International Conference on Computer Vision*, June 20–23, MIT Press, Cambridge, 1995, pp. 466–472.