**International Journal of Computational Vision and Biomechanics**
**Vol.2 No. 1 (January-June, 2016)**

# Textual Description of Shapes

**Slimane LARABI**

*Computer Science Department, Faculty of Electronic and Computer Science*
*University of Sciences and Technologies HOUARI BOUMEDIENNE, BP 32, El Alia, Algiers, Algeria*

*We present in this paper a part-based method for shape representation and description using both its outer contour and the interior regions that contains. We present firstly a part based method for silhouette description invariant to translation, rotation and scale change and written with an XML language XLWDOS (XML Language for Writing Descriptors of Outlines Shapes).*

*Any shape is characterized by the list of encompassed regions, the inclusion relation between these regions, the position and the description of each one. Shape is then represented by a tree structure which is translated to a textual marked descriptor using a proposed XML language XLWDS (XML Language for Writing Descriptors of Shapes).*

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

***Key words:*** *Shape - Silhouette - Concavity - Part - Description - Language – Textual - XML*

## 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 follow:

– Part-based methods where silhouette is decomposed into parts. Different criterions have been proposed: limb and neck [32], optimization of the convexity of all parts [27] and morphology criteria [21, 26]. In [36] 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 [14]

– Aspect-graph methods are viewer-centered representations of a three-dimensional object. The underlying theory was introduced by J. J. Koenderink and V. Doorn [15] 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 [9], authors 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. Authors in [38] propose flexible object recognition and modeling system where shape representation is based on medial axis extraction and part segmentation using deformable circles. In [28] author 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) [12].

– Methods based on the shock graph that is an emerging shape representation for object recognition, in which a 2-D silhouette is decomposed into a set of qualitative parts, captured in a directed acyclic graph [33]. Authors in [29] 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 authors in [2] 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 [18], 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 [13] have approximated shapes as polygonal curves: for each vertex, a local feature is defined by considering the internal angle at the vertex, the distance from the adjacent vertex, and the vertex coordinates. Petrakis, Milios in [25] have approximated shapes as a sequence of convex/concave segments between two consecutive curvature points. Concave and convex sections are used in [23] 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. Author in [8] 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. Authors present in [20, 21] propose a multiscale curvature-based shape representation technique for planar curves. In [31] 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 [5].

– Methods based on the attributes of outline shape. In [1], 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 [4], shape is represented directly by its contour in term of the angle from the centroïd 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 [24]. Authors in [30] 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 [3].

– Appearance-based methods proposed initially by H. Murase, S. K. Nayer [22].

A review of shape representation methods can be found in [6, 37]

Silhouette associated to the image of object isn't sufficient in many cases for recognition of object. Internal regions to the main silhouette are fundamental and must be included in shape representation. Authors in [11] 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 in 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 part-based method for image representation is proposed in [34] where authors 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 of any image is represented by a hierarchical graph similar to parsing sentences in speech and natural language. 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 on.

Even if each one of the proposed method for outline shape and shape description 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 descriptors are numerical values stored with a specific data structure.

Our aim is to homogeneous all databases of shapes models and to offer accessibility to all users. For this, 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. The written of the shape descriptions with an XML language is the format that satisfies these criterions.

Tacking in account the above recommendations, we propose in this paper is a new method for shape representation and description parsing the shape into outer contour and interior regions. Relatively to the work of Z. Tu *et al.* [34], the output of shape parsing is our case is both a tree structure and a geometrical description of all outline regions of the shape written with an XML language.

Firstly we propose a part-based method for silhouette representation that decomposes silhouette in parts and separating lines and describes them geometrically. The obtained description is translated to a textual marked descriptor using a proposed language XLWDOS (XML Language for Writing Descriptors of Outline Shapes) [16].

We assume that shape may contain internal regions. These regions may also contain internal regions, and so on. 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 on. The output of shape parsing 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.

The proposed method verifies the uniqueness, invariance to translation, rotation and scale change properties. Compared with existing part-based representations, the proposed scheme provides the following additional properties: Coding of shapes with an XML text, reducing the size of computed description, the easiness for index extraction and the preservation of perceptual structure.

This paper is organized as follow: Section 2 describes the part-based method for representing objects from their silhouettes. In section 3, we present the XML version of the language LWDOS allowing writing silhouette descriptors. Section 4 introduces a new description method of shapes tacking in account the internal regions and the XML language noted XLWDS (XML Language for Writing Descriptors of Shapes). In section 5 we present and discuss some results of experiments conducted over real images demonstrating the utility of this description. Finally, section 6 concludes this paper.

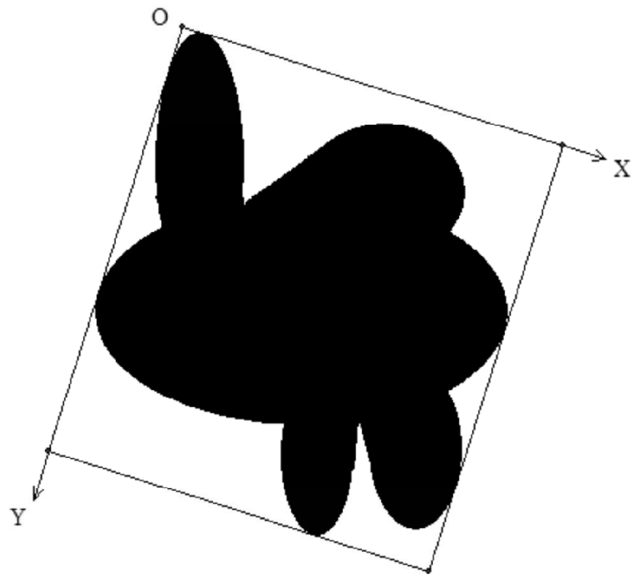



**Figure 1: The rectangle of minimum area encompassing the silhouette**

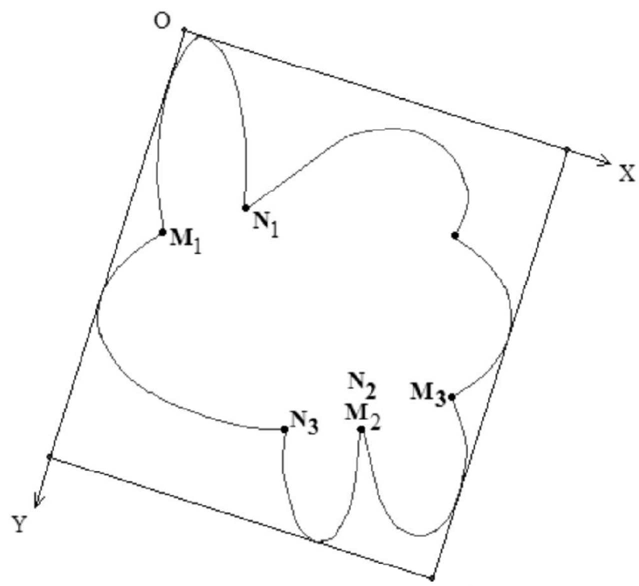



**Figure 2: The set of concave points**

## 2. DESCRIPTION OF SILHOUETTES

### 2.1. Silhouette Decomposition

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 $S$

– $OXY$ 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 $RM$

– $(X_Q, Y_Q)$ 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 figure 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 $(\Delta_i)$ 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 $(\Delta'_i)$ 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 $(\Delta_i)$ decompose the silhouette $S$ into parts following the X-direction (see figure 3) and the set of lines $(\Delta'_i)$ decompose the silhouette into parts following the Y-direction (see figure 4).

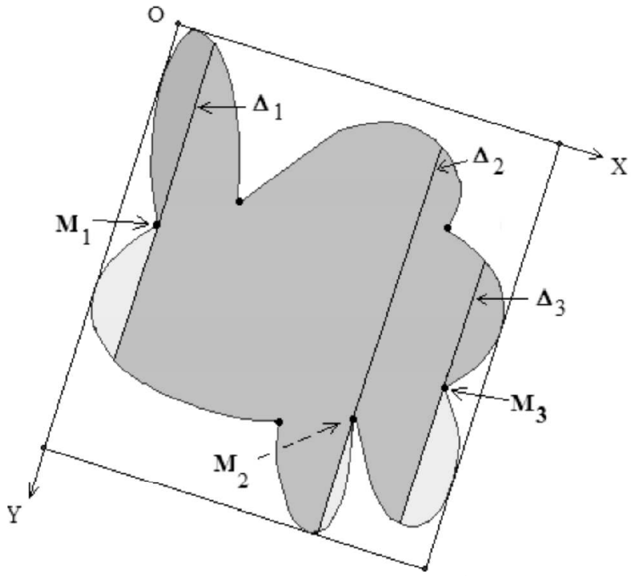



**Figure 3: X-Decomposition of (S)**

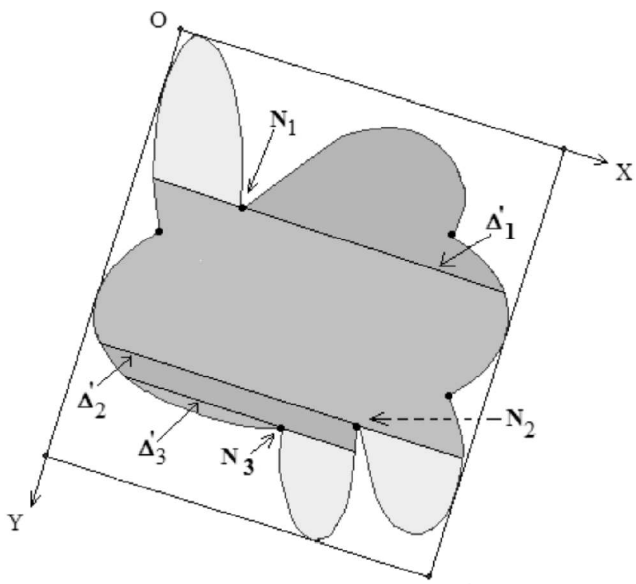



**Figure 4: Y-Decomposition of (S)**

When many concave points $M_i$ (or $N_i$) appertain to the same separating line, the separating process is identical. Figure 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.

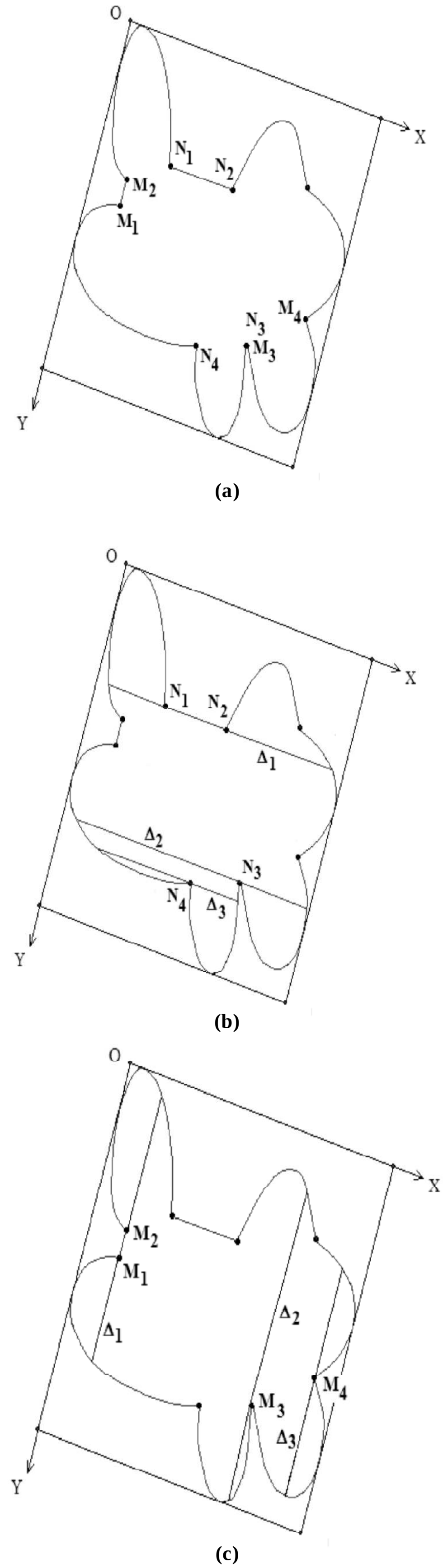



**Figure 5: Case where many concave points appertains to the same separating line (a) the silhouette (S), (b) the X-Decomposition of (S), (c) the Y-Decomposition of (S)**

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 figure 6.

**Table 1**
**Extremities of boundaries part**

| | *Left boundary* | | *Right boundary* | |
|---|---|---|---|---|
| *Position of the part* | *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 |

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

Using curvature points, each boundary is segmented into elementary contours whose description is defined by the following parameters (see figure 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.

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 figure 7):

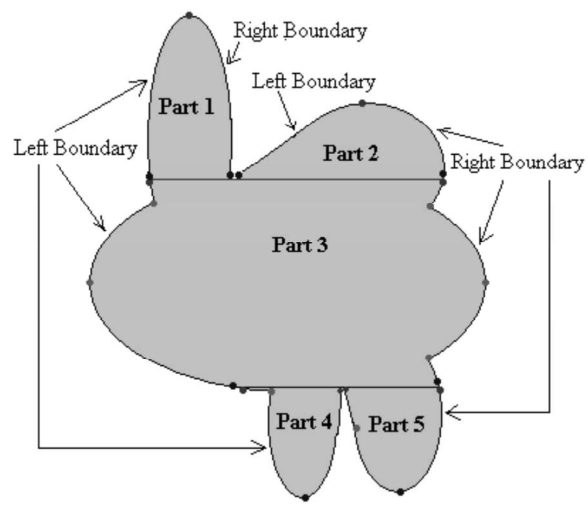



**Figure 6: Boundaries of parts**

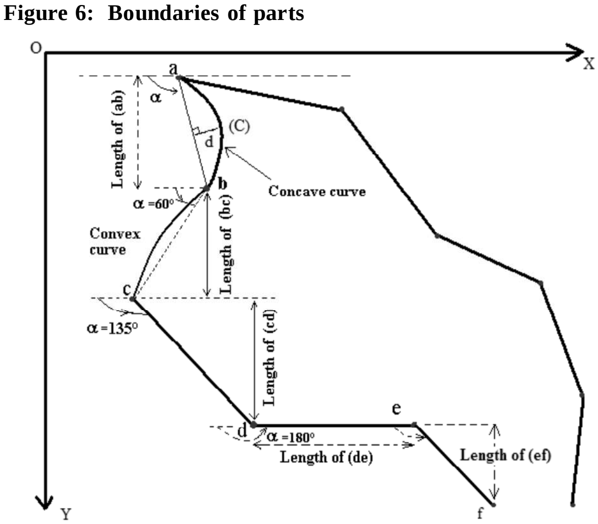



**Figure 7: Description of elementary contours**

- 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:

((Type, ReferenceNumber1, ReferenceNumber2*, Length)$_1$,

(Type, ReferenceNumber1, ReferenceNumber2*, Length)$_2$, …….,

(Type, ReferenceNumber1, ReferenceNumber2*, 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 figure 8 as follow:

((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.

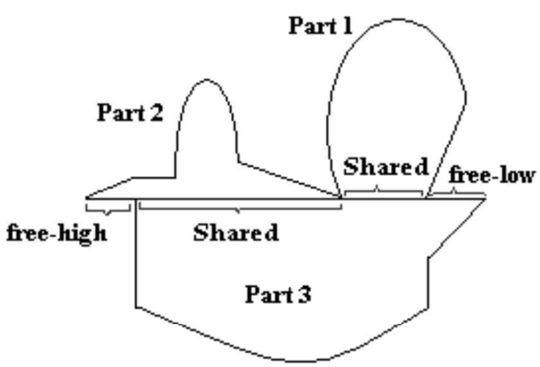



**Figure 8: Description of partitioning line**

## 2.4. Properties of the Silhouette Description

The method presented in this paper for silhouette representation verifies the following properties:

**Property 1**: Uniqueness and preservation of perceptual structure

The description of silhouette proposed reflects precisely the geometry of its contour. Thus, it may be possible the reconstruction of the silhouette whose accuracy depends on the accuracy of the boundary description. We explain in follow how the outline may be drawn from its description without ambiguity.

Let ($a_i$, $b_i$,...., $x_i$, $y_i$, $z_i$) (resp. $(a'_i, b'_i, ...., x'_i, y'_i, z'_i)$) be the set of curvature points located on the left (resp. right) boundary of the part $P_i$ where $(a_i, a'_i)$ are the beginning points and $(z_i, z'_i)$ are the end points.

The part $P_i$ may be drawn by fixing the end point ($z_i$) and the draw of the contour ($z_i$ $y_i$) knowing its length, inclination angle and concavity or convexity degree in case of curve contour. Each one of other contours of the left boundary: $y_i x_i$… $c_i b_i$ and $b_i a_i$ will be drawn using their computed end point and their description. The right boundary is drawn following the same steps.

To joint the two boundaries, three cases may occur (see Figure 9):

Case 1: $a_i = a'_i$ and $z_i = z'_i$ (it occurs when the silhouette contains only one part)

Case 2: $a_i = a'_i$ and $z_i \neq z'_i$ (it occurs when this part is followed by a separating line)

Case 3: $a_i \neq a'_i$ and $z_i = z'_i$ (it occurs when this part follows a separating line)

Case 4: $a_i \neq a'_i$ and $z_i \neq z'_i$ (it occurs when this part is neighbor to two separating lines)

The left and right boundaries of any part for the three cases 1, 2, and 3 have a common point and therefore the part drawn corresponds exactly to the described part (see Figure 10). However, in case 4 the two boundaries may be drawn but the distance between their beginning points and the distance between their end points are unknown. Each one of these distances is computed as the sum of lengths of segments appearing in the correspondent separating line that refers to this part. Figure 10 illustrates the result of the drawing process of parts and partitioning lines of the silhouette of figure 1. There are two parts of case 2, two parts of case 3 and one part of case 4.

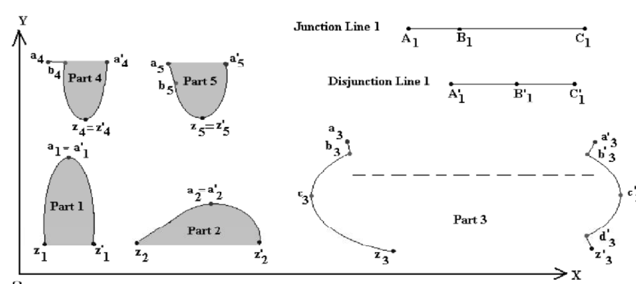



**Figure 9: Result of separation of silhouette into parts and partitioning lines**

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.

We are studying the accuracy of silhouettes reconstruction from their description for visualizing images in multimedia applications. We can already note that the accuracy of silhouettes reconstruction depends on the nature of its boundary and on the result of its

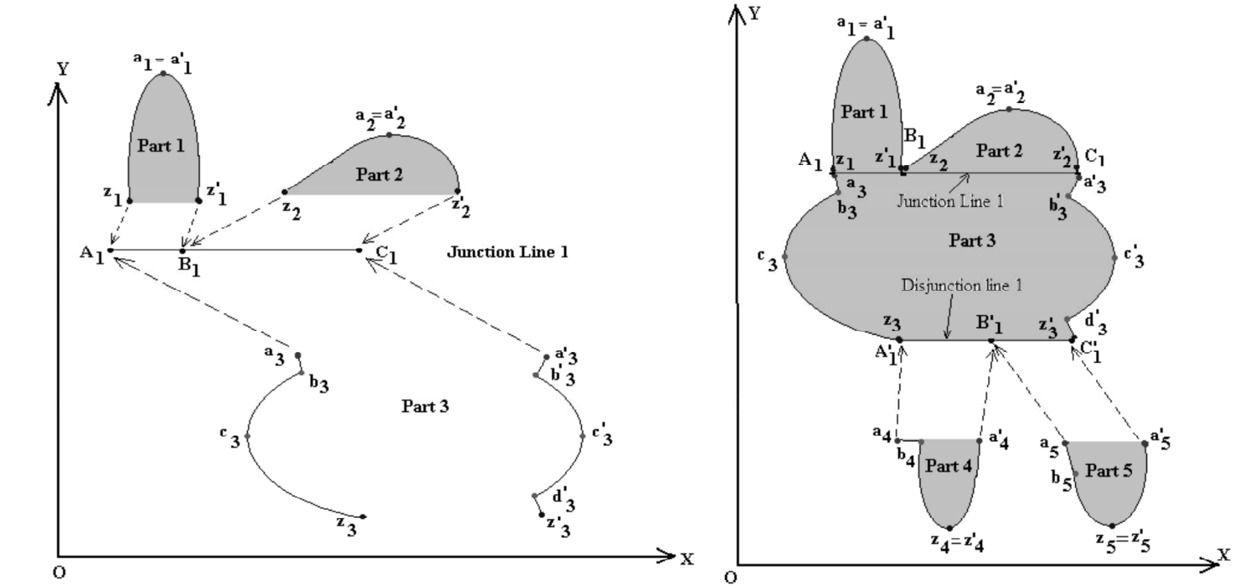


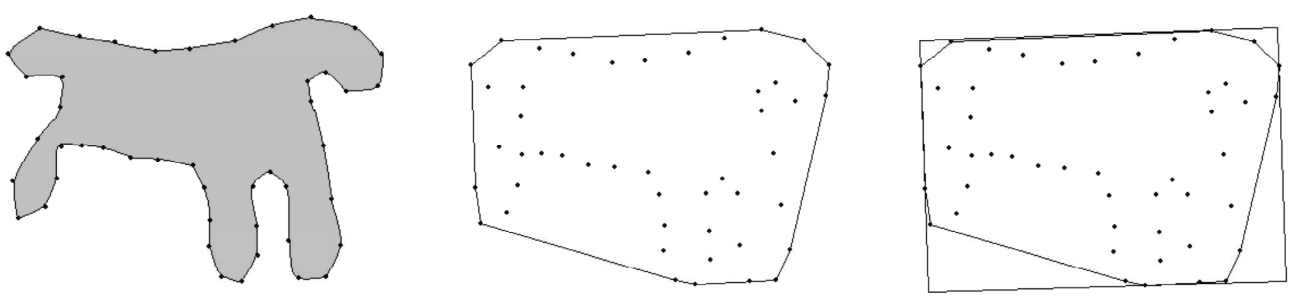

**Figure 10: Different steps of the drawing process**

segmentation into elementary contours: more curvature points increases the accuracy of the reconstruction.

**Property 2**: Invariance to rotation

The result of the decomposition process described above depend on the position of the silhouette relatively to the coordinate system OXY. The sweep up of the silhouette following one of the directions of rectangle of minimum area encompassing the silhouette implies the invariance to rotation of this description because the uniqueness of this rectangle (see Figure 11).

For each silhouette, the minimum bounding box is firstly computed and its highest corner M is located (see figure 12). A rotation of the box is performed with an angle β around the point M so as its width coincides with



(a) (b) (c)

**Figure 11: Outline shape (a), the convex hull (b) and the minimum bounding box associated (c)**

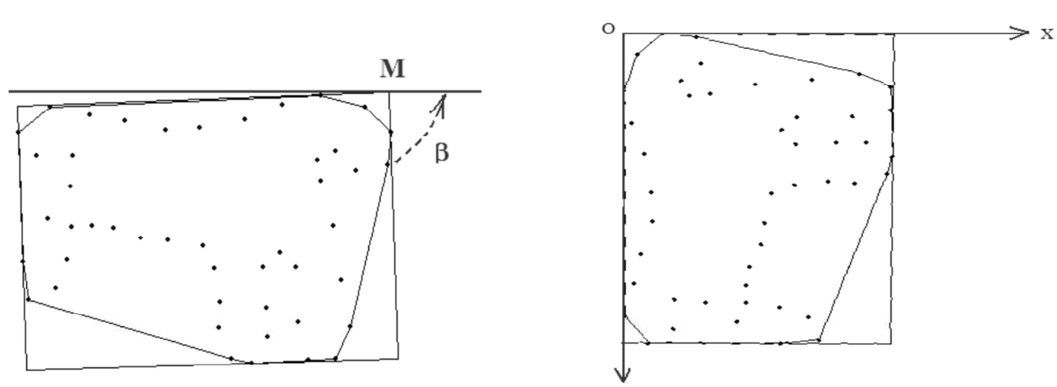



**Figure 12: Position of the coordinate system OXY after the rotation of the minimum bounding box**

the rows. The coordinates system OXY is then attached to the box so as OX coincides with rows and OY with columns. More explanations are given in the experimentation section.

**Property 3**: Invariance to scale change

The use of absolute lengths produces an absolute description of silhouette. For computer vision applications as recognition of objects, relative lengths are used to guaranty the scalability property. All lengths are computed relatively to the length of the minimum bounding box.

## 3. 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 [35], authors 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 …) in the SVG format of images.

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

In [20] we 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. Thereafter, we introduce the notion of composed part that permits the grouping of all components of the silhouette.

### 3.1. Description of Silhouette Elements

#### *3.1.1. 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 follow:

Part → <P PartNumber> LeftBoundary RightBoundary </P PartNumber>

LeftBoundary → <L> Contour $_1$ … Contour $_n$ </L>

RightBoundary → <R> Contour $_1$ … Contour $_m$</R >

Contour$_i$ → cv Convexity-degree Inclination-angle Length/ cc Concavity-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 concavity, 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.

#### *3.1.2. Description of Separating Lines*

The separating line that may be a junction or disjunction line is described as follow:

Junction line → <J JunctionNumber> Segment $_1$ …. Segment$_n$ </J JunctionNumber >

Disjunction line → <D DisjunctionNumber > Segment $_1$ …Segment $_m$ </D DisjunctionNumber >

Segment $_i$ → s High-Part-Number Low-Part-Number Length / w Low-Part-Number Length / h High-Part-Number Length

Where:

- <J>, </J> and <D>, </D> 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, FreeLow and FreeHigh attributes for the segments

For example, the text: <J1>w P5 40 s P1 P5 40 w P5 40 s P2 P5 40 w P5 40 </J1> 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.

#### *3.1.3. 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 follow:

Composed Part → <CP> $P_1$ $P_2$ …. $P_n$ $J_k$ $P_{n+1}$ </CP> /

<CP> $P_1$ $D_l$ $P_{i1}$ $P_{i2}$ …. $P_{in}$ </CP>

Where:

– <CP> and </CP> 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, <CP> <CP> P1 P2 J1 P3</CP> P4 J2 P5</CP> is a composed part that groups the composed part <CP> P1 P2 J1 P3</CP>, the parts P4, P5 and the junction line J2.

*3.1.4. 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>

### 3.2. Computation of the Composed Part Associated to the Silhouette

The final composed part associated to the silhouette is written using the result of silhouette decomposition into parts and separating lines. In case where one part (P1) composes the silhouette, the correspondent composed part is then written as: <CP>$P_1$</CP> (see figure 13.a). Otherwise, there are many parts, junction and disjunction lines. In this case we write for each separating line the correspondent composed part that may have one of the following expressions:

– <CP> $P_{i1}$ ….. $P_{in}$ $J_i$ $P_i$ </CP> that means: a set of parts $P_{i1}$ …..$P_{in}$ are joined through the junction line $J_i$ to one part $P_i$ (see figure 13.b).

– <CP>$P_j$ $D_j$ $P_{j1}$ ….. $P_{jn}$</CP> that means: a part $P_j$ is joined through a disjunction line to set of parts $P_{j1}$ …..$P_{jn}$ (see figure 13.c).

– <CP>$P_{k1}$ ….. $P_{kn}$ $J_k$ $P_e$ </CP> and <CP>$P_e$ $D_m$ $P_{m1}$ ….. $P_{mn}$</CP> that means: a set of parts $P_{k1}$ ….. $P_{kn}$ are joined through the junction line $J_k$ to many parts $P_{m1}$ …. $P_{mn}$ (see figure 13.d). In this case, the separating line is both junction and disjunction line. The empty part that is noted $P_e$ whose left and right boundaries are empty is used in order to obtain the same syntax for composed parts.

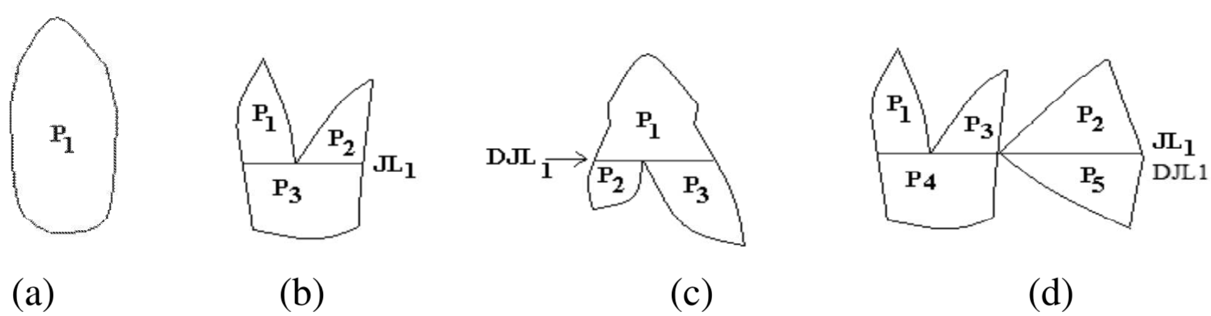



**Figure 13: Possibilities of composed parts**

We call principal part any part $P_i$ appearing at the right of junction line or at the left of disjunction line in the expression of composed part, otherwise, it will be called a secondary part.

After the writing of all composed parts, a process of substitutions is performed in order to obtain the composed part corresponding to the silhouette. We assume that we have written a set of $CPJ_i$ (composed parts with junction lines) and a set of $CPD_j$ (composed parts with disjunction lines), the process of substitutions works as follow:

**Rule 1**: If $P_i$ is a principal part in composed part $CPJ_i$ and a secondary part in another composed part $CPJ_j$ then $P_i$ in $CPJ_j$ is replaced by the expression of $CPJ_i$.

For example, the two composed parts written for the silhouette of figure 14.a are:

$CPJ_1$: <CP>$P_1$ $P_2$ $J_1$ $P_4$</CP>

$CPJ_2$: <CP>$P_3$ $P_4$ $J_2$ $P_5$</CP>

The substitution of $CPJ_1$ in $CPJ_2$ gives the description of the silhouette

<CP>$P_3$ <CP>$P_1$ $P_2$ $J_1$ $P_4$</CP> $J_2$ $P_5$</CP>

**Rule 2**: If $P_i$ appears as a principal part in two composed parts: $CPD_k$ and $CPJ_k$ then $P_i$ in $CPD_k$ is replaced by the expression of $CPJ_k$.

For example, the two composed parts written for the silhouette of figure 14.b are:

$CPJ_1$: <CP>$P_1$ $P_2$ $J_1$ $P_3$</CP>

$CPD_1$: <CP>$P_3$ $D_1$ $P_4$ $P_5$</CP>

The substitution of $P_3$ in $CPD_1$ by the expression of $CPJ_1$ gives the description of the silhouette:

<CP><CP>$P_1$ $P_2$ $J_1$ $P_3$</CP> $D_1$ $P_4$ $P_5$</CP>

**Rule 3**: If $P_i$ appears as a secondary part in $CPJ_k$ and $CPD_k$, then $P_i$ in $CPJ_k$ is replaced by the expression of $CPD_k$.

For example, composed parts written for the silhouette of figure 14.c are:

$CPD_1$: <CP>$P_2$ $D_1$ $P_3$ $P_4$</CP>

$CPD_2$: <CP>$P_1$ $D_2$ $P_5$ $P_6$</CP>

$CPJ_1$: <CP>$P_4$ $P_5$ $J_1$ $P_7$</CP>

Applying the rule 3 for $CPD_1$ and $CPJ_1$ we obtain:

$CPJ_2$: <CP><CP>$P_2$ $D_1$ $P_3$ $P_4$</CP> $P_5$ $J_1$ $P_5$</CP>

Applying the rule 3 for $CPD_2$ and the computed $CPJ_2$ we obtain the composed part of the silhouette:

<CP><CP>$P_2$ $D_1$ $P_3$ $P_4$</CP> <CP>$P_1$ $D_2$ $P_5$ $P_6$</CP> $J_1$ $P_6$</CP>

**Rule 4**: If $P_i$ appears as a principal 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 $CPD_j$.

For example, the two composed parts written for the silhouette of figure 14.d are:

$CPD_1$: <CP>$P_1$ $D_1$ $P_2$ $P_3$</CP>

$CPD_2$: <CP>$P_2$ $D_2$ $P_4$ $P_5$</CP>

The substitution of $P_2$ in $CPD_1$ by the expression of $CPD_2$ gives the description of the silhouette:

<CP>$P_1$ $D_1$ <CP>$P_2$ $D_2$ $P_4$ $P_5$</CP> $P_3$ </CP>

All of other cases do not occur.

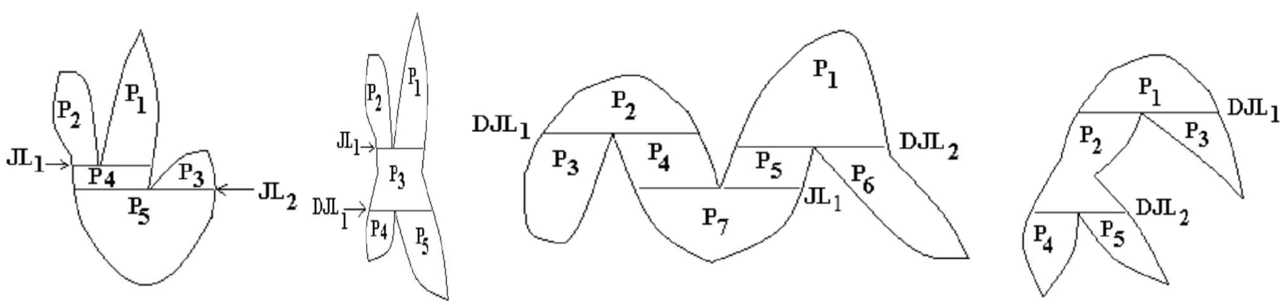



**Figure 14: The four cases of composed part**

### 3.3. Grammar of the Language XLWDOS

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:

$V_N$ = {¡Silhouette¿, ¡Composed-Part¿, ¡Part¿, ¡List-Parts¿, ¡Left-Boundary¿, ¡Right-Boundary¿, ¡Contour-Descriptor¿, ¡Type¿, ¡Length¿, ¡Junction-Line¿, ¡Inclination-Angle¿, ¡Disjunction-Line>, ¡Convexity-Degree¿, ¡Concavity-Degree¿, ¡Segment-Description¿, ¡High-Part-Number¿, ¡Low-Part-Number¿, ¡NumPart¿, ¡NumJunction¿, ¡NumDisjunction¿, ¡Segments¿}

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:

For example, the following text corresponds to the XLWDOS descriptor of the silhouette illustrated by figure 15.

```
<DXLWDOS><Name>Silhouette 1></Name>
<CP>
<CP><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>
    <J1> s P1 P3 57 w P3 2 s P2 P3 147 </J1>
```

**Table 2**
**Grammar of XLWDOS language**

```
¡Silhouette¿ → <DXLWDOS>
               <Name> “Silhouette name” </Name>
               <CP> ¡Part¿ </CP>
               </DXLWDOS>
¡Part¿ → ¡Composed-Part¿/
         <P ¡NumPart¿><L>¡Left-Boundary¿</L><R>¡Right-Boundary¿</R> </P ¡NumPart¿>
¡Composed-Part¿ → ¡Part¿ ¡List-Parts¿ ¡Junction-Line¿ ¡Part¿/
                  ¡Part¿ ¡Disjunction-Line¿ ¡Part¿ ¡List-Parts¿
¡List-Parts¿ → ¡Part¿ ¡List-Parts¿ / ¡Part¿
¡Left-Boundary¿ → ¡Contour-Descriptor¿ ¡Left-Boundary¿ / ¡Contour-Descriptor¿
¡Right-Boundary¿ → ¡Contour-Descriptor¿ ¡Right-Boundary¿/ ¡Contour-Descriptor¿
¡Contour-Descriptor¿ → ¡Type¿ ¡Inclination-Angle¿ ¡Length¿
¡Type¿ → cv ¡Convexity-Degree¿/cc ¡Concavity-Degree¿ /r
¡Inclination-Angle¿ → INTEGER VALUE
¡Length¿ → INTEGER VALUE
¡Junction-Line¿ → <J ¡NumJunction¿> ¡Segments¿ </J ¡NumJunction¿>
¡Disjunction-Line¿ → <D ¡NumDisjunction¿> ¡Segments¿ </D ¡NumDisjunction¿>
¡Segments¿ → ¡Segment-Description¿ ¡Segments¿/ ¡Segment-Description¿
¡Segment-Description¿ → s ¡High-Part-Number¿ ¡Low-Part-Number¿ ¡Length¿/
                        w ¡Low-Part-Number¿ ¡Length¿/ h ¡High-Part-Number¿ ¡Length¿
¡High-Part-Number¿ → INTEGER VALUE
¡Low-Part-Number¿ → INTEGER VALUE
¡Convexity-Degree¿ → INTEGER VALUE
¡Concavity-Degree¿ → INTEGER VALUE
¡NumJunction¿ → INTEGER VALUE
¡NumDisjunction¿ → INTEGER VALUE
¡NumPart¿ → INTEGER VALUE
```

```
      <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>
   </CP>
   <D1> h P3 27 s P3 P4 50 h P3 2 j P3 P5 66 </D1>
   <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>
</CP>
</DXLWDOS>
```

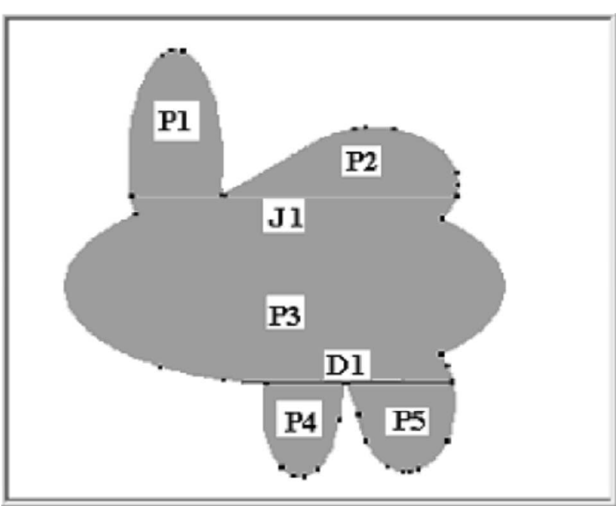



**Figure 15: Result of the decomposition of silhouette of figure 1**

### 3.4. Advantages of the XML Description of Silhouettes

The written description with an XML format gives the following advantages relatively to the known methods of silhouettes description:

#### *3.4.1. Efficiency in the Space Memory Occupied*

Let’s estimate the size of XML descriptors of silhouette (S). Let:

– NP be the number of parts of (S),
– NPL be the number of separating lines of (S),
– NCPP be the average number of curvature points for each part of (S),
– NCPS be the average number of curvature points of (S).

Let’s compute the size of XLWDOS descriptors.

– In the unfavorable case, each elementary contour of each part is a concave or convex curve. Thirteen characters code the description of each one implying that the size of the description of the part is equal to: (NCPP + 1) × 13 + 25 characters, where 25 are characters corresponding to the marks: <Pxx> <L> </L> <R> </R> </Pxx>.
– The number of segments of all partitioning lines is (NP – 1) × 3 considering the unfavorable case where each part has three segments in the correspondent separating line. The description of each segment is written using 13 characters (x Pxx Pxx xxx), then the separating lines are described using (NP – 1) × 3 × 13 + NPL × 11 characters, where 11 is the number of characters of the marks: <Jxx> </Jxx> or <Dxx> </Dxx>.
– The number of composed parts is equal to the number of separating lines, then they will necessitate NPL × 9 characters, where the number 9 corresponds to the tags: <CP> </CP>.
– The header size of XLWDOS descriptor is equal to 32 bytes,

The size of the XLWDOS descriptor is then equal to:

Size = NP × (NCPP + 1) × 13 + 25 + (NP – 1) × 3 × 13 + NPL × 11 + NPL × 9 + 32

Size = 13 NP × NCPP + 52 NP + 18 + 20 NPL

Size = 13 × NCPS + 53 NP + 18 + 20 NPL

The memory space is a linear function of the number of curvature points of the silhouette (NCPS).

The first advantage of the proposed method is then compactness in storage space: descriptors of silhouettes are stored using few hundreds of bytes. More results are given in fifth section.

### *3.4.2. Easiness for Indexation*

Another advantage is that stored information is structured into elements (parts, junction and disjunction lines) and relations between these elements. From the XML descriptor it is easy to extract the following elements:

– Number of parts: computed as the number of <L></L> or <R></R>
– Numbers of junction and disjunction lines: computed as the number of <J></J> and <D></D>
– Position order in the silhouettes of junction and disjunction lines
– Number of parts neighboring to each junction or disjunction lines
– Position of parts onto the separating lines which is indicated by the list of characters s, w, and h as written in the descriptor.

These elements constitute the first index that permit to differentiate silhouettes. A second index that must be extracted from the textual descriptor is actually investigated and concerns the geometry of outline contour.

## 4. DESCRIPTION OF SHAPES

### 4.1. Basic Principle of the Method

The use of silhouettes descriptions for object recognition isn’t sufficient in many cases. When shapes contain internal regions corresponding to specific texture or color, this can facilitate the recognition process. The set of internal regions constitutes additional information that allows the obtaining of faithful description for shapes and more information for shape comparison in the recognition process of objects from their 2D images.

We assume that image is segmented into regions corresponding to shapes and internal regions that they encompass. Internal regions 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 figure 16).

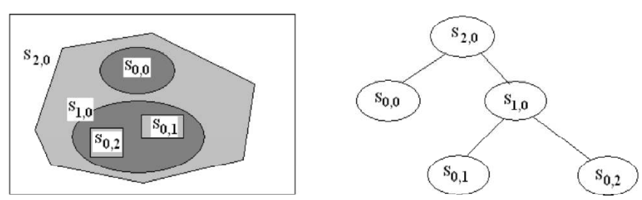



**Figure 16: Shape parsing into internal regions and the associated tree structure**

We associate for each region the notation $S_{ij}$ where:

– i corresponds to its position (level) in the tree beginning from the bottom to the up. This value begins from zero.
– j corresponds to its order number relatively to region having the same level. This value beginning from 0 is given to regions located sweeping top-bottom and left-right the image.

Thus, the level zero (0) is given to regions that not contain internal regions. The level one (1) is given to regions that contain regions of level 0. 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 figure 16 contains for example three regions of level 0 ($S_{0,0}$, $S_{0,1}$, $S_{0,2}$) and one region of level 1 ($S_{1,0}$) encompassing $S_{0,1}$, $S_{0,2}$ and one region $S_{2,0}$ of level 2 containing $S_{1,0}$ and $S_{0,0}$.

The tree structure can be automatically achieved starting from a segmented image, by the location on the image all elementary regions that are regions not encompassing other internal regions. For each of these regions we associate the level 0. The next step is the location of regions of level 1 that are regions encompassing regions of level 0. We repeat this treatment in order to locate regions of level i corresponding 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.

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. Figure 17 illustrates an example where the new region $S_{1,0}$ is the region obtained as the union the neighboring regions $S_{0,1}$, $S_{0,2}$, $S_{0,3}$ and $S_{0,4}$.

At each node of the tree structure we must add a set of attributes describing the outline region (silhouette), the position and orientation of the rectangle of minimum area RM encompassing it. For silhouette description, we use the language XLWDOS presented in the third section, whereas the position and orientation of RM are respectively the position of its highest left corner and the inclination angle aligning its width with image rows (see figure 18).

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

**Algorithm**

Begin

– Locate all silhouettes

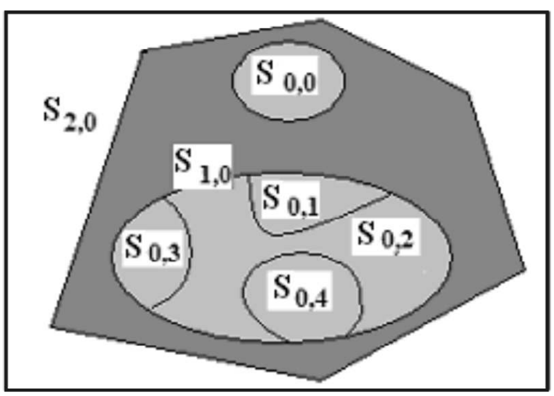
For each elementary silhouette (that not encompasses other silhouettes)

Do

- Compute the associated minimum rectangle RM
- Compute the XLWDOS descriptor
- Save the X and Y-coordinates of the highest left corner of RM
- Save the rotation angle b aligning the width of RM with image rows
- Level-of-silhouette=0

EndDo

Current level =1



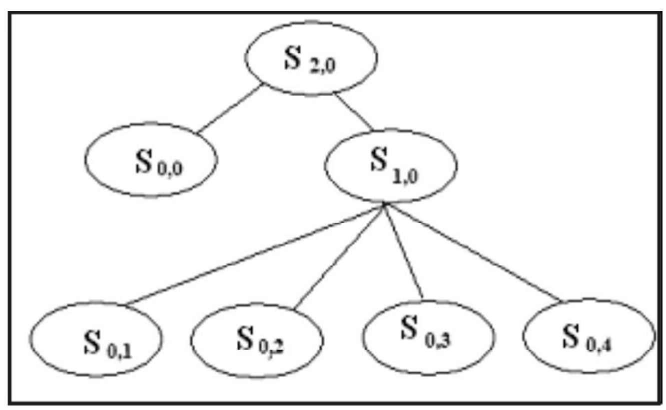
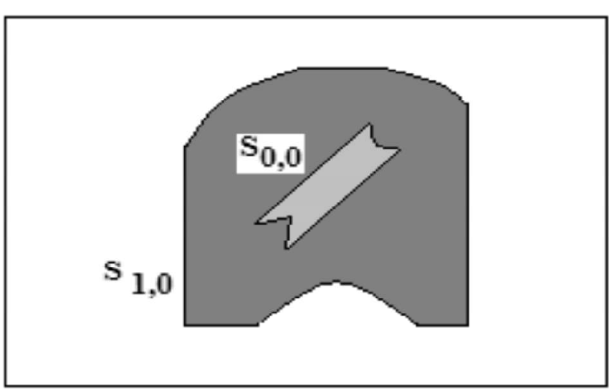
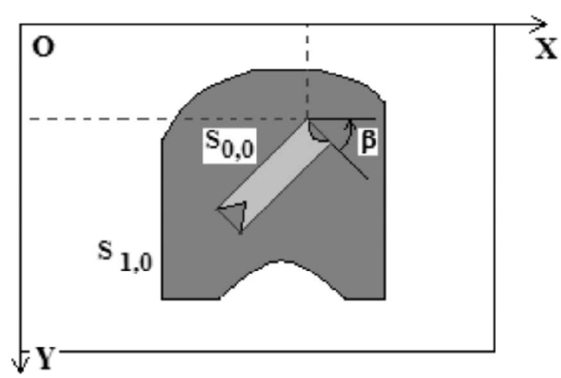




**Figure 17: Case of neighboring internal regions**




(a)




(b)

**Figure 18: (a) Initial shape, (b) minimum rectangle of internal region**

Repeat

While there is silhouette encompassing silhouettes verifying

Level-of-Silhouette < Current level

Do

- Compute the associated minimum rectangle RM
- Compute the XLWDOS descriptor
- Save the X and Y coordinates of the highest left corner of RM
- Save the rotation angle b aligning the width of RM with image rows
- Level-of-silhouette= Current level

EndDo

Current level = Current level +1

Until all silhouettes have been processed

End

### 4.2. An XML 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. In addition to the rules of the language XLWDOS, we must add rules in order to describe the inclusion of silhouettes.

Shape may be a silhouette without internal silhouettes or contains at different levels internal silhouettes. The following syntax is used to describe recursively this inclusion:

<Shape>

Position of the highest left corner of the minimum rectangle encompassing the outline shape Orientation of the associated minimum rectangle

XLWDOS Descriptor of the outline shape

<IS>

Position of the highest left corner of associated minimum rectangle

Orientation of the associated minimum rectangle

XLWDOS Description of the outline shape

</IS>

……

<IS>

Position of the highest left corner of associated minimum rectangle

Orientation of the associated minimum rectangle

XLWDOS Description of the outline shape

</IS>

</Shape>

Recursively, if any internal region contains other internal regions, we use the same syntax to describe its internal regions.

The grammar of the language XLWDS is the same as of XLWDOS with the following additional rules:

**Table 3**
**Additional rules of the grammar of XLWDS language**

S → <DXLWDS> <Name> “name of the shape” </Name>
¡Shape¿
</DXLWDS>

¡Shape¿ → <Shape> ¡Position¿ ¡Orientation¿¡Silhouette¿ ¡List-IS¿ </Shape>

¡List-IS¿ → ¡IS¿ ¡List-IS¿/ ¡IS¿

¡IS¿ → <IS><Name> “name of internal shape” </Name> ¡Position¿ ¡Orientation¿ ¡Silhouette¿
</IS> /
<IS> <Name> “name of internal shape” </Name>
¡Position¿ ¡Orientation¿ ¡Silhouette¿ ¡List-IS¿
</IS>

¡Position¿ → ¡X-Position¿ ¡Y-Position¿

¡X-Position¿ → INTEGER VALUE

¡Y-Position¿ → INTEGER VALUE

¡Orientation¿ → INTEGER VALUE

### 4.3. Properties of LWDS Descriptors

In addition to the properties of XLWDOS descriptors that also verified here, each internal region is described independently of the shape where it appears. Consequently the identification process may recognize it independently and this will contributes in the recognition of the shape.

Another property is the easiness for extraction of index from the LWDS descriptor constituted by the number of internal regions, the index XLWDOS of each region, and the tree that corresponds to the structure of the set of regions.

## 5. EXPERIMENTATION

### 5.1. Silhouettes Extraction and Computation of XLWDOS Descriptors

In our work we assume that silhouettes are located using image segmentation. A Graphic interface was developed with Visual C++ under Windows 2000XP for the computation and visualization of XLWDOS descriptors of silhouettes. Once the image is loaded, the following steps are performed:

– Computation of the rectangle of minimum area RM encompassing the silhouette
– 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 $\beta_W$ aligning its width, allows the sweep of the silhouette in the length direction of RM. The second rotation with angle $\beta_L$ aligning its length, allows the sweep of the silhouette in the width direction of RM.

– Decomposition of the silhouette after rotation into parts, separating lines and curvature points. Chetverikof's algorithm [7] is used for the location of points of high curvature (see annex 1) that uses two parameters: angle á and distance d. After this, the program computes the attributes of different contours and displays the XLWDOS descriptor and the partitioning result.

In order to locate the outline shapes in image, a good segmentation technique is required.

We used the software provided for research purposes by Y. Deng, and B. S. Manjunath [10] where the essential idea of segmentation method (JSEG) is to separate the segmentation process into two independently processed stages, color quantization and spatial segmentation. In the first stage, colors in the image are quantized to several representing classes that can be used to differentiate regions in the image. This quantization is performed in the color space alone without considering the spatial distributions. Afterwards, image pixel colors are replaced by their corresponding color class labels, thus forming a class-map of the image.

Figures 19.a 19.b illustrate image of a car (toy) and the correspondent silhouette is extracted using the segmentation method JSEG [10].

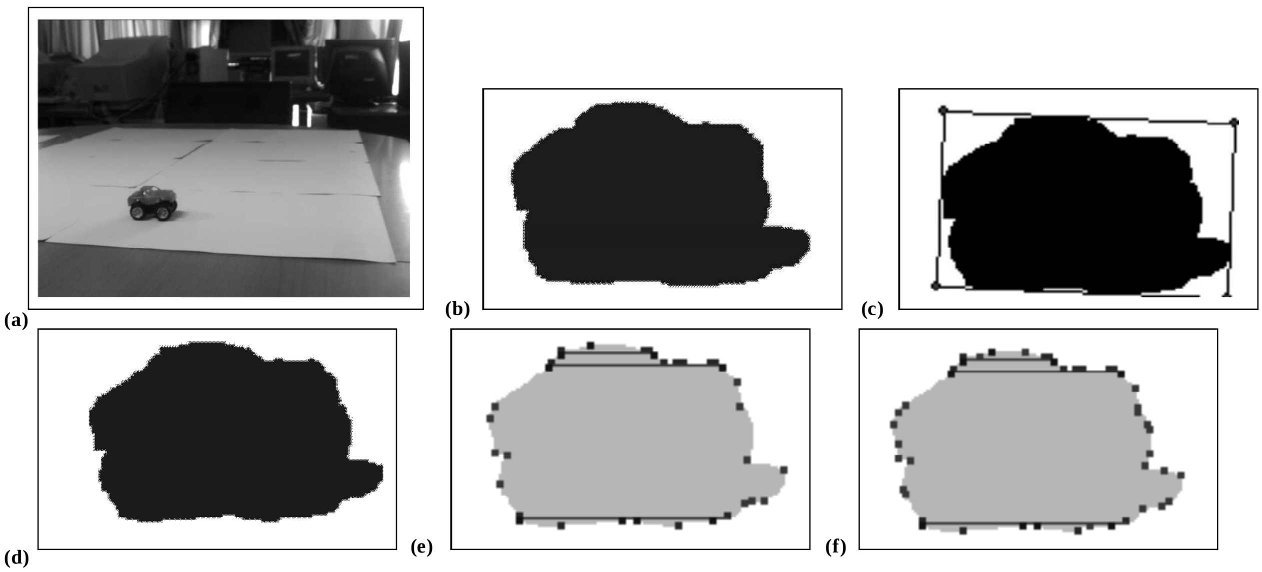



**Figure 19: (a) Image of moving car, (b) Extracted silhouette, (c) The associated minimum rectangle, (d) rotation of the silhouette (e) Silhouette decomposition for d = 5, $\alpha$ = 150°, (f) Silhouette decomposition for d = 5, $\alpha$ = 160°**

Figure 19.c illustrates the rectangle of minimum area RM encompassing the silhouette of the car. The rotation of the silhouette around the highest left corner of RM are performed with $\beta_L = -5.33°$ (see figure 19.d). We illustrate in the same figure the results of its decomposition using two different values of the parameter $\alpha$, the value of d is 5 pixels. The size of computed XLWDOS descriptors is 731 bytes for $\alpha$ = 150° and 815 bytes for $\alpha$ = 160°.

Figures 20 and 21 illustrate the case where the silhouette is extracted after image segmentation into regions using JSEG method [10]. We illustrate by this figure the obtained results applying the same steps described above for the computation of XLWDOS descriptors.

The second group of images used for the experimentation is the database of shapes of ETH80 of B. Leibe and B. Schiele [17]. We present in table 4 for some silhouettes the results of their decomposition.

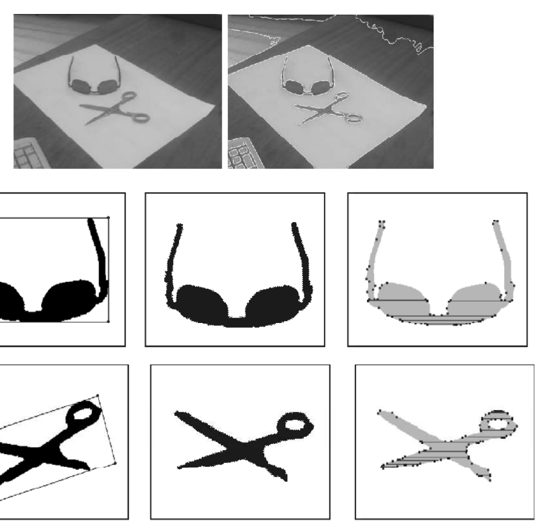


**Figure 21: The same steps applied for the computation of XLWDOS descriptors for the two located silhouettes (Glasses and scissors)**

**Table 4**
**Computation of XLWDOS descriptor for some images of THI database [17]**

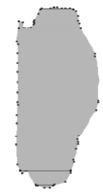
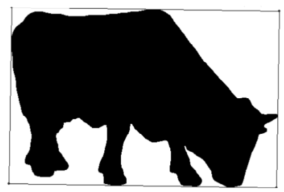
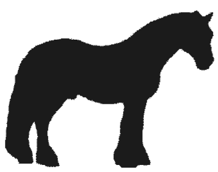
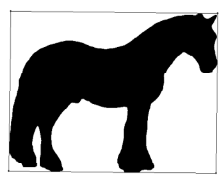
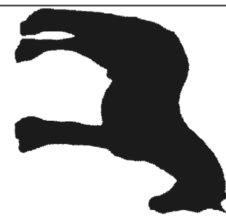
| Silhouette | Minimum Rectangle | Silhouette after rotation | Decomposition |
|---|---|---|---|
|  car |  |  +87.18° |  |
|  cow |  |  β= +89.08 |  |
|  horse |  |  β= +89.27° |  |

**Table 5**
**Comparison of the size of different formats**

| | *Y-XLWDOS, X-XLWDOS Descriptors* $\alpha = 150°$ | *Y-XLWDOS, X-XLWDOS Descriptors* $\alpha = 160°$ | *BMP Format* | *GIF Format* | *JPG Format* | *Scalar Format* |
|---|---|---|---|---|---|---|
| Car | 673 Bytes<br>731 Bytes | 745 Bytes<br>815 Bytes | 12 KB | 2 KB | 6 KB | 149 KB |
| Cup | 632 Bytes<br>854 Bytes | 897 Bytes<br>1.09 KB | 15 KB | 3 KB | 7 KB | 156 KB |
| Glasses | 1.18 KB<br>1.36 KB | 1.13 KB<br>1.35 KB | 8 KB | 2 KB | 6 KB | 144 KB |
| Scissors | 766 Bytes<br>1.02 KB | 1.03 KB<br>1.06 KB | 9 KB | 2 KB | 2 KB | 145 KB |
| Cow | 795 Bytes<br>920 Bytes | 1.26 KB<br>1.44 KB | 28 KB | 4 KB | 13 KB | 184 KB |
| Horse | 1.77 KB<br>1.68 KB | 1.68 KB<br>2.20 KB | 29 KB | 3 KB | 11 KB | 186 KB |

## Discussion

The major results of the proposed method are:

– Reduced size of computed descriptors. Indeed, the size of silhouettes varies from 100 to 2000 bytes according the geometry of the outline: more curvature points imply the increase of descriptor size.

We give in table 5 the size of the two computed XLWDOS descriptors for some silhouettes and their size in various formats. We can see that XLWDOS descriptor is interesting for the occupied space memory. This space may be decreased if the XLWDOS descriptors are computed after smoothing the outline shape with a Gaussian filter [20].

– The semantic contained in the XLWDOS description, such as the number of parts, of junction and disjunction lines. The set of marks of the descriptor constitutes a good index in for database of silhouettes.

### 5.2. Computation of XLWDS Descriptors

Firstly, we explain how the XLWDS descriptors of shape are computed. Thereafter, we present some results obtained applying our method over images of interior and exterior scenes.

Starting from segmented image, the following steps are performed:

– Computation for each region of the image, its level value (as explained in subsection 5.1) and its order number sweeping image in the direction top-bottom and left-right

– A tree structure is created representing the organization of different regions.

– For each region of the tree, the rectangle of minimum area encompassing it is computed. The X and Y-coordinates of the highest left corner, and the rotation angle β allowing the aligning of its width with the rows of the image are saved.

– The region is then rotated with the angle β and the XLWDOS descriptor is computed.

– The ultimate step is the writing of all computed data of the tree following the syntax of XLWDS language.

### Computation of XLWDS Descriptors

To validate our method images of 3D objects are used. Figure 22 illustrates image of a car (toy). Shape corresponding to the object is extracted using JSEG method [10].

**Figure 22:** Image of a car and the segmented image

Figure 23 illustrates the software is developed that start from the segmented image, compute and visualizes.

– the different regions using different colors (the background (Bg) is also colored),

– the tree structure correspondent

– the codification of different colors. When a new region is added as the union of neighboring region a new color not appearing in colored regions is associated.

– The XLWDOS descriptor of each region using the two parameters of Chetverikov's algorithm [7]: the angle a and the distance d.

– The XLWDS descriptor of the shape

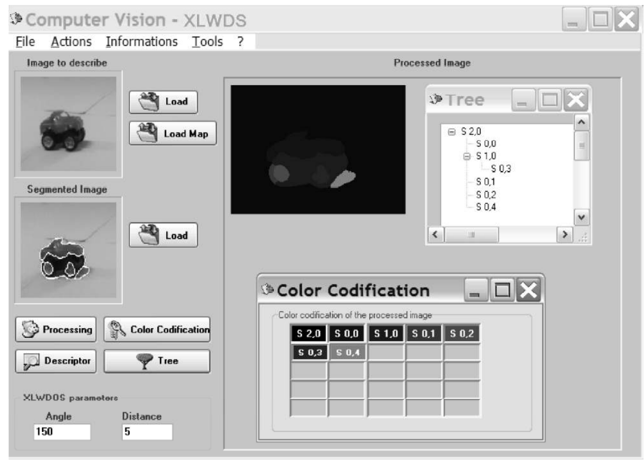



**Figure 23:** Graphical results of XLWDS descriptor computation for the car image

The first step is the location of regions of level 0. Four regions are located $S_{0,0}$, $S_{0,1}$, $S_{0,2}$, $S_{0,3}$ and $S_{0,4}$. The second step is the location of regions of level 1. There is one region $S_{1,0}$ that encompasses $S_{0,3}$. The third step is the location of $S_{2,0}$ of level 2 corresponding to the outline shape that encompasses all located regions. After this, for each region the minimum rectangle RM encompassing it is computed, the X and Y-coordinates of the highest left corner of RM and the value of the angle â are saved. Figure 24 illustrates this process for the S2,0 shape.

The XLWDOS descriptors are computed using the rotated region. All data computed are used producing the following XML file.

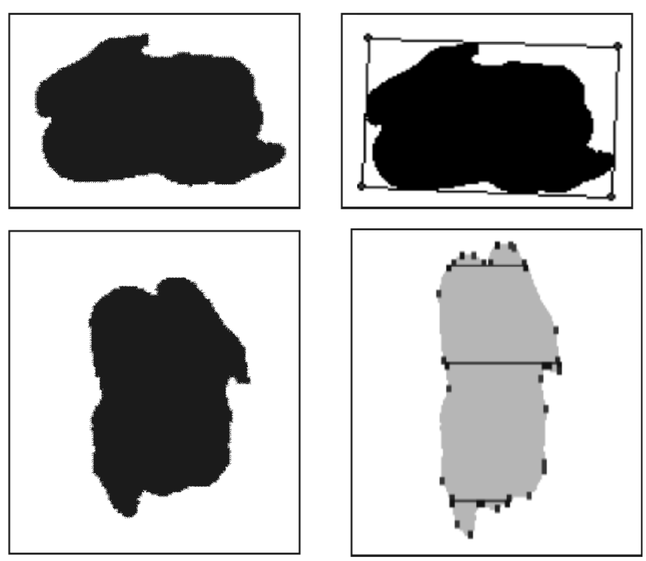


**Figure 24:** For S2,0 region, the minimum rectangle, rotated region, the result of XLWDOS description

<DXLWDS>

<Shape>

<Name> S2,0 </Name> 46 57 87.58 XLWDOS descriptor of S2,0 <IS>

<Name> S0,0 </Name> 141 55 -85.10 DXLWDOS of S0,0</IS>

<IS><Name> S1,0 </Name> 52 87 90 DXLWDOS of S1,0

<IS><Name> S0,3 </Name>111 102 57.09 DXLWDOS of S0,3</IS>

</IS>

<IS><Name> S0,1 </Name> 56 101 6.34 DXLWDOS of S0,1</IS>

<IS><Name> S0,2 </Name> 62 100 42.71 DXLWDOS of S0,2</IS>

<IS> <Name> S0,4 </Name> 153 106 -63.43 DXLWDOS of S0,4</IS>

</Shape>

</DXLWDS>

Figure 25 and 26 illustrate other examples of shape decomposition and representation by a tree structures of silhouettes.

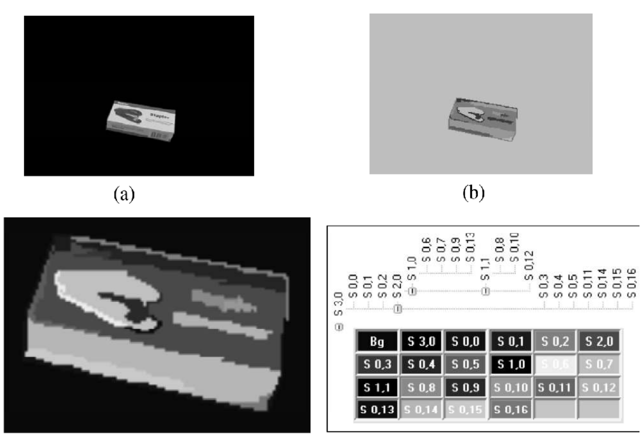
### 5.3. Discussion

The obtained results in conducted experimentation allow us to assert that the proposed descriptor for shapes has the following advantages comparatively to other methods:

– the description of interior regions of shapes

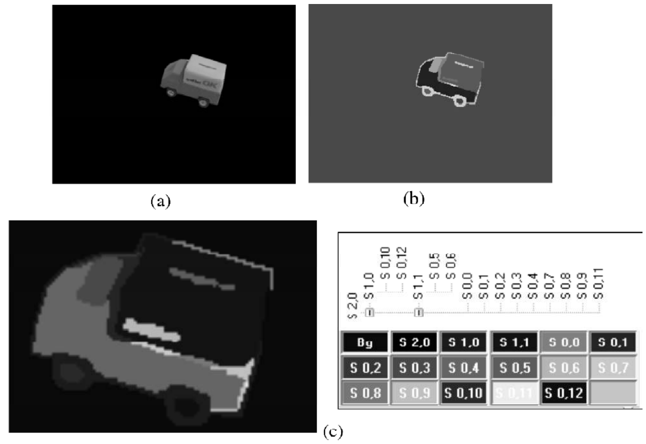
– a textual marked descriptor is computed instead of numerical values

– structured information is produced allowing the easiness for indexation and comparison of shapes descriptors



(a) (b) (c)

**Figure 25: (a) Initial image of a box, (b) the segmented image, (c) color codification of located shapes and associated tree**



(a) (b) (c)

**Figure 26: (a) Initial image of a car, (b) the segmented image, (c) color codification of shapes of the car and associated tree**

The major difficulty for the application of our method is the extraction of the shape and its internal regions. Robust segmentation method is then required for computer vision applications.

## 6. CONCLUSION

In this paper we have presented a new method for shape representation and description. In the first, we have explained in section 2 the method for describing outline shape using parts, junction line and disjunction line. An XML language is proposed in section 3 for writing descriptors of outline shapes. We have proposed in section 4 a new 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 invariant for scale change and rotation. We presented also the XML language noted XLWDS that permit to write textually these descriptors. In section 6 are presented results obtained applying the proposed methods over real images

Another interesting application of our representation method is the coding of images using the XLWDS language. The XML text generated can be used for image coding and visualization.

**REFERENCES**


[1] N. Arica and F. T. Y. Vural, BAS: a perceptual shape descriptor based on the beam angle statistics, Pattern Recognition Letters, **24**(9-10), 2003, 1627-1639.

[2] O. E. Badawy, M. Kamel, Shape representation using concavity graphs, Proceeding of the 16th International Conference on Pattern Recognition, 2002.

[3] S. Belongie, J. Malik, J. Puzicha, Shape Matching and Object Recognition Using Shape Contexts, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 2002.

[4] T. Bernier, J. A. Landry, A new method for representing and matching shapes of natural objects, *Pattern Recognition* **36**, 1711-1723, 2003.

[5] S. Berretti, A. Bimbo, Retrieval by shape similarity with perceptual distance and effective indexing, Transactions on Multimedia, **2**(4), 2000.

[6] R. J. Campbell and P. J. Flynn, A Survey of Free-Form Object Representation and Recognition Techniques, *Computer Vision and Image Understanding*, **81**, 166-210, 2001.

[7] D. Chetverikov: A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves. 10th International Conference, CAIP 2003, Groningen, The Netherlands, August 25-27, 2003.

[8] T. M. Cronin, Visualizing concave and convex partitioning of 2D contours, *Pattern Recognition Letters*, **24** (2003), 429-443.

[9] C. M. Cyr and B. B. Kimia, A similarity-based aspect-graph approach to 3D object recognition, *International Journal of Computer Vision*, **57** (1), 5-22, 2004.

[10] Y. Deng, and B. S. Manjunath, Unsupervised segmentation of color-texture regions in images and video, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, August 2001.

[11] P. F. Felzenszwalb, D. P. Huttenlocher, Pictorial structures for object recognition, *International Journal of Computer Vision*, **61**(1), 55-79, 2005.

[12] D. Geiger, T. Liu, and R. V. Kohn, Representation and Self-Similarity of shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**(1), 2003.

[13] W. I. Grosky, R. Mehrota, Index-based recognition in pictorial data management. *Computer Vision Graphics and Image Processing*, **52**, 416-436, 1990.

[14] D. H. Kim, I. D. Yun, S. U. Lee, A new shape decomposition scheme for graph-based representation, *Pattern Recognition*, **38**, 673-689, 2005.

[15] J. J. Koenderink and V. Doorn, the internal representation of solid shape with respect to vision, *Biol. Cyber.*, **32**, 1976.

[16] S. Larabi, S. Bouagar, F. M. Trespaderne, E. L. Fuente, LWDOS: Language for Writing Descriptors of Outline Shapes, Lecture Notes in Computer Science, Proceeding of Scandinavian Conference on Image Analysis, June 29 - July 02, Gotborg, Sweden, 2003, Volume 2749, Pages 1014-1021.

[17] B. Leibe and B. Schiele, Analyzing appearance and contour based methods for object categorization, Proceedings of the *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[18] T. Lourens, R. P. Wurtz, Object recognition by matching symbolic edge graphs, In the proceeding of ACCV' 98, pp. 193-200.

[19] P. A. Maragos and R. W. Schafer, Morphological skeleton representation and coding of binary images, *IEEE transactions on Acoustic, Speech Signal Processing*, **34**(5), 1986.

[20] 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), 789: 805, 1992.

[21] F. Mokhtarian, Silhouette-Based isolated object recognition through curvature scale space, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(5): 539-544, 1995.

[22] H. Murase, S. K. Nayer, Visual learning and recognition of 3-D objects from appearance, *International Journal of Computer Vision*, **14**(1), 1995.

[23] R. C. Nelson, A. Selinger, large-scale tests of a keyed, appearance-based 3-D object recognition system, *Vision Research*, **38**(15-16), 1998.

[24] C. Orrite and J. E. Herreo, Shape matching of partially occluded curves invariant under projective transformation, *Computer Vision and Image Understanding*, **93**(1), 2004.

[25] 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.

[26] I. Pitas and A. N. Venetsanopoulos, Morphological shape decomposition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(1), 38-45, 1990.

[27] P. L. Rosin, Shape partitioning by convexity, British Machine Vision Conference 1999.

[28] C. Ruberto, Recognition of shapes by attributed skeletal graphs, *Pattern Recognition*, **37**(2004), 21-31.

[29] 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.

[30] 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), 73-86, 2004.

[31] Z. Shao, J. Kittler, Shape representation and recognition based on invariant unary and binary relations, *Image and Vision Computing*, **17** (1999), 429-444.

[32] K. Siddiqi, B. B. Kimia, Parts of visual form computational aspects, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(3), 239-251, 1995.

[33] K. Siddiqi and B. B. Kimia, A shock grammar for recognition, *Conference of Computer Vision and Pattern Recognition*, 1996.

[34] Z. Tu, X. Chen, A. L. Yuille, S. Zhu, Image parsing: Unifying Segmentation, Detection, and Recognition, *International Journal of Computer Vision*, **63**(2), 113-140, 2005.

[35] M. H. Yu, C. C. Lim, J. S. Jin, Shape similarity search using XML and portal technology, *Workshop on visual information processing*, VIP 2003, Sydney.

[36] L. Yu, R. Wang, Shape representation based on mathematical morphology, *Pattern Recognition Letters*, 2004.

[37] D. Zhang and G. Lu, Review of shape representation and description techniques, *Pattern Recognition*, **37**(2004), 1-19.

[38] S. C. Zhu and A. L. Yuille, FORMS: A flexible object recognition and modeling system, *Fifth International Conference on Computer Vision*, M.I.T. Cambridge, 1995.