Comparison of Detailed Descriptors of Noisy Silhouettes

Saliha AOUAT [1] **Slimane LARABI** [1]

[1] *LRIA, Computer Science Department,*
University of Sciences and Technology – Houari Boumediene
Algiers, Algeria
{saouat, slarabi}@usthb.dz,
http://www.usthb.dz

Abstract

In this paper we propose a new method for silhouettes comparison. Silhouettes of 3D objects, extracted from 2D images, are described according to LWDOS language [1]. Sometimes, even silhouettes appear identical their LWDOS detailed descriptors may be very different therefore we wrongly conclude that the two silhouettes are different. In order to resolve such problem due to several phenomenons, we try to deduce the same detailed descriptor from the two slightly different descriptors of two silhouettes.

Keywords

Detailed Descriptor, Silhouette, LWDOS, comparison

1. Introduction

Several methods of image matching, indexing, categorization and recognition exist in the literature. Despite these methods use different representations, they all proceed on feature comparison at several levels of abstraction. 3D objects in our method are represented by their silhouettes, which are described by a symbolic text [1].

Several 3D objects representations from 2D images have been proposed in the literature. Volumic primitives that contain geometric and photometric informations [2,3], 3D boundaries rebuilt from stereovision [4], appearance-based representation which use objects aspects for retrieval and recognition [5,6,7], 2D boundary curves of silhouettes using Curvature Tuned Smoothing (CTS) or Curvature Scale Space (CSS) [8, 9, 10,11], Other methods use invariants describing objects for image indexing and recognition [12, 13, 14,15].

In this work, we use the part-based method published in [1] for the representation of an object from a number of its silhouettes.

Silhouettes are important features for object recognition in the natural vision system [16] and are therefore used in several artificial systems for object recognition and representation [17,18,19].

Silhouettes are described in the literature by languages developed and used to solve specific problems. Among them, the FOHDEL language [20] and more recent research [21,22] are devoted to the handwritten symbol.

Others, as the LWDOS language [1], are more specific to a geometric object description. As we are dealing with 3D object recognition and matching, we used the LWDOS language to solve the matching problem.

As a little variation about the positions of the objects or exterior phenomenon such as noise may generate noisy silhouettes, the recognition and matching processes cannot be addressed before the problem of silhouettes description has been treated.

The matching process necessitates at least two descriptions (global and detailed descriptions) and takes into account distortions due to noise. Noise may modify both descriptions. In this paper we have developed a method to correct detailed descriptors after the modelisation of silhouette distortion.

The paper is organized as follow: A first we introduce the silhouette description according to LWDOS language in the second section. In the third section we explain the effect of noise when comparing LWDOS descriptors. Distortion of detailed descriptors will be presented in the fourth section. We propose a comparison model in the fifth section. The sixth section shows our experimental results. And in the appendix in the end of the paper we give more details about LWDOS language.

2. Silhouettes Description according to LWDOS Language

LWDOS language [1] uses the minimum rectangle (MR) that encloses the outline shape [23]. (OXY) is the referential attached to MR chosen such as the origin O is the left top edge of MR and the OX (resp. OY) axis corresponds to the width (resp. length) of outline shape [1] (fig.1).

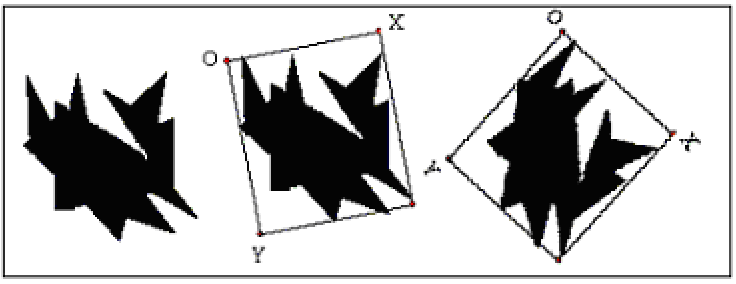


Fig. 1. Initial silhouette, the minimum rectangle MR encompassing it, and the rotated silhouette

From this geometric description, outline shape may be drawn without ambiguity implying the propriety of uniqueness and preservation of perceptual structure. The invariance of this description to rotation is guaranteed by the sweep up of the silhouette following one of the directions of the minimum rectangle encompassing it. All silhouettes descriptions mentioned in this paper are given relatively to such rectangle. In the following, horizontal direction is the (OX) direction, and the vertical direction is the

(OY) direction relatively to (MR). The Grammar and production rules of LWDOS language are given in the appendix at the end of the paper.

The first step of shape description is its decomposition into parts and separating lines. For this, we sweep the outline shape from top to bottom following the horizontal direction of the (MR), and we locate concave points for which the outer contour changes the direction top-bottom-top or bottom-top-bottom (fig. 2).

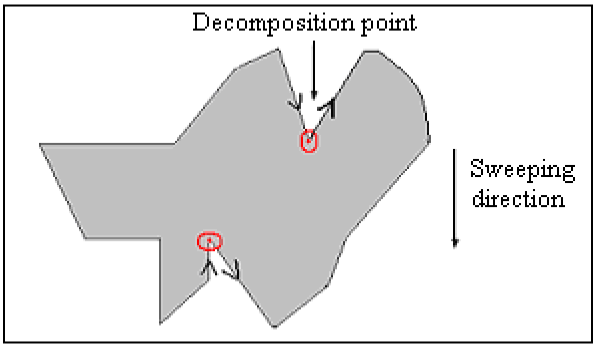


Fig.2. Location of decomposition points

The outline shape is decomposed at these points into parts, and separating lines. Two parts or more are joined with a third part through a junction line. One part is joined with two parts or more through a disjunction line. This process applied for example to silhouette of figure 2 produces five parts (P1 until P5), one junction line (JL1) and one disjunction line (DJL1). Parts and separating lines are numbered from top to bottom and left to right. (fig. 3)

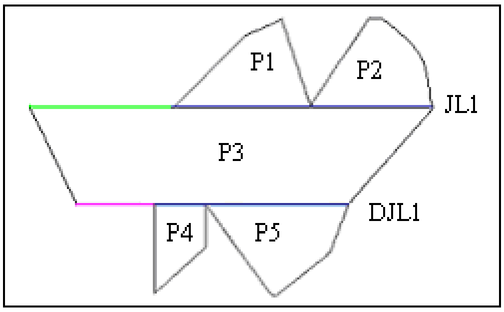


Fig.3. Parts, junction and disjunction lines

2.1 Global Descriptor

Global LWDOS descriptor of a silhouette written with the LWDOS language [1] is the descriptor for which appear only parts, junctions, and disjunctions lines without their geometric descriptions. For example, the global LWDOS descriptor of the silhouette in figure 3 is: [$P1 \cup P2 \oplus \uparrow JL1 \oplus P3 \oplus \downarrow DJL1 \oplus P4 \cup P5$].

JLi indicates the i^{th} junction line encountered when scanning the image from top to bottom, DJLj indicates the j^{th} disjunction line.

Different silhouettes may have the same global descriptor

2.2 Detailed Descriptor

The second step is the description of each element (each part and line) in order to guaranty the uniqueness of the outline shape representation.

A detailed description is based on the silhouette features geometry defined below:

- A part is described by its left and right boundaries.

- A boundary is a succession of primitives.

- A primitive may have one of the three following shapes: Convex curve (Cv), Concave curve (Cc), or a straight line segment (Ln) [1]. A primitive is characterized by its inclination angle and length.

- The length of a primitive refers to the high of the primitive, except when the primitive is horizontal (fig. 4).

- When the primitive is a convex curve (resp. concave curve), it is also characterized by its convexity degree (resp. concavity degree). (fig. 5).

- Length and width of an outline shape refers to the dimensions of its encompassing box which is the minimum rectangle (MR).

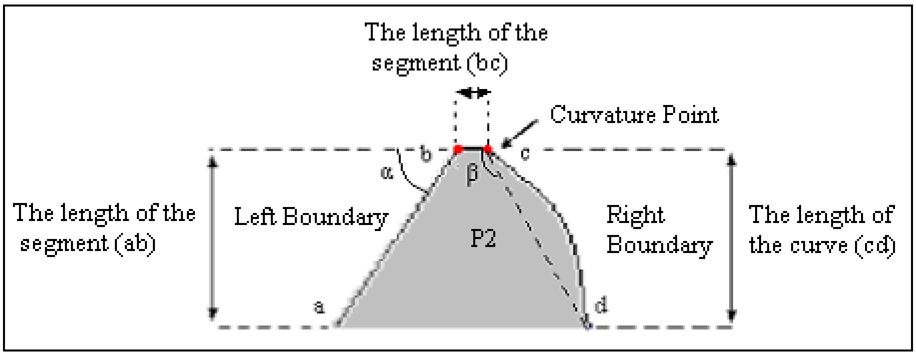


Fig.4. Detailed descriptor of a part

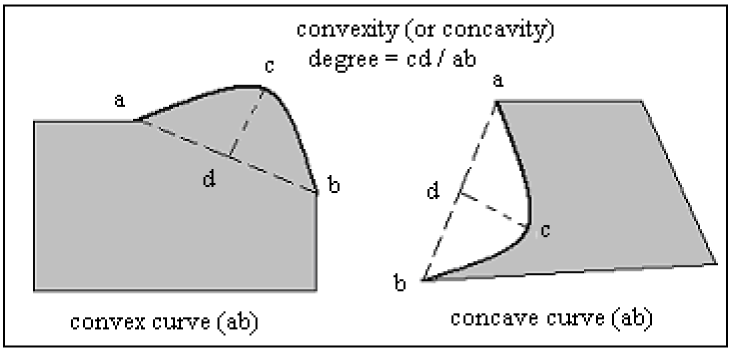


Fig.5. Convexity and concavity degrees

The left boundary of part P2 (fig. 4) is composed by a line segment with α =43° as inclination angle and 109 pixels as length. The right boundary is composed by a straight line segment of 180° as inclination (horizontal segment) and 17 pixels as length followed by convex curve (noted Cv) with 0.09 (9%) as degree of convexity, β=108° as inclination angle and 79 pixels as length. LWDOS language uses absolute dimensions for multimedia applications and relative dimensions for computer vision applications. In this paper we use relative dimensions relatively to MR such as is shown in an example (subsection 2.3). Relative dimensions are expressed as a percentage of the size of the (MR).

A contour of a shape is segmented into a set of elementary contours (primitives) by determining curvature points (inflexion points) located by one of the known algorithms [24]. The number of inflexion points is proportional to the description precision (fig. 6).

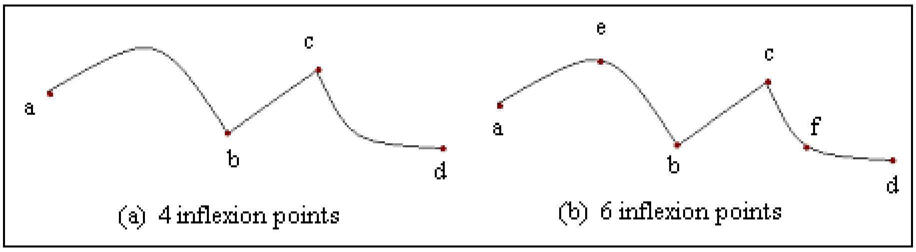


Fig.6. Inflexion points

Separating lines are decomposed into segments. Each segment is described with its type and length. Three types for a segment are possible: "Junction" if the segment is adjacent to two parts, "Free-High" if it is adjacent to the high part, and "Free-Low" if it is adjacent to the low part (fig. 7).

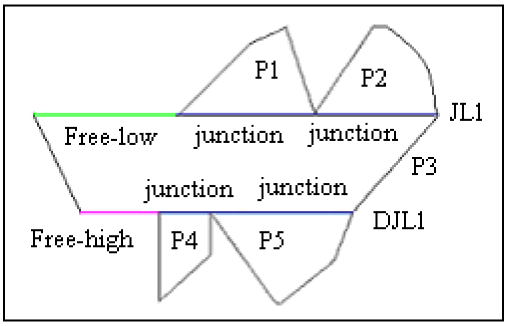


Fig.7. Description of separating lines

For example the junction line JL1 is decomposed into three segments: the first is "free-low segment" because it is adjacent to P3, the second is "junction segment" because it's adjacent to both P1 and P3, and the last one is junction segment because it's adjacent to P2 ad P3. The disjunction line DJL1 has three segments: free-high, junction, and junction.

The length of each segment is expressed as a percentage relatively to the minimum rectangle (MR).

2.3. Example

Let's consider the silhouette of the following object (fig. 8).

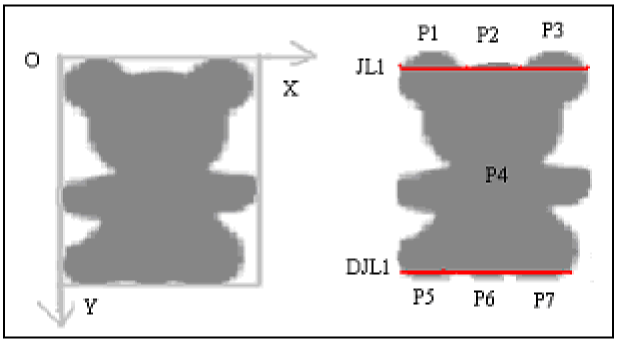


Fig 8. Silhouette decomposition

Step1: Obtain the outline shape components by sweeping the image from top to bottom: three parts (P1, P2, P3), one junction line JL1, one part P4, one disjunction line DJL1 and three parts (P5, P6, P7).

Step2: For each part, extract the boundaries features.

Step 3: Write the description of each part according to the syntax of LWDOS. The attributes *Very_short, short, less_short, middle, enough_long, long, very_long* and *entire* of relative length have been affected to the following intervals: [0 0.1],]0.1 0.2],]0.2 0.4],]0.4 0.55],]0.55 0.7],]0.7 0.85],]0.85 0.99] and [1 1]. The attributes *Weakly convex* or *Weakly concave, Convex* or *Concave, Strongly convex* or *Strongly concave* have been affected respectively for the conditions: (r/L<=¼), (¼<r/L<=¾) and (¾ < r/L). Where r and L correspond respectively to the lengths of segments (cd) and (ab) used to compute convexity degree such as shown in figure 5. The attributes: *Horizontal, Vertical, Weakly inclined, Inclined* and *Strongly inclined* have been affected respectively for the values and intervals of inclination angle .: 0º, 90º,] 0º, 30º],]30º, 60º],]60º, 90°[.

The following descriptions has been obtained: (The symbol “⊗ ” is a separator between left and right boundaries of a part).

P1→{*Wcv_Il* very_short ⊗ *Wcv_Ir* very_short}

P2→{*Wcv_Wil* very_short ⊗ *Wcv_Wir* very_short}

P3→ { *Wcv_Il* very_short ⊗ *Wcv_Ir* very_short}

P4→{ *Wcv_Ir* short *Wcv_Sir* less_short *Wcv_Il* very_short Cv_Ir short Ln_Il short *Cv_Vrt* short ⊗ *Wcv_Il* short *Wcv_Sil* less_short *Wcv_Ir* very_short *Cv_Il* short *Ln_Ir* short *Cv_Vrt* short }

P5→{*Wcv_Wir* very_short ⊗ *Wcv_Wil* very_short}

P6→{*Wcv_Ir* very_short ⊗ *Wcv_Il* very_short},

P7→ {*Wcv_Wir* very_short ⊗ *Wcv_Wil* very_short}

Step 4: Write the description of junction, disjunction lines according to the LWDOS syntax.

JL1→ *junction less_short, junction less_short, junction less_short*

DJL1→ *junction less_short, junction less_short, junction less_short*

Step 5: write the global descriptor of the outline shape

Outline Shape → [[P1 ∪ P2 ∪ P3 ⊕ ↑ JL1 ⊕ P4] ⊕ ↓ DJL1 ⊕ P5 ∪ P6 ∪ P7].

Step 6: Silhouette detailed descriptor is obtained after replacing all parts, junction lines, disjunction lines by their corresponding syntax.

3. Comparison of silhouettes descriptors

A little difference in silhouettes of 3D objects wouldn't be a reason to conclude that those objects are different. The LWDOS silhouette description is very sensitive to any alteration, even if the difference is measured by few pixels. The difference may be due to the acquisition conditions, or to several other phenomenons such as noise in the image.

In order to cope with this problem and permit more tolerance in the comparison process, we propose a method allowing the transformation of LWDOS silhouette descriptors. If descriptors of two silhouettes can be transformed to a same other descriptor, then we conclude that the two silhouettes represent the same object.

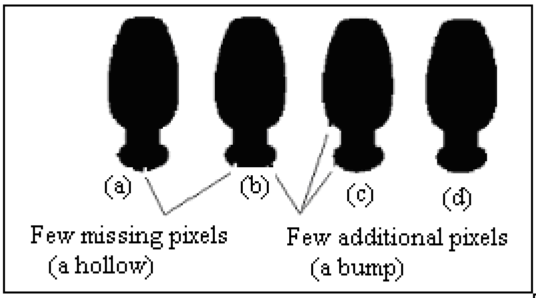


Fig. 9 Noisy silhouettes

Let us consider the four silhouettes of figure 9. The four shapes obviously represent the same object despite little differences due to few additional (bumps outside the silhouette) or missing pixels (hollows in the silhouette). Unfortunately their LWDOS descriptions are very different and their comparison concludes that the represented objects are different. Indeed the LWDOS descriptors of silhouettes (a), (b), (c), and (d) (fig. 9) are respectively:

- [P1⊕↓DJL1⊕P2∪P3]

- [P1⊕↓DJL1⊕ [P2⊕↓DJL2⊕P3∪P4] ∪[P5⊕↓DJL3⊕P6∪P7]]

- [P1⊕↓DJL1⊕P2∪[P3⊕↓DJL2⊕P4∪P5]]

- [P1]

Several techniques of image thresholding and smoothing remove noise from images by elimination of the additional pixels or by the edge closing process (fig. 10).

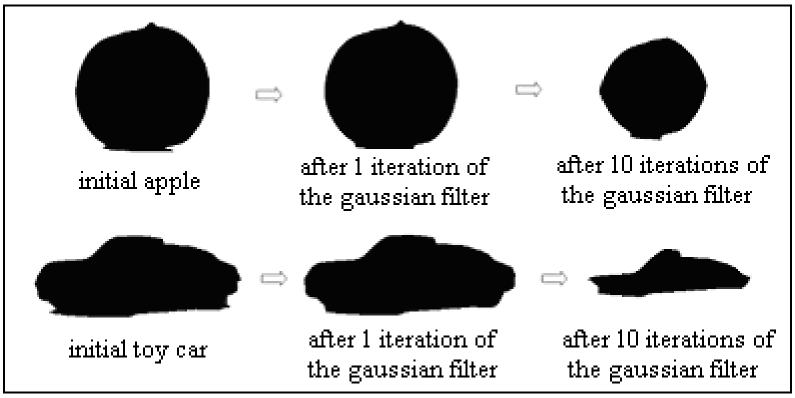


Fig. 10 Results after applying Gaussian filter (σ=2)

Figure 10 shows how an iteration of the smoothing process with the Gaussian filter modifies the outlines of represented shapes (an apple and a toy car) and eliminates some noise on the outlines. Indeed we can see that few elements are eliminated from their contours. Of course it's not correct to apply this filter many times until modifying the object shapes as it's shown in the same figure.

In this paper, we propose to implement the similar process using LWDOS descriptors, by using only textual transformations of silhouettes descriptors. No changes are done on images but only on their LWDOS descriptors. So we want to perform a "textual smoothing".

The feasibility of this approach depends on the deterioration degree of the silhouettes which will not exceed a tolerance threshold for the descriptor transformation otherwise false results may be obtained such as wrongly transforming descriptors of different shapes to the same new descriptor.

The deterioration degree depends on the length of noisy primitives relatively to other primitives (fig 11).

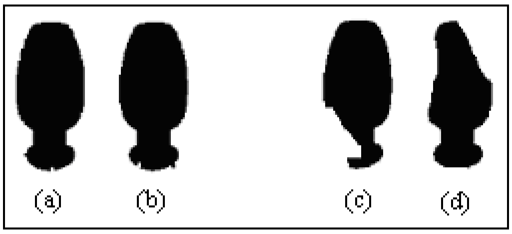


Fig. 11 Deterioration degree

Let's suppose that we can transform both descriptors of shapes (a) and (b) of figure 11 to the same other descriptor, in this case it is correct to conclude that the two shapes look alike because the size of noisy primitives are very little relatively to the silhouettes dimensions.

Let's now consider the two other shapes (c and d of figure 11). Differences of shapes are very important (not few pixels). In this case even we can find the same descriptor for both two shapes, we can not conclude that they are similar. After many experiments, we determined that the size of noisy primitives must not exceed a twentieth of the silhouette dimension (but the software permits to give other values for the threshold).

In this work, we suppose that the distortion does not change the global LWDOS descriptor of silhouettes, otherwise the global descriptor will be corrected at first by applying another method to correct such deterioration [25].

4. Distortion of detailed descriptor

We call a global distortion a noise that modifies the global descriptor of silhouettes.

We call a detailed distortion a noise that doesn't modify the global descriptor of silhouettes but changes only the silhouette detailed descriptor (Fig. 12.c). In this case, the modification can be observed on contours such as distortion of their shapes or relative dimensions.

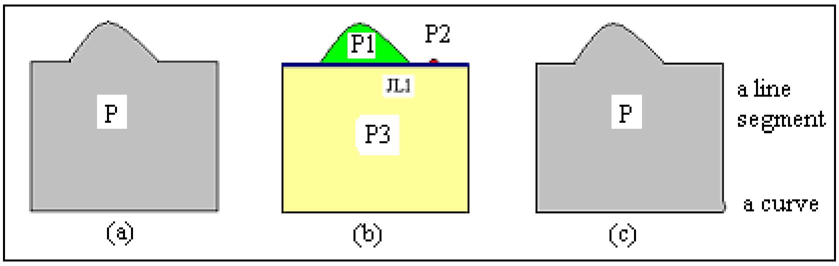


Fig. 12. Distortion of global and detailed descriptors.

Let's suppose that the image of figure 12.a is a noise free silhouette. This synthetic picture is evidently fictional because noise is always present in an image and there is no method capable to eliminate all noise.

The LWDOS global descriptor of the silhouette 12.a is [P] because it contains only one part according to LWDOS language.

Let’s suppose that noise creates a bump (some additional pixels) on the silhouette, therefore there is an apparition of another part (P2) in the descriptor. The LWDOS descriptor becomes [P1 $\cup$ P2 $\oplus \uparrow$ JL1 $\oplus$ P3] (figure 12.b). Such noise witch changes the global descriptor, is corrected by the method in [25].

Global descriptor of the silhouette 12.c is still [P] despite noise on the right boundary. Such distortion is treated in the present paper because it doesn’t modify the global descriptor.

We note that on the right boundary of the silhouette 12.c there is a succession of a straight line segment and a little curve (the curve is supposed due to noise). Our objective consists in recovering the initial descriptor after applying a set of approximation rules. This approach is based only on textual descriptions of silhouettes. After approximation, the obtained descriptors of both silhouettes 12.a and 12.c will be identical according to the LWDOS language and we will conclude that they represent the same object because the difference is only due to noise. Otherwise, if the obtained descriptors are different after approximation, then the compared silhouettes are different and we conclude that they do not describe similar objects.

It is important to precise that our algorithm is recursive, it can be applied several times in order to reach the desired approximation (fig 13).

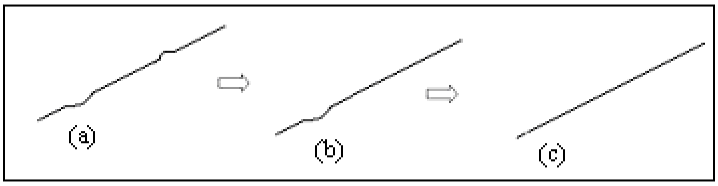


Fig. 13 Applying recursively the approximation process

The figure 13(a) is a distorted silhouettes, it contains two noisy curves. After the first approximation, we remove one curve (fig. 13.b). A second application of the approximation process removes the second curve (fig. 13.c).

5. Modelisation of the Distortion

Noise in an image distorts the contours of the object therefore its LWDOS descriptor will be modified.

Possible distortions on silhouette contours that modify only detailed descriptors are:

1) A curve distorted to a curve and a straight line segment.

2) A line segment distorted to a straight line segment and a (convex or concave) curve.

Our goal consists in applying the approximation rules which will be defined, in order to correct such distortions of silhouettes descriptors. Such rules represent all possible configurations because we took into account all possible orientations and relative lengths according to LWDOS description.

We apply these approximation rules on silhouettes having a same global descriptor. If descriptors are different, the global description will be corrected at first by applying the algorithm developed in [25].

We consider the following notations:

SizeC :the size of a curve primitive, SizeL :the size of straight line segment primitive, and Size=SizeL+SizeC .

InclinC :curve inclination, InclinL : straight line segment inclination, and Inclin : the primitive inclination after the correction of the distortion.

Very short (VS) relatively to the rectangle including the silhouette, Short (Sh), Enough Short (ES), Middle (MI), Enough Long (EL), Long (LO), VeryLong (VL). Entier (EN), Horzr : horizontal oriented on the right, Horzl : horizontal oriented on the left, il: inclined to left (resp. ir inclined to the right).

wil (resp.wir): weakly inclined to left (resp. to the right), gil (resp. gir) : greatly inclined to left (resp. to right). gcv(resp. wcv) : a curve greatly (resp. weakly) convex. gcc(resp. wcc) : a curve greatly (resp. weakly) concave.

5.1 Distortion of a curve into a succession of a straight line segment and a curve

In this case noise distorts a curve to two primitives: a straight line segment and a curve. We try to do the reverse operation and restore the curve from the succession of the initial primitives.

We distinguish two possible types (noted category1 and category2) that concern the approximation by a curve of a succession of a curve and a straight line segment (resp. of a succession of a straight line segment and a curve). The condition in this case is that SizeC is widely greater than SizeL, (SizeC>>SizeL) and 0°<=|InclinC-InclinL|<=threshold_Inclin. After many experiments, Threshold_Inclin is the inclination threshold fixed to 10°, and (SizeC / SizeL) > 20. These are best thresholds but one can give other values, and obtain therefore, other results.

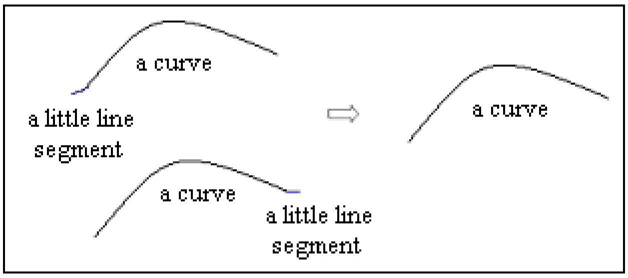


Fig. 14. Examples of approximation by a curve

Figure 14 shows an example of approximation by a curve. We can see that the segments are very little therefore it is possible to suppose that they are due to noise in order to be more tolerant in the comparison process. Figures 15 and 16 are drawn only to better explain the categories of approximation. Noise can not produce such an effect on silhouettes.

Category 1 : (fig. 15)

convex curve InclinC SizeC +Segment InclinL SizeL→ convex curve Inclin Size

concave curve InclinC SizeC +Segment InclinL SizeL→concave curve Inclin Size

Category 2 : (Fig. 16)

Segment InclinL SizeL+ convex curve InclinC SizeC→ convex curve Inclin Size

Segment InclinL SizeL+ concave curve InclinC SizeC→ concave curve Inclin Size

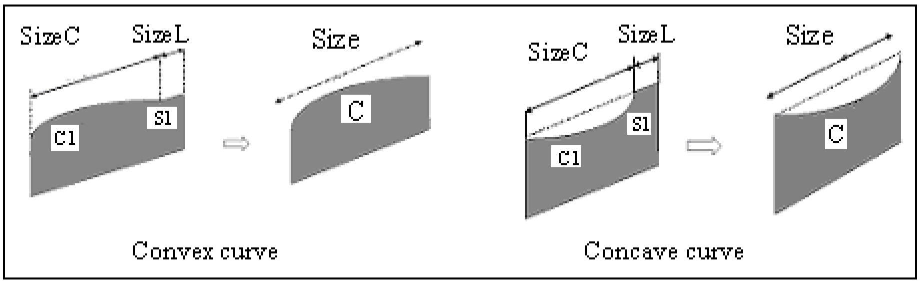


Fig. 15. Approximation by a curve of a succession of a curve and a line segment

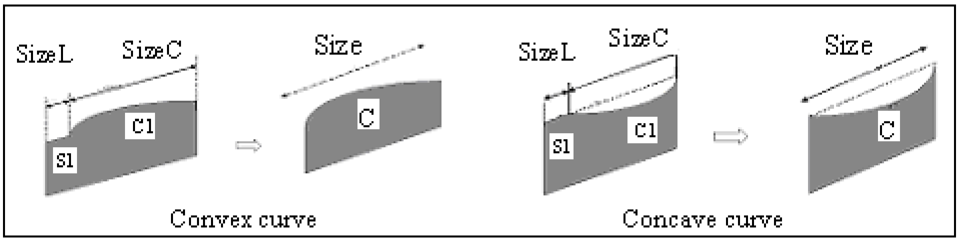


Fig. 16. Approximation by a curve of a succession of a line segment and a curve

5.1.1. Approximation rules: All possible approximation rules are presented in tables 1 and 2

Category 1

Convex curve:	**Concave curve:**
wcv_wir_SizeC+Ln_wir_SizeL→wcv_wir_Size	wcc_wir_SizeC+Ln_wir_SizeL→wcc_wir_Size
wcv_wil_SizeC+Ln_wil_SizeL→wcv_wil_Size	wcc_wil_SizeC+Ln_wil_SizeL→wcc_wil_Size
wcv_ir_SizeC+Ln_ir_SizeL→ wcv_ir_Size	wcc_ir_SizeC+Ln_ir_SizeL→wcc_ir_Size
wcv_il_SizeC+Ln_il_SizeL→wcv_il_Size	wcc_il_SizeC+Ln_il_SizeL→wcc_il_Size
wcv_Gir_SizeC+Ln_Gir_SizeL→wcv_Gir_Size	wcc_Gir_SizeC+Ln Gir_SizeL→wcc_Gir_Size
wcv_Gil_SizeC+Ln_Gil_SizeL→wcv_Gil_Size	wcc_Gil_SizeC+Ln_Gil_SizeL→wcc_Gil_Size
wcv_wir_SizeC+Ln_ir_SizeL→wcv_wir_Size	wcc_wir_SizeC+Ln_ir_SizeL→wcc_wir_Size
wcv_ir _SizeC+Ln_Gir_SizeL→wcv_ir_Size	wcc_ir_SizeC+Ln_Gir_SizeL→wcc_ir_Size
wcv_Gir_SizeC+Ln_il_SizeL→wcv_vrt_Size	wcc_Gir_SizeC+Ln_il_SizeL→wcc_vrt_Size
wcv_Gil_SizeC+Ln_ir_SizeL→wcv_vrt_Size	wcc_Gil_SizeC+Ln_ir_SizeL→wcc_vrt_Size

wcv_wil_SizeC+Ln_il_SizeL→wcv_wil_Size	wcc_wil_SizeC+Ln_il_SizeL→wcc_wil_Size
wcv_il_SizeC+Ln_Gil_SizeL→wcv_il_Size	wcc_il_SizeC+Ln_Gil_SizeL→wcc_il_Size
wcv_vrt_SizeC+Ln_ir_SizeL→wcv_Gir_Size	wcc_vrt_SizeC+Ln_ir_SizeL→wcc_Gir_Size
wcv_vrt_SizeC+Ln_il_SizeL→wcv_Gil_Size	wcc_vrt_SizeC+Ln_il_SizeL→wcc_Gil_Size
wcv_Gir_SizeC+Ln_Gil_SizeL→wcv vrt_Size	wcc_Gir_SizeC+Ln_Gil_SizeL→wcc_vrt_Size
wcv_Gil_SizeC+Ln_Gir_SizeL→wcv_vrt_Size	wcc_Gil_SizeC+Ln_Gir_SizeL→wcc_vrt_Size
wcv_hozr_SizeC +Ln_hozr_SizeL→wcv_hozr_Size	wcc_hozr_SizeC+Ln_hozr_SizeL→wcc_hozr_Size
wcv_hozl_SizeC+Ln_hozl_SizeL→wcv_hozl_Size	wcc_hozl_SizeC+Ln_ hozl_SizeL→wcc_hozl_Size
wcv_vrt _SizeC+Ln_vrt_SizeL→wcv_vrt_Size	wcc_vrt_SizeC+Ln_vrt_SizeL→wcc_vrt_Size
wcv il Sizec+ Ln vrt SizeL → wcv il Size	wcc il Sizec+ Ln vrt SizeL → wcc il Size
wcv il Sizec+ Ln vrt SizeL → wcv wil Size	wcc il Sizec+ Ln vrt SizeL → wcc wil Size
wcv il Sizec+ Ln vrt SizeL → wcv Gil Size	wcc il Sizec+ Ln vrt SizeL → wcc Gil Size
wcv ir Sizec+ Ln vrt SizeL → wcv ir Size	wcc ir Sizec+ Ln vrt SizeL → wcc ir Size
wcv Gil Sizec+ Ln vrt SizeL → wcv Gil Size	wcc Gil Sizec+ Ln vrt SizeL → wcc Gil Size
wcv Gir Sizec+ Ln vrt SizeL → wcv Gir Size	wcc Gir Sizec+ Ln vrt SizeL → wcc Gir Size
wcv wil Sizec+ Ln vrt SizeL → wcv wil Size	wcc wil Sizec+ Ln vrt SizeL → wcc wil Size
wcv wir Sizec+ Ln vrt SizeL → wcv wir Size	wcc wir Sizec+ Ln vrt SizeL → wcc wir Size

Table1. Approximation rules for category 1

Category 2

Convex curve :	**Concave curve:**
Ln_wir_SizeL+wcv_wir_SizeC→wcv_wir_Size	Ln_wir_SizeL+wcc_wir_SizeC→wcc_wir_Size
Ln_wil_SizeL+wcv_wil_SizeC→wcv_wil_Size	Ln_wil_SizeL+wcc_wil_SizeC→wcc_wil_Size
Ln_ir_SizeL+wcv_ir_SizeC→wcv_ir_Size	Ln_ir_SizeL+wcc_ir_SizeC→wcc_ir_Size
Ln_il_SizeL+wcv_il_SizeC→wcv_il_Size	Ln_il_SizeL+wcc_il_SizeC→wcc_il_Size
Ln_Gir _SizeL+wcv_Gir_SizeC→wcv_Gir_Size	Ln_Gir_SizeL+wcc_Gir_SizeC→wcc_Gir_Size
Ln_Gil_SizeL+wcv_Gil_SizeC→wcv_Gil_Size	Ln_Gil_SizeL+wcc_Gil_SizeC→wcc_Gil_Size
Ln_vrt_SizeL+wcv_vrt_SizeC→wcv_vrt_Size	Ln_vrt_SizeL+wcc_vrt_SizeC→wcc_vrt_Size
Ln_ir_SizeL+wcv_wir_SizeC→wcv_wir_Size	Ln_ir_SizeL+wcc_wir_SizeC→wcc_wir_Size
Ln_Gir_SizeL+wcv_ir_SizeC→wcv_ir_Size	Ln_Gir_SizeL+wcc_ir_SizeC→wcc_ir_Size
Ln_il_SizeL+wcv_Gir_SizeC→wcv_vrt_Size	Ln_il_SizeL+wcc_Gir_SizeC→wcc_vrt_Size
Ln_ir_SizeL+wcv_Gil_SizeC→wcv_vrt_Size	Ln_ir_SizeL+wcc_Gil_SizeC→wcc_vrt_Size
Ln_il_SizeL+wcv_wil_SizeC→wcv_wil_Size	Ln_il_SizeL+wcc_wil_SizeC→wcc_wil_Size
Ln_Gil_SizeL+wcv_il_SizeC→wcv_il_Size	Ln_Gil_SizeL+wcc_il_SizeC→wcc_il_Size
Ln_Gil_SizeL+wcv_Gir_SizeC→wcv_vrt_Size	Ln_Gil_SizeL+wcc_Gir_SizeC→wcc_vrt_Size
Ln_Gir_SizeL+wcv_Gil_SizeC→wcv_vrt_Size	Ln_Gir_SizeL+wcc_Gil_SizeC→wcc_vrt_Size
Ln_hozr_SizeL+wcv_hozr_SizeC→wcv_hozr_Size	Ln_hozr_SizeL+wcc_hozr_SizeC→wcc_hozr_Size
Ln_hozl_SizeL+wcv_hozl_SizeC→wcv_hozl_Size	Ln_hozl_SizeL+wcc_hozl_SizeC→wcc_hozl_Size
Ln vrt SizeL+ wcv il Sizec→ wcv il Size	Ln vrt SizeL+ wcc il Sizec→ wcv il Size
Ln vrt SizeL+ wcv il Sizec→ wcv wil Size	Ln vrt SizeL+ wcc il Sizec→ wcv wil Size
Ln vrt SizeL+ wcv il Sizec → wcv Gil Size	Ln vrt SizeL+ wcc il Sizec → wcv Gil Size
Ln vrt SizeL+ wcv ir Sizec → wcv ir Size	Ln vrt SizeL+ wcc ir Sizec → wcv ir Size
Ln vrt SizeL+ wcv Gil Sizec → wcv Gil Size	Ln vrt SizeL+ wcc Gil Sizec → wcv Gil Size
Ln vrt SizeL+ wcv Gir Sizec→ wcv Gir Size	Ln vrt SizeL+ wcc Gir Sizec→ wcv Gir Size
Ln vrt SizeL+ wcv wil Sizec→ wcv wil Size	Ln vrt SizeL+ wcc wil Sizec→ wcv wil Size
Ln vrt SizeL + wcv wir Sizec→ wcv wir Size	Ln vrt SizeL + wcc wir Sizec→ wcv wir Size

Table2. Approximation rules for category 2

5.2 Distortion of a line segment into a succession of a straight segment and a curve

In this case noise distorts a straight line segment into two primitives: a straight line segment and a curve. We try to do the reverse operation and restore the straight line segment from the succession of the initial primitives.

We distinguish two possible types (noted category3 and category4) that concern the approximation by a straight line segment of a succession of a straight line segment and a curve (resp. of a succession of a curve and a straight line segment). The condition in this case is that SizeL is extensively greater than SizeC, (SizeL>>SizeC) and

$0° <= |InclinL-InclinC| <= threshold\_Inclin$. After many experiments, Threshold_Inclin is the inclination threshold fixed to 10°, and (SizeL / SizeC) > 20. These are best thresholds but one can give other values, and obtain therefore, other results.

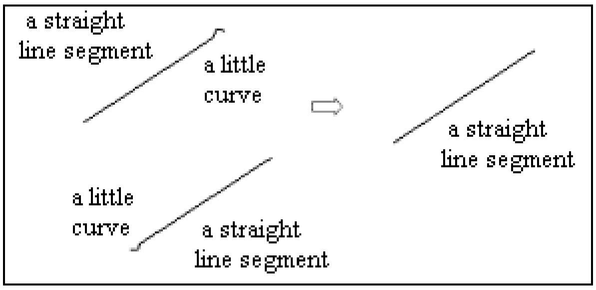


Fig. 17. Distortion of global and detailed descriptors.

Figure 17 shows an example of approximation by a straight line segment. We can see that the curves are very little therefore it is possible to suppose that they are due to noise in order to be more tolerant in the comparison process. Figures 18 and 19 are drawn only to better explain the categories of approximation. Noise can not produce such an effect on silhouettes.

Category 3 (Fig. 18)

Segment InclinL SizeL+ convex curve InclinC SizeC→ Segment Inclin Size

Segment InclinL SizeL+ concave curve InclinC SizeC→ Segment Inclin Size

Category 4 (Fig. 19)

convexe curve InclinC SizeC +Segment InclinL SizeL→ Segment Inclin Size

concave curve InclinC SizeC +Segment InclinL SizeL→ Segment Inclin Size

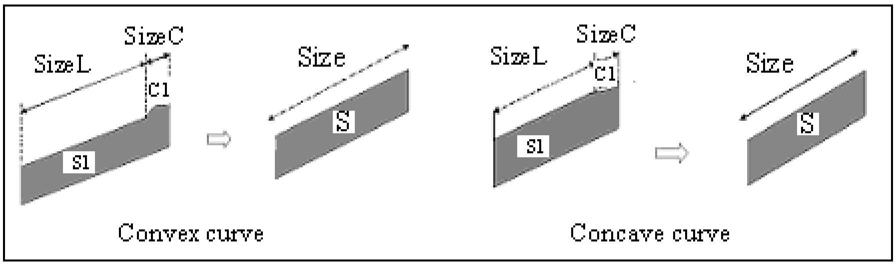


Fig. 18. Approximation by a line segment of a succession of a line segment and a curve

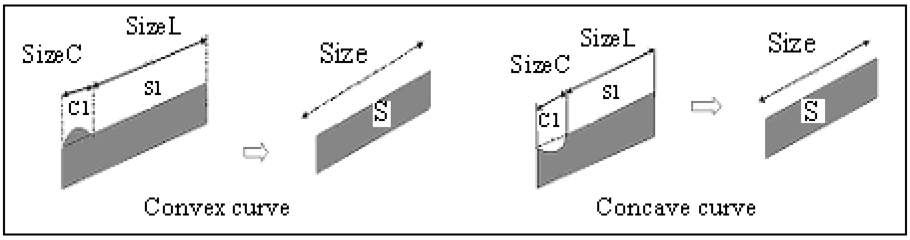


Fig. 19. Approximation by a line segment of a succession of a curve and a line segment

5.2.1. Approximation rules : All possible approximation rules are presented in tables 3 and 4

Category 3

Convex curve :	**Concave curve:**
Ln_wir_SizeC+wcv_wir_SizeL→ln_wir_Size	Ln_wir_SizeC+wcc_wir_SizeL→ln_wir_Size
Ln_wil_SizeC+wcv_wil_SizeL→ln_wil_Size	Ln_wil_SizeC+wcc_wil_SizeL→ln_wil_Size
Ln_ir_SizeC+wcv_ir_SizeL→ln_ir_Size	Ln_ir_SizeC+wcc_ir_SizeL→ln_ir_Size
Ln_il_SizeC+wcv_il_SizeL→ln_il_Size	Ln_il_SizeC+wcc_il_SizeL→ln_il_Size
Ln_Gir_SizeC+wcv_Gir_SizeL→ln_Gir_Size	Ln_Gir_SizeC+wcc_Gir_SizeL→ln_Gir_Size
Ln_Gil_SizeC+wcv_Gil_SizeL→ln_Gil_Size	Ln_Gil_SizeC+wcc_Gil_SizeL→ln_Gil_Size
Ln_wil_SizeC+wcv_wir_SizeL→ln _hozr_Size	Ln_wil_SizeC+wcc_wir_SizeL→ln_hozr_Size
Ln_wir_SizeC+wcv_wil_SizeL→ln_hozl_Size	Ln_wir_SizeC+wcc_wil_SizeL→ln_hozl _Size
Ln_Gir_SizeC+wcv_Gil_SizeL→ln_vrt_Size	Ln_Gir_SizeC+wcc_Gil _SizeL→ln_vrt_Size
Ln_Gil_SizeC+wcv_Gir_SizeL→ln_vrt_Size	

Ln_hozr_SizeL+wcv_hozr_SizeC→ln_hozr_Size	Ln_Gil_SizeC+ wcc_Gir_SizeL→ln_vrt_Size
Ln_hozl_SizeL+wcv_hozl_SizeC→ln_hozl_Size	Ln_hozr _SizeL+wcc_hozr_SizeC→ln_hozr_Size
Ln_vrt_SizeL+wcv_vrt_SizeC→ln_vrt_Size	Ln_hozl_SizeL+wcc_hozl_SizeC→ln_hozl_Size
Ln vrt SizeL+ wcv il Sizec→ Ln il Size	Ln_vrt_SizeL+wcc_vrt_SizeC→ln_vrt_Size
Ln vrt SizeL+ wcv il Sizec→ Ln wil Size	Ln vrt SizeL+ wcc il Sizec→ Ln il Size
Ln vrt SizeL+ wcv il Sizec → Ln Gil Size	Ln vrt SizeL+ wcc il Sizec→ Ln wil Size
Ln vrt SizeL+ wcv ir Sizec → Ln ir Size	Ln vrt SizeL+ wcc il Sizec → Ln Gil Size
Ln vrt SizeL+ wcv Gil Sizec → Ln Gil Size	Ln vrt SizeL+ wcc ir Sizec → Ln ir Size
Ln vrt SizeL+ wcv Gir Sizec→ Ln Gir Size	Ln vrt SizeL+ wcc Gil Sizec → Ln Gil Size
Ln vrt SizeL+ wcv wil Sizec→ Ln wil Size	Ln vrt SizeL+ wcc Gir Sizec→ Ln Gir Size
Ln vrt SizeL + wcv wir Sizec→ Ln wir Size	Ln vrt SizeL+ wcc wil Sizec→ Ln wil Size
	Ln vrt SizeL + wcc wir Sizec→ Ln wir Size

Table3. Approximation rules for category 3

Category 4

Convex curve :	**Concave curve:**
wcv_wir_SizeC+Ln_wir_SizeL→ln_wir_Size	wcc_wir_SizeC+Ln_wir_SizeL→ln_wir_Size
wcv_wil_SizeC+Ln_wil_SizeL→ln_wil_Size	wcc_wil_SizeC+Ln_wil_SizeL→ln_wil_Size
wcv_ir_SizeC+Ln_ir_SizeL→ln_ir_Size	wcc_ir_SizeC+Ln_ir_SizeL→ln_ir_Size
wcv_il_SizeC+Ln_il_SizeL→ln_il_Size	wcc_il_SizeC+Ln_il_SizeL→ln_il_Size
wcv_Gir_SizeC+Ln_Gir_SizeL→ln_Gir_Size	wcc_Gir_SizeC+Ln_Gir_SizeL→ln_Gir_Size
wcv_Gil_SizeC+Ln_Gil_SizeL→ln_Gil_Size	wcc_Gil_SizeC+Ln_Gil_SizeL→ln_Gil_Size
wcv_vrt_SizeC+Ln_vrt_SizeL→ln_vrt_Size	wcc_vrt_SizeC+Ln_vrt_SizeL→ln_vrt_Size
wcv_wil_SizeC+Ln_wir_SizeL→ln_hozr_Size	wcc_wil_SizeC+ Ln_wir_SizeL→ln_hozr_Size
wcv_wir_SizeC+Ln_wil_SizeL→ln_hozl _Size	wcc_wir_SizeC+Ln_wil_SizeL→ln_hozl_Size
wcv_Gir_SizeC+Ln_Gil_SizeL→ln_vrt_Size	wcc_Gir_SizeC+Ln_Gil_SizeL→ln_vrt_Size
wcv_Gil_SizeC+Ln_Gir_SizeL→ln_vrt _Size	wcc_Gil_SizeC+Ln_Gir_SizeL→ln_vrt_Size
wcv_hozr_SizeC+Ln_hozr_SizeL→ln_hozr_Size	wcc_hozr_SizeC+Ln_hozr_SizeL→ln_hozr_Size
wcv_hozl_SizeC+Ln_hozl_SizeL→ln_hozl_Size	wcc_hozl_SizeC+Ln_hozl_SizeL→ln_hozl_Size
wcv il Sizec+ Ln vrt SizeL → Ln il Size	wcc il Sizec+ Ln vrt SizeL → Ln il Size
wcv il Sizec+ Ln vrt SizeL → Ln wil Size	wcc il Sizec+ Ln vrt SizeL → Ln wil Size
wcv il Sizec+ Ln vrt SizeL → Ln Gil Size	wcc il Sizec+ Ln vrt SizeL → Ln Gil Size
wcv ir Sizec+ Ln vrt SizeL → Ln ir Size	wcc ir Sizec+ Ln vrt SizeL → Ln ir Size
wcv Gil Sizec+ Ln vrt SizeL → Ln Gil Size	wcc Gil Sizec+ Ln vrt SizeL → Ln Gil Size
wcv Gir Sizec+ Ln vrt SizeL → Ln Gir Size	wcc Gir Sizec+ Ln vrt SizeL → Ln Gir Size
wcv wil Sizec+ Ln vrt SizeL → Ln wil Size	wcc wil Sizec+ Ln vrt SizeL → Ln wil Size
wcv wir Sizec+ Ln vrt SizeL → Ln wir Size	wcc wir Sizec+ Ln vrt SizeL → Ln wir Size

Table4. Approximation rules for category 4

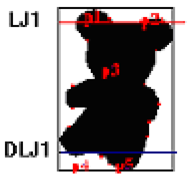
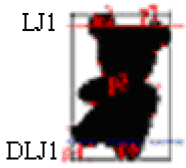
6. Experimentation

We applied our proposed algorithms on silhouettes of real 3D objects. The left and right images are taken from adjacent points of view such as stereoscopic images or images of nearby moving objects. The silhouettes are obtained after image thresholding and binarization. We determine global and detailed descriptors for each couple of silhouettes.

After correction of global descriptors and elimination of additional parts due to noise [25], we apply the approximating rules in order to correct detailed descriptors.

The determination of detailed descriptors requires the localization of inflection points on the silhouette, the number of these points is proportional to the precision of the description (fig. 6). For the localization of these points we used the algorithm of Chetverikov [24]. Three examples of experimentation results are presented below for witch detailed descriptors become similar after applying reduction rules.

The underlined descriptors below are those concerned by the approximation rules.

a couple of silhouettes		
Global Descriptor	[p1Up2⊕↑LJ1⊕p3⊕↓DLJ1⊕p4Up5]	[p1Up2⊕↑LJ1⊕p3⊕↓DLJ1⊕p4Up5]
Junction Lignes	Junction Middle, Free Low Very Short, Junction Short	Junction Middle, Free Low Very Short, Junction Short
Disjunction Lignes	Junction Short, Free Heigh Very Short, Junction Middle	Junction Short, Free Heigh Very Short, Junction Middle
Detailed Descriptor before approximation	[{wcv wil SH, Ln Vrt VS ⊗ wcv ir SH}U {Ln Hoz VS, wcv il VS ⊗ Ln Hoz VS, wcv wir VS} ⊕↑LJ1⊕ {wcv ir SH, wcv Gir VS, Ln Vrt VS, wcv ir VS, wcv wil SH, wcv ir SH, Ln Hoz VS, wcv il ES, wcv wir VS ⊗ wcv ir VS, wcv il VS, wcv Gil SH, wcv ir VS, Ln Vrt SH, wcv il VS, Ln Vrt VS, wcv Gil ES} ⊕↓DLJ1⊕ {Ln ir VS ⊗ wcv wil VS} U {wcv ir VS, wcv ir VS, Ln Hoz SH ⊗ wcv Gil SH, Ln Hoz SH}]	[{wcv wil SH,Ln Vrt VS ⊗ wcv ir SH}U {Ln Hoz VS, wcv il VS ⊗ Ln Hoz VS, wcv wir VS}⊕↑LJ1⊕ {wcv ir SH, wcv Gir VS, Ln Vrt VS, wcv ir VS, wcv wil SH, **Ln Vrt VS, wcv ir SH**, Ln Hoz VS, **wcv il SH, Ln Vrt VS** , wcv wir VS ⊗ wcv ir VS, wcv il VS, wcv Gil SH, wcv ir VS, Ln Gil SH, wcv Gil ES}⊕↓DLJ1⊕ {Ln ir VS ⊗ wcv wil VS}U { wcv ir VS, wcv ir VS, Ln Hoz SH ⊗ wcv Gil

		SH, Ln Hoz SH}]
Detailed Descriptor after approximation	[{wcv wil SH, Ln Vrt VS ⊗ wcv ir SH}U {Ln Hoz VS, wcv il VS ⊗ Ln Hoz VS, wcv wir VS} ⊕↑LJ1⊕ {wcv ir SH, wcv Gir VS, Ln Vrt VS, wcv ir VS, wcv wil SH, wcv ir SH, Ln Hoz VS, wcv il ES, wcv wir VS ⊗ wcv ir VS, wcv il VS, wcv Gil SH, wcv ir VS, Ln Gil SH , wcv Gil ES} ⊕↓DLJ1⊕ {Ln ir VS ⊗ wcv wil VS} U {wcv ir VS, wcv ir VS, Ln Hoz SH ⊗ wcv Gil SH, Ln Hoz SH}]	[{wcv wil SH, Ln Vrt VS ⊗ wcv ir SH}U {Ln Hoz VS, wcv il VS ⊗ Ln Hoz VS, wcv wir VS} ⊕↑LJ1⊕ {wcv ir SH, wcv Gir VS, Ln Vrt VS, wcv ir VS, wcv wil SH, wcv ir SH, Ln Hoz VS, wcv il ES, wcv wir VS ⊗ wcv ir VS, wcv il VS, wcv Gil SH, wcv ir VS, Ln Gil SH , wcv Gil ES} ⊕↓DLJ1⊕ {Ln ir VS ⊗ wcv wil VS} U {wcv ir VS, wcv ir VS, Ln Hoz SH ⊗ wcv Gil SH, Ln Hoz SH}]
Used rules	Ln vrt sizel + wcv il sizec→wcv Gil size Ln vrt sizel + wcv Gil sizec→ wcv Gil size	Ln vrt sizel + wcv ir sizec→wcv ir sizec wcv il sizec + Ln vrt sizel→wcv il size

Fig. 20: silhouettes of a toy bears

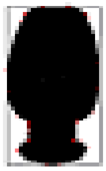
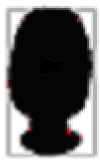
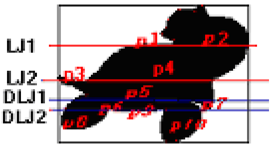
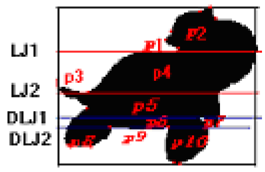
a couple of silhouettes		
Global Descriptor	[P1]	[P1]
Detailed Descriptor before approximation	[{Ln Hoz SH, wcv wil VS, Ln Vrt VS, wcv Gil ES, Ln Vrt VS**, wcv ir ES, Ln ir VS, Ln Vrt VS**, Ln Vrt VS, wcv il VS, wcv ir VS, Ln Hoz ES ⊗ Ln Hoz SH, wcv wir VS, Ln Vrt VS, wcv Gir ES, Ln Vrt VS, wcv Gil ES, Ln il VS, Ln Vrt VS, wcv ir VS, wcv il SH, Ln Hoz ES}]	[{Ln Hoz SH, wcv wil VS, Ln Vrt VS, wcv Gil ES, Ln Vrt VS, **wcv ir SH, Ln ir VS**, Ln Vrt VS,wcv il VS, wcv ir VS, Ln Hoz ES ⊗ Ln Hoz SH, wcv wir VS, Ln Vrt VS, wcv Gir ES, Ln Vrt VS, wcv Gil ES, Ln il VS, Ln Vrt VS, wcv ir VS, wcv il SH, Ln Hoz ES}]
Detailed Descriptor after approximation	[{Ln Hoz SH, wcv wil VS, Ln Vrt VS, wcv Gil ES, Ln Vrt VS, wcv ir ES, Ln Vrt VS,wcv il VS, wcv ir VS, Ln Hoz ES ⊗ Ln Hoz SH, wcv wir VS, Ln Vrt VS, wcv Gir ES, Ln Vrt VS, wcv Gil ES, Ln il VS, Ln Vrt VS, wcv ir VS, wcv il SH, Ln Hoz ES}]	[{Ln Hoz SH, wcv wil VS, Ln Vrt VS, wcv Gil ES, Ln Vrt VS, wcv ir ES, Ln Vrt VS,wcv il VS, wcv ir VS, Ln Hoz ES ⊗ Ln Hoz SH, wcv wir VS, Ln Vrt VS, wcv Gir ES, Ln Vrt VS, wcv Gil ES, Ln il VS, Ln Vrt VS, wcv ir VS, wcv il SH, Ln Hoz ES}]
Used rules	wcv ir sizec + Ln ir sizel→wcv ir size wcv ir sizec + Ln vrt sizel→wcv ir size	wcv ir sizec + Ln ir sizel→wcv ir size

Fig. 21: silhouettes of cups

a couple of silhouettes		
Global Descriptor	[p1Up2⊕↑LJ1⊕p3Up4⊕↑LJ2⊕p5⊕↓DLJ1 ⊕[p6⊕↓DLJ2⊕p8Up9Up10]Up7]	[p1Up2⊕↑LJ1⊕p3Up4⊕↑LJ2⊕p5⊕↓DLJ1⊕[p6 ⊕↓DLJ2⊕p8Up9Up10]Up7]
Junction Lignes	1) Junction Very Short, Junction Enough Short 2) Junction Very Short, Free Low Very Short, Junction Enough Long	1) Junction Very Short, Junction Enough Short 2) Junction Very Short, Free Low Very Short, Junction Enough Long
Disjunction Lignes	1) Junction Enough Long, Junction Very Short 2) Junction Enough Short, Junction Enough Short, Junction Enough Short	1) Junction Enough Long, Junction Very Short 2) Junction Enough Short, Junction Enough Short, Junction Enough Short
Detailed Descriptor before approximation	[{Ln Hoz SH, wcv wil VS ⊗ Ln Hoz SH, wcv, wir VS}U {Ln Hoz VS, Ln il VS, Ln Hoz VS, **wcv il SH, Ln il VS**, Ln Vrt VS, Ln wir VS, **Ln Vrt SH, wcc Vrt VS** ⊗ Ln Hoz VS, Ln ir VS, Ln Hoz VS, wcv ir SH, Ln Vrt VS} ⊕↑LJ1 ⊕ {Ln Hoz VS, wcv Gir VS ⊗ Ln Hoz VS, wcv wir VS} U {Ln il VS, wcv il SH, Ln il SH, Ln Vrt VS ⊗ wcv il SH, wcc il VS} ⊕↑LJ2⊕ {Ln ir VS, Ln Hoz VS, wcv wir VS, Ln il VS ⊗ Ln Vrt VS} ⊕↓ DLJ1⊕[{Ln il VS ⊗ wcv ir VS} ⊕↓ DLJ2 ⊕ {Ln Gil VS, Ln Vrt VS, Ln ir VS, Ln Hoz SH ⊗ wcv il SH, Ln Hoz SH}U {wcv wir VS, Ln Hoz SH ⊗ wcv wil VS, Ln Hoz SH}U {Ln Vrt SH, Ln ir VS, Ln Hoz VS ⊗ Ln Vrt VS, wcv il SH, Ln Hoz VS}]U {Ln ir VS ⊗ wcv Gil VS}]	[{Ln Hoz SH, wcv wil VS ⊗ Ln Hoz SH, wcv, wir VS} U {Ln Hoz VS, Ln il VS, Ln Hoz VS, **wcv il ES, Ln Vrt VS**, Ln wir VS, Ln Vrt ES ⊗ Ln Hoz VS, Ln ir VS, , Ln Hoz VS, wcv ir SH, Ln Vrt VS } ⊕↑LJ1 ⊕ {Ln Hoz VS, wcv Gir VS ⊗ Ln Hoz VS, wcv wir VS} U {Ln il VS, wcv il SH, Ln il SH, Ln Vrt VS ⊗ wcv il SH, wcc il VS} ⊕↑LJ2⊕ {Ln ir VS, Ln Hoz VS, wcv wir VS, Ln il VS ⊗ Ln Vrt VS} ⊕↓ DLJ1⊕[{Ln il VS ⊗ wcv ir VS} ⊕↓ DLJ2 ⊕ {Ln Gil VS, Ln Vrt VS, Ln ir VS, Ln Hoz SH ⊗ wcv il SH, Ln Hoz SH}U {wcv wir VS, Ln Hoz SH ⊗ wcv wil VS, Ln Hoz SH}U {Ln Vrt SH, Ln ir VS, Ln Hoz VS ⊗ Ln Vrt VS, wcv il SH, Ln Hoz VS}]U {Ln ir VS ⊗ wcv Gil VS}]
Detailed Descriptor after approximation	[{Ln Hoz SH, wcv wil VS ⊗ Ln Hoz SH, wcv, wir VS}U {Ln Hoz VS, Ln il VS, Ln	[{Ln Hoz SH, wcv wil VS ⊗ Ln Hoz SH, wcv, wir VS}U {Ln Hoz VS, Ln il VS, Ln Hoz VS,

	Hoz VS, **wcv il ES,** Ln wir VS, Ln Vrt ES ⊗ Ln Hoz VS, Ln ir VS, Ln Hoz VS, wcv ir SH, Ln Vrt VS} ⊕↑LJ1 ⊕ {Ln Hoz VS, wcv Gir VS ⊗ Ln Hoz VS, wcv wir VS} U {Ln il VS, wcv il SH, Ln il SH, Ln Vrt VS ⊗ wcv il SH, wcc il VS} ⊕↑LJ2⊕ {Ln ir VS, Ln Hoz VS, wcv wir VS, Ln il VS ⊗ Ln Vrt VS} ⊕↓ DLJ1⊕[{Ln il VS ⊗ wcv ir VS} ⊕↓ DLJ2 ⊕ {Ln Gil VS, Ln Vrt VS, Ln ir VS, Ln Hoz SH ⊗ wcv il SH, Ln Hoz SH}U {wcv wir VS, Ln Hoz SH ⊗ wcv wil VS, Ln Hoz SH}U {Ln Vrt SH, Ln ir VS, Ln Hoz VS ⊗ Ln Vrt VS, wcv il SH, Ln Hoz VS}]U {Ln ir VS ⊗ wcv Gil VS}]	**wcv il ES**, Ln wir VS, Ln Vrt ES ⊗ Ln Hoz VS, Ln ir VS, Ln Hoz VS, wcv ir SH, Ln Vrt VS} ⊕↑LJ1 ⊕ {Ln Hoz VS, wcv Gir VS ⊗ Ln Hoz VS, wcv wir VS} U {Ln il VS, wcv il SH, Ln il SH, Ln Vrt VS ⊗ wcv il SH, wcc il VS} ⊕↑LJ2⊕ {Ln ir VS, Ln Hoz VS, wcv wir VS, Ln il VS ⊗ Ln Vrt VS} ⊕↓ DLJ1⊕[{Ln il VS ⊗ wcv ir VS} ⊕↓ DLJ2 ⊕ {Ln Gil VS, Ln Vrt VS, Ln ir VS, Ln Hoz SH ⊗ wcv il SH, Ln Hoz SH}U {wcv wir VS, Ln Hoz SH ⊗ wcv wil VS, Ln Hoz SH}U {Ln Vrt SH, Ln ir VS, Ln Hoz VS ⊗ Ln Vrt VS, wcv il SH, Ln Hoz VS}]U {Ln ir VS ⊗ wcv Gil VS}]
Used rules	wcv il sizec + Ln il sizel→wcv il size Ln vrt sizel + wcv vrt sizec→ Ln vrt size	wcv il sizec + Ln vrt sizel→wcv il size

Fig. 22: silhouettes of a toy turtles

7. Conclusion

We presented in this paper a new method for silhouettes comparison. The silhouettes are described according to the LWDOS language. In order to cope with little differences in the silhouettes shapes that unfortunately yield very different descriptors, we have developed a set of approximation rules in order to approximate the noisy silhouette description. To validate the proposed approach, the algorithms have been applied for pairs of silhouettes extracted from real images.

Once the global description is corrected (not the aim of this paper), we apply the approximating rules to reduce noise that doesn't produce additional parts or lines in silhouettes but modifies only the detailed descriptor.

Tests achieved on real images show that it is possible to correct the textual description of silhouettes in order to be able to compare them and then to say if they represent the same object. This technique can be used in matching and image recognition processes.

8. Appendix

The grammar G $_{LWDOS}$ of LWDOS language is given by the 4-tuple GLWDOS=(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, and P is a finite set of production rules of the type $\alpha \rightarrow$ ß, where $\alpha \in V_N$ and ß$\in$ (V_N U V_T)* of all string. The non-terminal vocabulary of LWDOS language is written as [1]:

V_N= {S_0, <Composed_part>, <United_part>, <Part>, <Left_Boundary>, <Right_Boundary>, <Contour_descriptor>, <Contour_geometry>, <Relative_length>, <Junction_line>, <Disjunction_line>, <State>}

The terminal vocabulary of LWDOS language is written as: .

V_T={ ⊕, ⊗ , [,] , { , } , ↑ , ↓ , U, *Junction*, *Free_High*, *Free_Low*, *very_short*, *short*, *less_short*, *middle*, *enough_long*, *long*, *very_long*, *entire*} U Set_attr_feat Where Set_attr_feat = {*Ln_Vrt*, *Wcv_Vrt*, *Cv_Vrt*, *Scv_Vrt*, *Wcc_Vrt*, *Cc_Vrt*, *Scc_Vrt*, *Ln_Hor*,*Wcv_Hor*, *Cv_Hor*, *Scv_Hor*, *Wcc_Hor*, *Cc_Hor*, *Scc_Hor*, *Ln_Wil*, *Wcv_Wil*, *Cv_Wil*, *Scv_Wil*,*Wcc_Wil*, *Cc_Wil*, *Scc_Wil*, *Ln_Il*, *Wcv_Il*, *Cv_Il*, *Scv_Il*, *Wcc_Il*, *Cc_Il*, *Scc_Il*, *Ln_Sil*, *Wcv_Sil*,*Cv_Sil*, *Scv_Sil*, *Wcc_Sil*, *Cc_Sil*, *Scc_Sil*, *Ln_Wir*, *Wcv_Wir*, *Cv_Wir*, *Scv_Wir*, *Wcc_Wir*,*Cc_Wir*, *Scc_Wir*, *Ln_Ir*, *Wcv_Ir*, *Cv_Ir*, *Scv_Ir*, *Wcc_Ir*, *Cc_Ir*, *Scc_Ir*, *Ln_Sir*, *Wcv_Sir*, *Cv_Sir*,*Scv_Sir*, *Wcc_Sir*, *Cc_Sir*, *Scc_Sir*}

The LWDOS production rules P is written as follows:

S0 $\rightarrow$ <Composed_part >

<Composed_part> $\rightarrow$ [<United_part>⊕↑<Junction_line>⊕<Part>] /

[<Part>⊕↓<Disjunction_line>⊕<United_part>]

<United_part> → <Part> U <United-part> / <Part>

<Part>→{<Left_Boundary>⊗<Right_Boundary>}/<Composed_part>/ <United_part>

<Left_Boundary>→ <Contour_descriptor>< Left_Boundary> / <contour_descriptor>

<Right_Boundary>→<Contour_descriptor><Right_Boundary>/<Contour_descriptor >

<Contour_descriptor > → <Contour_geometry> < Relative_length>

<Contour_geometry> → *Ln_Vrt/ Wcv_Vrt/ ..../ Wcc_Sir/ Cc_Sir/ Scc_Sir*

<Junction_line> → <State><Relative_length>/

<State><Relative_length><Junction_line>

<Disjunction_line>→ <State><Relative_length>/<State><Relative_length>

<Disjunction_line>

<State> → *Junction / Free_High / Free_Low*

< Relative_length>→ *very_short /short /less_short/ middle/ enough_long/ long*

very_long / entire

9. References

[1] S.Larabi, S.Bouagar, F.Trespaderne, and E.Lopez, “LWDOS language for writing descriptors of outline shapes”, In the 13th Scandinavian Conference on Image Analysis, Goteburg Sweden, june, 2003 .

[2] D. L.Borges, and R. B.Fisher, “Class-Based Recognition of 3D Objects Represented by Volumetric Primitives”. In R. Fisher E. Trucco (eds.), British Machine Vision Conference 1: 303-312, 1996.

[3] G. G.Medioni, and A. R. J.Franois, “3-D structures forgeneric object recognition”. Invited conference paper, In the proc. of the Int. Conference on Pattern Recognition, 1: 30-37, Barcelona, Spain, September, 2000.

[4] Y.Sumi, Y.Kawai, T.Yoshimi, and F.Tomita, “3D object recognition in cluttered environments by segment based stereo vision”. Int. Journal of Computer Vision, 46(1), 2002.

[5] H.Borotschnig, L.Paletta, M.Prantl, and A.Pinz, "Appearance based active object recognition". Image and Vision Computing, 18: 715-727, 2000.

[6] R. J.Campbell, and P. J.Flynn, "Eigenshapes for 3D object recognition in range data". In the proc. of IEEE Conf. Computer Vision and Pattern Recognition, Colorado, June 1999.

[7] Y.Rui, and T.S.Huang, "Image retrieval, "current technologies, promising directions, and open issues". Journal of Visual Communications and Image Representation, 10: 39-96, 1999

[8] T.Bandlow, A.Hauck, T.Einsele, and G.Farber, "Recognising objects by their silhouette". Computer Engineering in Systems Applications , 1998.

[9] B. J.Super, "Fast retrieval of isolated visual shapes". Computer Vision and Image Understanding, 85: 1-21,2002.

[10] A. Sethi, D. Renaudie, D. Kriegman and J. Ponce, "Curve and surface duals and the recognition of curved 3D objects from their silhouettes". Int. Journal of computer vision 58 (1), 73-86, 2004

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

[12] J.L.Chen, and G. C.Stockman, "3D free form object recognition using indexing by contour features". Computer Vision and Image Understanding, 71: 334- 335, 1998.

[13] D.Hong, T.Sarkodie-Gyan, A.W.Campbell, andY.Yan, "A prototype indexing approach to 2-D object description and recognition". Pattern Recognition, 31(6): 699-725, june, 1998.

[14] R.Meathrel, and A.Galton, "Ahierarchy of *boundarybased shape descriptors".* In the proc. of the seventeenth Int. Joint Conference on Artificial Intelligence, 2: 1359-1364, Seattle, Washington, August, 2001.

[15] T.Sebastian, P.Klein, and B.Kimia, "Recognition of Shapes by Editing Their Shock Graphs". IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(5): 550-571, may, 2004.

[16] W. G.Hayward, "Effects of outline shape in object recognition". Journal of experimental psychology: human perception and Performance, 24: 427-440, 1998.

[17] A.Quddus, F.Alaya Cheikh, and M. Gabbouj, "Content based Object Retrieval Using Maximun Curvature Points In ContourImages". Proc. of Symposium, On Storage and Retrieval for Media Databases 2000, San Jose, California, USA, January, 2000.

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

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

[20] A.Malaviya, and L.Peters, "Fuzzy handwriting description language: FOHDEL". Pattern Recognition, 33(1): 119-131, January 2000.

[21] C.Alvarado, M.Oltmans, and R.Davis, "A Framework for Multi-Domain Sketch Recognition". In the proc. Of AAAI Spring Symposium on Sketch Understanding, 2002

[22] R.Davis, "Position statement and overview, sketch recognition at MIT". In the proc. of AAAI Spring Symposium on Sketch Understanding, 2002.

[23] R. L. Graham, "An efficient algorithm for determining the convex hull of a finite planar set". Information Processing Letters, 132-133, 1972.

[24] D. Chetverikov: « A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves », 10th Int. Conference, Groningen, the Netherlands, August 2003

[25] S.Aouat, S.Larabi « Comparaison of silhouettes descriptions, Int. Conference on Graphic vision and image processing, Cairo, Egypt, December 2005.