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

# Coarse Comparison of Silhouettes using XLWDOS Language

**Saliha AOUAT, Slimane LARABI**

*Computer Science Department, Faculty of Electronic and Computer Science*
*University of Sciences and Technologies HOUARI BOUMEDIENNE*

*Shape query using images database is an important area in computer vision researches. We propose in this paper a method for the matching between a shape query and a shape model. Assuming that shape models are stored in the database using their textual descriptors XLWDOS [14, 15], we propose in this paper a first step of the shape retrieval which consists to coarse matching of shapes. This step identical to shape indexing must be followed by full shapes matching and similarity measure.*

*The proposed coarse matching of shapes takes into account deformation of outline shapes due to noise or to varying camera position. We demonstrate that two slightly different descriptors have the same index. Experiments over real images are conducted and explained.*

***Keywords:*** *Shape retrieval, Shape indexing, Coarse matching, Textual descriptor, noise*

## 1. INTRODUCTION

Various methods have been developed in order to represent shapes in an abstract and efficient way. The most interesting methods are the part-based methods where a silhouette is decomposed into parts [11, 22, 25, 28, 32, 37], the aspect-graph methods that are viewer-centered representations of a three-dimensional object [5, 12], methods that use the medial axis of silhouettes [8, 29, 39] and appearance-based methods [23].

The recognition of object using database of shapes is a current and difficult problem. Different methods based on different representations of shapes have been proposed. All of these methods proceed on descriptors comparison and a similarity measure is then computed.

Indexing database of images facilitates the retrieval problem of any image. Depending on the shape descriptors, many solutions have been proposed:

- A global statistical description of image features has been widely used in image analysis for image description, indexing, and retrieval. Such global feature descriptors include the image's color histogram [34], edge statistical information [7, 30], and wavelet [26]. Statistical description of pixel intensities over space quantifies the statistical similarity between an image and a model, the likelihood of the collection of feature vectors extracted from the image is computed based on the stochastic process derived from the image categories [19, 16, 6]
- Primitive features such as edges, corners, blobs, model specific types of distributions and constitute the building blocks of image content [20, 21, 38, 33]
- Semantic categorization allows the dominant regions to be extracted using multi level color segmentation [35, 36, 31], texture [27], relation between regions [1] and Markov model classification [18, 3].
- The hashing paradigm [24, 13] was originally developed for object recognition problems. It introduces an indexing approach based on transformation invariant representations and is especially geared toward efficient recognition of partial structures in rigid objects belonging to large databases.
- Geometric methods [10] that are adapted to assumptions made about the illumination and viewpoint variations existing between two images. They determine the optimal matching of two patch-based image representation under rotation, scaling, and translations. Typically objects are subject to quite complex transformations when projected into an image plane. Therefore, the real transformations are approximated by easier transformations that can be mathematically treated with reasonable effort [9, 2]

In order to retrieve a shape in images database, the comparison between shapes may be done in two stages: coarse matching and full matching allowing similarity measure.

In this work, we focus us only on the first step. We propose a method for coarse matching of silhouettes

correspondence author: *E-mail: saouat,slarabi@usthb.dz*

represented by their textual descriptors [14, 15]. We show at first how to index shapes, and next how to associate the same index for two shapes whose outlines are few different.

We give in section 2 an overview on the representation of outline shapes with a textual descriptor. In section 3, we present our method for coarse comparison of outline shapes based on their textual descriptors. We show how to index a database of shapes models, and how to associate to a shape query a set of shapes models as possible matches. We propose then our algorithm for the coarse matching of shapes. In section 4 we present and discuss the obtained experimental results.

## 2. TEXTUAL DESCRIPTION OF OUTLINE SHAPES

The first step of shape description is its decomposition into parts and separating lines. For this, we sweep the outline shape from the top to bottom, and we locate concave points for which the outer contour changes the direction top-bottom-top or bottom-top-bottom (see fig. 1).

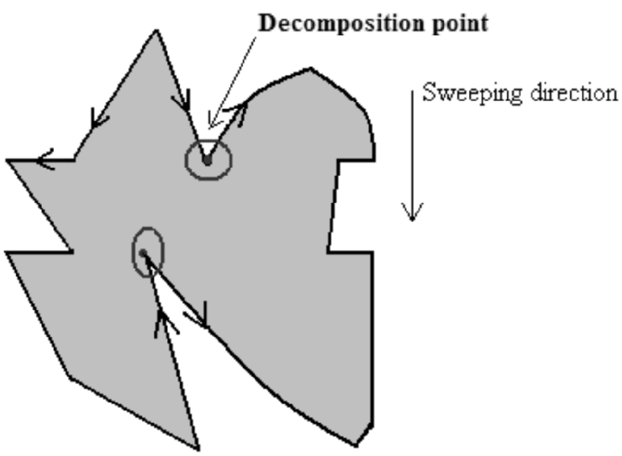



**Figure 1: Location of decomposition points**

The outline shape is decomposed at these points into parts, and separating lines: either, two parts or more are joined with a third part through a junction line, or a part is joined with two parts or more through a disjunction line. This process applied for example to silhouette of figure 1 produces five parts, one junction and one disjunction line (see fig. 2). The part and separating lines are numbered from the top to bottom and left to right.

The second step is the representation of these elements with a graph structure where nodes are the different parts and separating lines.

The third step is the description of each element in order to guaranty the uniqueness of the outline shape representation.

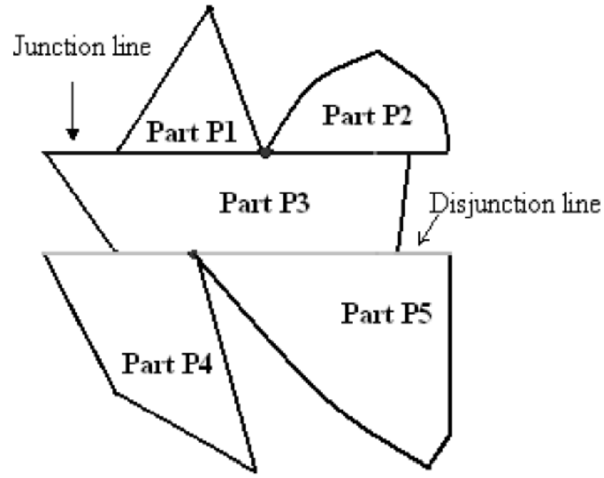



**Figure 2: Parts, junction and disjunction lines**

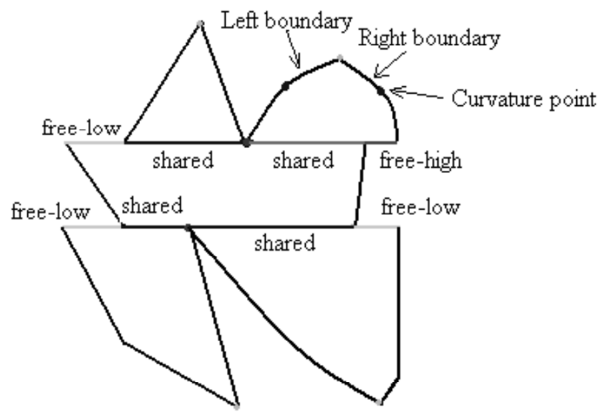



**Figure 3: Description of parts and separating lines**

Part is defined by its two boundaries (left and right) (see fig. 3). Using the curvature points located by one of the known algorithms [4], the two boundaries of each part are segmented into a set of elementary contours (line, convex and concave contours) and described by the parameters: type (line, convex or concave curve), degree of concavity or convexity, angle of inclination and length.

Separating lines are decomposed into segments. Each segment is described with three parameters: its type, the reference numbers of linked parts and its length. Three types for segment are possible: Shared if it is common for two parts, Free-High if its neighbor is the high part and Free-Low if its neighbor is the low part (see fig. 3).

The graph structure of the outline shape is translated into an XML descriptor written following the XLWDOS language. The coarse descriptor of silhouette written with the XLWDOS language is a descriptor for which only appear parts, composed parts, junction lines and disjunction lines without their geometrical descriptions. For example, the coarse descriptor of the silhouette of figure 1 is:

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

Where: <CP> and </CP> are marks for composed part, $P_i$ designates the $i^{th}$ part, $J_k$ designates the $k^{th}$ junction line and $D_n$ designates the $n^{th}$ disjunction line.

To obtain the full XLWDOS descriptor, we replace in the coarse descriptor each part or separating line by its description. We give for example the XLWDOS descriptions of parts and separating lines obtained from the XLWDOS software (see fig. 4):

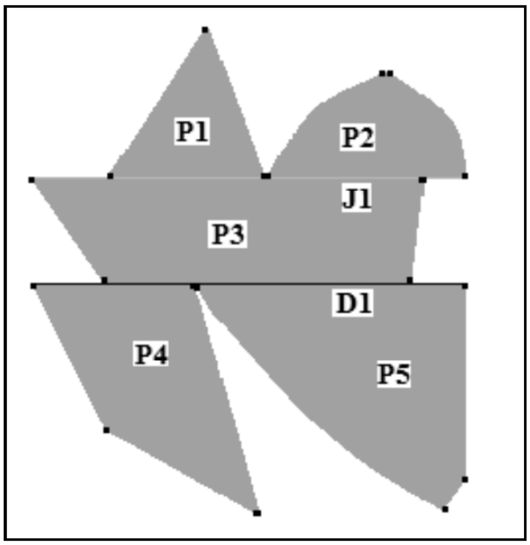



**Figure 4: The output of XLWDOS software**

<P1> <L>r 32 29 r43 58 </L>
<R>r109 79 </R></P1>

<P2> <L> r 43 109 </L>
<R>r108 17 CV 9 108 79 </R></P2>

<P3> <L>r116 80 </L>
<R>r49 82 </R></P3>

<P4> <L>r90 70 </L>
<R>r90 35r42 36 </R></P4>

<P5> <L>r126 76 </L>
<R>r69 41r39 32 </R></P5>

<J1> W P3 108 S P1 P3 112 S P2 P3 98 </J1>

<D1> H P3 66 S P3 P4 40 S P3 P5 115 </D1>

Where:

**-L, R** designate the left and right boundaries

**-r** designates a right contour (a straight contour) followed by its attributes: angle of orientation and length

**-CV** and **CC** designate respectively a convex and concave contour followed by its attributes: concavity degree, angle of orientation and length

**-S** designates a shared segment followed by its attributes: numbers of linked parts, its length

**-W**, **H** designates respectively a Free-Low and a Free-High segment followed by its attributes: the number of linked part, its length

## 3. COARSE COMPARISON OF SILHOUETTES DESCRIPTORS

### 3.1. Sensitivity to noise of XLWDOS Descriptors

The XLWDOS descriptor of silhouette is sensitive to noise. Indeed noise may modify and distorts the outlines and their descriptors such as shown in figures 5 and 6.

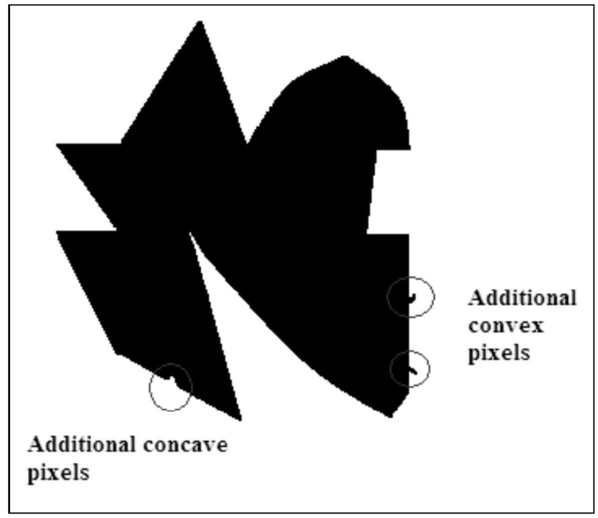



**Figure 5: Noisy silhouette**

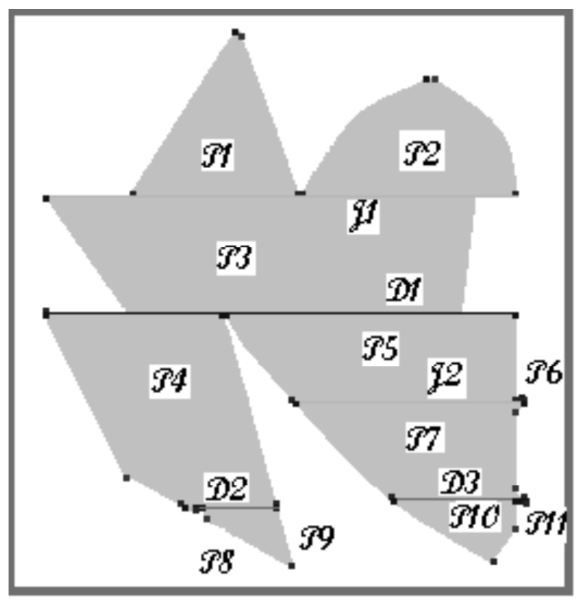



**Figure 6: XLWDOS decomposition for noisy shape**

Figure 5 represents the same shape as figure 1 with some differences due to additional pixels in the outline. Unfortunately their coarse XLWDOS descriptors are different (see fig. 6).

Indeed the coarse silhouette descriptor of figure 5 is:

<CP><CP>$P_1$ $P_2$ J1 $P_3$</CP> $D_1$ <CP>$P_4$ $D_2$ $P_8$ $P_9$</CP> <CP> <CP>$P_5$ $P_6$ $J_2$ $P_7$</CP> $D_3$ $P_{10}$ $P_{11}$</CP></CP> while the coarse silhouette descriptor of figure 1 was:

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

### 3.2. Indexing a Database of Shapes Models

The database of shapes models represented by their textual descriptors is indexed using the following data:

- The number of parts,
- The number of junction lines,
- The number of disjunction lines,
- The order of the separating lines represented by a list of '**0**' and '**1**' where '**0**' corresponds to a junction line and '**1**' to a disjunction line.
- The number of attached parts for each one of separating lines represented by a list of values $V_i$ where $V_i$ corresponds to the number of parts related to the separating line number i.
- The disposition of parts relatively to each one of separating lines represented by a list of characters '**j**', '**h**', '**w**' indicating for each separating line the disposition of the set of parts in relation (**'j'** for junction, **'h'** for free high, **'w'** for free low). A special character is inserted between the sets of characters of two separating lines.

For example, the index of the textual descriptor of the shape illustrated by figure 1 is (5, 1, 1, 01, 3, 3, wjjh&wjjw) (see fig. 7). where:

- 5 is the number of parts,
- 1 is the number of junction lines,
- 1 is the number of disjunction lines,
- 01 indicates that there is a junction line followed by a disjunction line,
- 3, 3 indicate respectively there are three parts in relation with the first and the second separating lines.

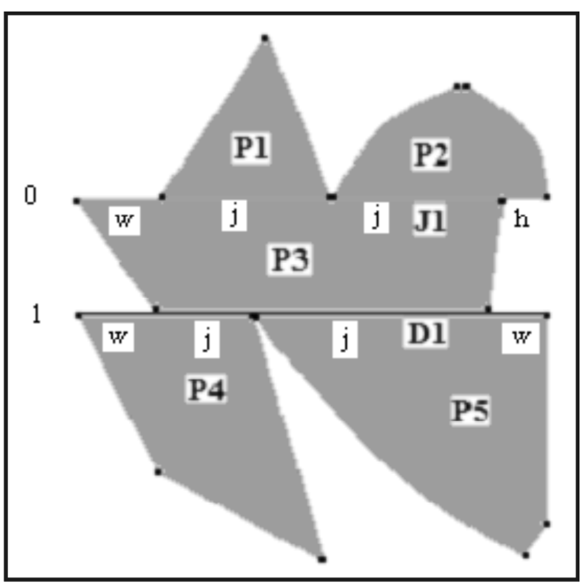



**Figure 7: Indexing a shape**

- The set of characters wjjh&wjjw indicates that in the first separating line, there are four segments with attributes w, j, j, h and in the second separating line, there are four segments with attributes w, j, j, w.

Many shapes may have the same index; the difference between them will be only in the geometry of their parts outlines.

### 3.3. Retrieval of a Shape Query using a Set of Shapes Models

#### *3.3.1. Position of the Problem*

All textual descriptors of shapes models are stored and indexed as explained in subsection 3.2.

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

As the shape descriptor used in this work is sensitive to noise, then two similar shapes may have different indexes because some additional parts and separating lines may appear or disappear in the descriptor of shape query due to the deformation of its outline.

Figures 8.a and 8.b illustrate examples of the outline shape deformation which produces additional parts and separating lines. In this case lengths of additional parts are very little comparing to other parts. The length of any part corresponds to its high according to XLWDOS description [14,15].

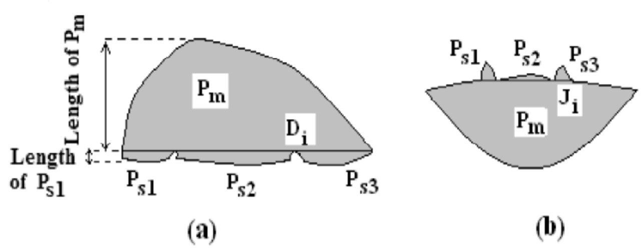



**Figure 8: Examples of noisy shapes**

Our aim is to associate the shape model to the shape query operating over their textual descriptors. We propose a set of rules in order to reduce (transform) the noisy descriptor of the shape query towards a new descriptor which would have the same index as the descriptor of a shape model.

#### *3.3.2. Transformation of the Descriptor of a Shape Query*

We define:

- A *secondary part* $P_{si}$ as the part appearing at the left of the junction line (before the junction line) or at the right of the disjunction line (after the disjunction line).

– A *main part* $P_m$ as the part appearing at the right of the junction line (after the junction line) or at the left of the disjunction line (before the disjunction line).

When the lengths of one or many secondary parts $P_{si}$ are very short (less than a threshold **L**) the associated composed part will be reduced as follow:

Let <CP>$P_{s1}$ $P_{s2}$ ... $\mathbf{P_{si}}$… $P_{sn}$ $J_i$ $P_m$</CP> or <CP> $P_m$ $D_i$ $P_{s1}$ $P_{s2}$.. $\mathbf{P_{si}}$….$P_{sn}$</CP> be a composed part and we assume that the length of ($\mathbf{P_{si}}$) is less than the fixed threshold **L.** The part $\mathbf{P_{si}}$ will be considered as noisy part and eliminated from the expression of the composed part.

Boundaries of the part ($\mathbf{P_{si}}$) should become the continuation of the right boundary of the part placed at its left. The composed part <CP> $P_{s1}$ $P_{s2}$ .. $P_{sh}$ $P_{si}$…$P_{sn}$ $J_i$ $P_m$ </CP> is then translated into <CP>$P_{s1}$ $P_{s2}$ .. $P'_{sh}$ …$P_{sn}$ $J'_i$ $P_m$</CP>.

The new right boundary of $P'_{sh}$ contains in addition boundaries of the noisy part $P_{si}$ and the segments of separating line having FreeHigh or FreeLow as attributes in relation with the noisy part.

In case where there is not part at the left of $\mathbf{P_{si}}$, then boundaries of $\mathbf{P_{si}}$ should be the continuation of the left boundary of the part placed at its right. The composed part <CP> $P_{s1}$ $P_{s2}$ .. $P_{si}$ $P_{sk}$…$P_{sn}$ $J_i$ $P_m$ </CP> is then translated into <CP>$P_{s1}$ $P_{s2}$ .. $P'_{sk}$ …$P_{sn}$ $J'_i$ $P_m$</CP>.

The new left boundary of $P'_{sk}$ contains in addition boundaries of the noisy part $P_{si}$ and the segments of separating line having FreeHigh or FreeLow as attributes in relation with the noisy part.

The description of the junction line $J_i$ must also change and becomes $J'_i$ where the segments associated to the part $P_{si}$ in $J_i$ will be considered as segments associated to the part $P_{sh}$.

In case of disjunction line, the same transformations will be applied on the descriptor <CP> $P_m$ $D_i$ $P_{s1}$ $P_{s2}$.. $\mathbf{P_{si}}$….$P_{sn}$</CP>, it will be translated into <CP> $P_m$ $D'_i$ $P_{s1}$ $P_{s2}$.. $\mathbf{P'_{sh}}$….$P_{sn}$</CP>

The description of the disjunction line $D_i$ must also change and becomes $D'_i$ where the segments associated to the part $P_{si}$ in $D_i$ will be considered as segments associated to the part $P_{sh}$.

The application of this reducing process may be recursive. For a chosen threshold L and a chosen length of noisy parts, all composed parts containing noisy parts are translated and a new descriptor of the shape query is obtained. Incrementing the length of noisy parts, we can apply again the same transformation until the descriptor of the shape query will have an index which appertains to the database index table, or the lengths of noisy parts exceeds the threshold L.

Figure 9 illustrates three examples of the reducing of the composed part <CP> P0 P1 P2 J1 P3 </CP> into a new composed part as follow:

– In Figure (9.a), P2 is a noisy part, the new composed part is <CP> P0 P’1 J’1 P3 </CP>, two different colors indicate the left and the right boundaries of the part P’1

– In Figure (9.b), P1 is a noisy part, the new composed part is <CP> P0 P’2 J’1 P3 </CP>, two different colors indicate the left and the right boundaries of the part P’2

– In Figure (9.c), P0, P1 and P2 are noisy parts, the new composed part is <CP> P’3 </CP>, where two different colors indicate the left and right boundaries of the part P’3. In this case all secondary parts are noisy.

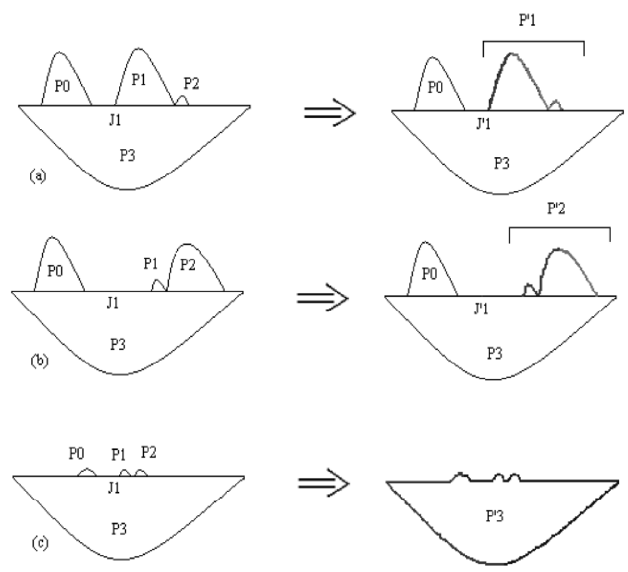



**Figure 9: Applying the reducing process in case of a junction line**

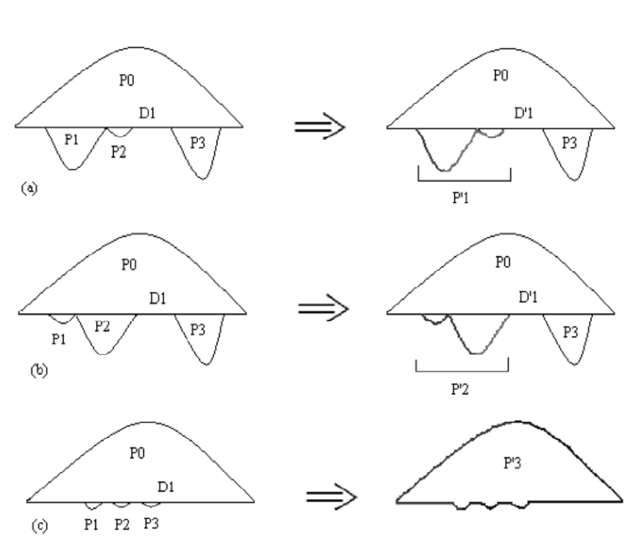



**Figure 10: Applying the reducing process in case of a disjunction line**

Figure 10 illustrates three other examples of the reducing of the composed part <CP> P0 D1 P1 P2 P3 </CP> into a new composed part as follow:

- In Figure (10.a), P2 is a noisy part, the new composed part is <CP> P0 D’1 P’1 P3 </CP>, two different colors indicate the left and the right boundaries of the part P’1
- In Figure (10.b), P1 is a noisy part, the new composed part is <CP> P0 D’1 P’2 P3 </CP>, two different colors indicate the left and the right boundaries of the part P’2
- In Figure (10.c), P0, P1 and P2 are noisy parts, the new composed part is <CP> P’3 </CP>, where two different colors indicate the left and right boundaries of the part P’3. In this case all secondary parts are noisy.

Figures 9 and 10 are drawn to better explain the reducing process. All transformations are applied only to shape descriptors. No image transformation was done.

### *3.3.3. Algorithm of Coarse Comparison and Shape Retrieval*

Let $S_R$ be the shape query and $I_R$ the index of its textual descriptor. Our aim is to find if there is the same index in the database indexes of shape models. In the favorable case such index will be found, elsewhere the transformation of the shape query descriptor is done until there is correspondence between the indexes of the two descriptors or the threshold is reached without matching indexes.

The following algorithm is explained by the diagram of figure 11;

**Algorithm**

**Begin**

1. Fixing of the threshold L, and initializing the $L_n$=0 (Ln is the length of a noisy part)

2. Compute indexes of $S_R$ and $S_M$

**While** (the indexes of $S_R$ and $S_M$ descriptors are not matched) and ($L_n$<= L)

**Do**

- $L_n = L_n + 1$
- Apply the reducing process for all parts whose length is less than L.
- Compute the index of the new descriptor

**EndDo**

**If** ($L_n$ <= L) **then** the index of shape models is found **EndIf**

**End.**

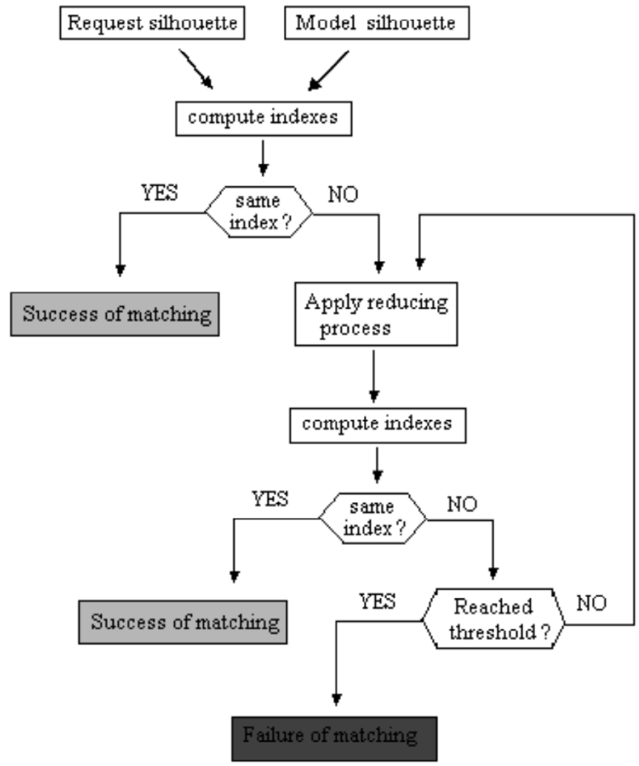



**Figure 11: Coarse comparison of shapes**

## 4. EXPERIMENTATION

### 4.1. Matching of Shapes

The first application is the matching of two images of a moving object where the extracted shapes are little different.

Figure 12 illustrates two images of a moving toy car. After silhouettes extraction using background subtraction technique, we show in the same figure their decomposition into parts, separating lines and elementary contours by means of curvature points [4].

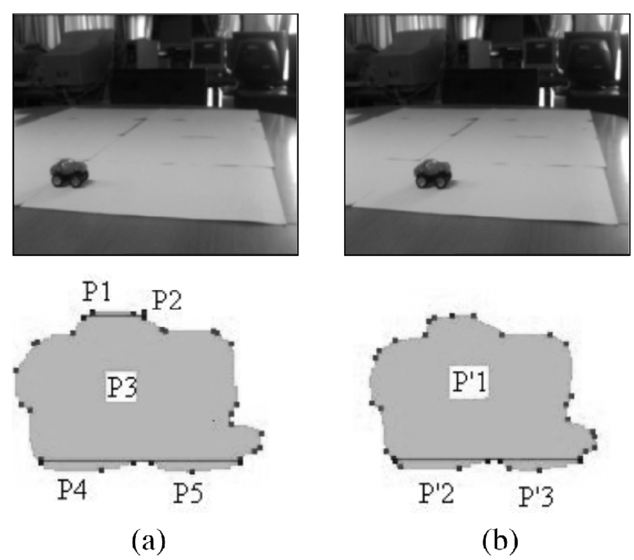



**Figure 12: Initial images and decomposition into parts of extracted shapes**

We can see that there is an additional noisy part P2 in the XLWDOS descriptor of the shape extracted from the left image.

We give the coarse XLWDOS descriptor for each silhouette:

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

(b) </CP> $P'_1$ $D'_1$ $P'_2$ $P'_3$</CP>

Their indexes are respectively:

(5, 1, 1, 01, 3, 3, wjw&jhj) and (3, 1, 1, 3, jhj). These indexes are different due to the additional noisy part P2. The coarse matching of (a) and (b) is possible because after applying the reducing process, the index of silhouette (a) becomes similar to the index of silhouette (b). Indeed the following reduction is applied:

<CP>$P_1$ $P_2$ $J_1$ $P_3$</CP> $\rightarrow P''_1$, where $P''_1 = P_1 \cup P_2 \cup P_3$

From this coarse matching, descriptors of the two silhouettes (a) and (b) are candidates for similarity measure.

### 4.2. Indexing a Database of Shapes Models

The second application of our method is the indexing of database of shapes. We used some images of database of shapes built by B. Leibe and B. Schiele [17] where each object is represented by 41 views spaced evenly over the upper viewing hemisphere. Figure 13 illustrates a sequence of shapes corresponding to images of a dog taken by rotating the camera with an angle of 24°.

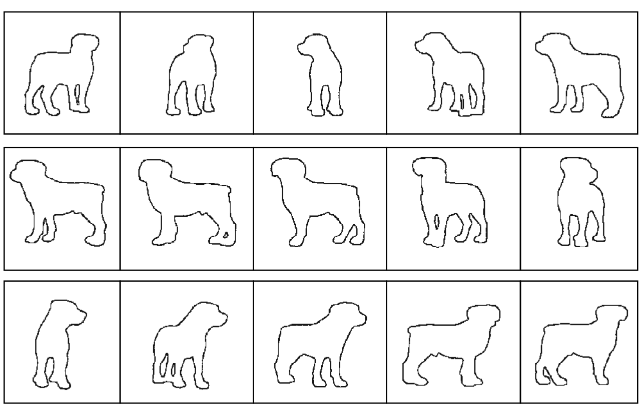


**Figure 13: A set of shapes of the database of Leibe and Schiele [17]**

In the indexing task of database of shape models, if outlines of two adjacent shapes (corresponding to two images taken in adjacent positions of rotated camera) are little different, we verify if their descriptors have the same index or if they can be reduced to new descriptors having the same index.

For example, figure 14 illustrates two successive images (S1) and (S2) of a dog and the result of their decomposition into parts and separating lines.

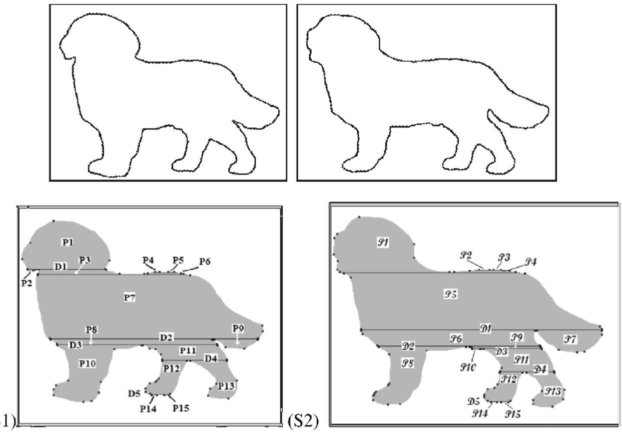



**Figure 14: Two successive images of a dog and their decomposition into parts**

The two computed XLWDOS descriptors are:

DS(S1)=<CP><CP><CP>$P_1$ $D_1$ $P_2$ $P_3$</CP> <CP>$P_4$ $P_5$ $J_1$ $P_6$</CP> $J_2$ $P_7$</CP> $D_2$ <CP>$P_8$ $D_3$ $P_{10}$ <CP>$P_{11}$ $D_4$ <CP>$P_{12}$ $D_5$ $P_{14}$ $P_{15}$</CP> $P_{13}$</CP></CP> $P_9$</CP>

DS(S2)=<CP><CP>$P_1$ <CP> $P_2$ $P_3$ $J_1$ $P_4$ </CP> $J_2$$P_5$ </CP>$D_1$ <CP>$P_6$ $D_2$ $P_8$<CP>$P_9$ $D_3$ $P_{10}$ <CP>$P_{11}$ $D_4$ <CP>$P_{12}$ $D_5$ $P_{14}$ $P_{15}$ </CP> $P_{13}$ </CP></CP> </CP> $P_7$ </CP>

We give XLWDOS indexes of silhouettes (S1) and (S2) of figure 14 respectively:

(15, 1, 6, 1011111, 3, 3, 3, 3, 3, 3, 3, jhj&hjhjh&jwj&wjwjw&jwj&jwj&jwj) and

(15, 2, 5, 0110111, 3, 3, 3, 3, 3, 3, 3, jwj&jwj&jwj&jwj&hjwj&jwjw&wjwjw)

The indexes are different; therefore we apply the reducing process on the descriptors which gives the following index for each silhouette.

(8, 1, 4, 01111, 3, 3, 3, 3, 3, jwj&jwj&wjwj&hjwj&wjwjw).

Indeed, we can see that:

– $P_2$ is noisy part; we can then reduce the composed part <CP>$P_1$ $D_1$ $P_2$ $P_3$</CP> to a single part $P'_2$ whose boundary is the union of boundaries of the three parts.

– $P_{10}$ is noisy part; we can also reduce the composed part <CP> $P_9$ $D_3$ $P_{10}$ $P_{11}$</CP> to a single part $P'_{11}$ whose boundary is the union of boundaries of the three parts.

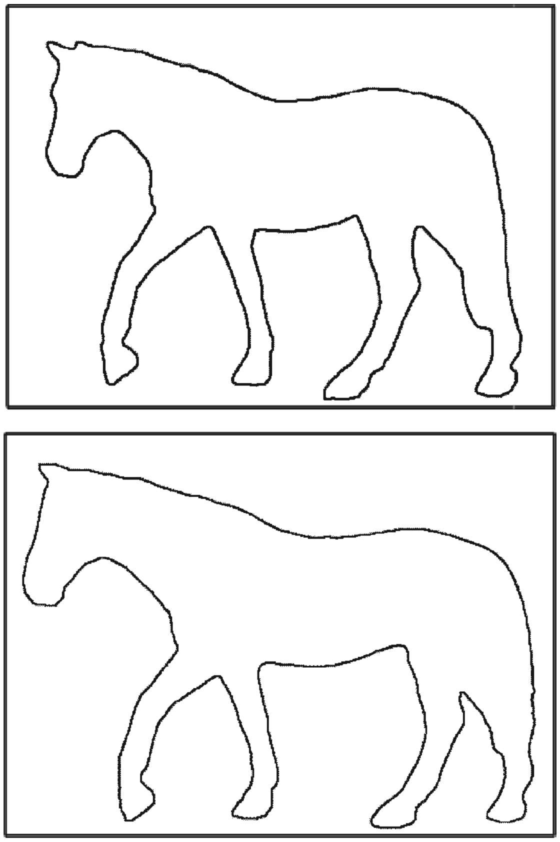
Figure 15 illustrates two different images of a horse but not successive. The computation of their indexes gives the following results (figure 16):

The index of the first shape (S1) is:

(17, 2, 6, 00111111, 3, 3, 3, 3, 3, 3, jwj&jhj&jwj&jhj&jhj&jhjw&jwj&jwjw).

After applying the reducing process on the descriptor, this index becomes:

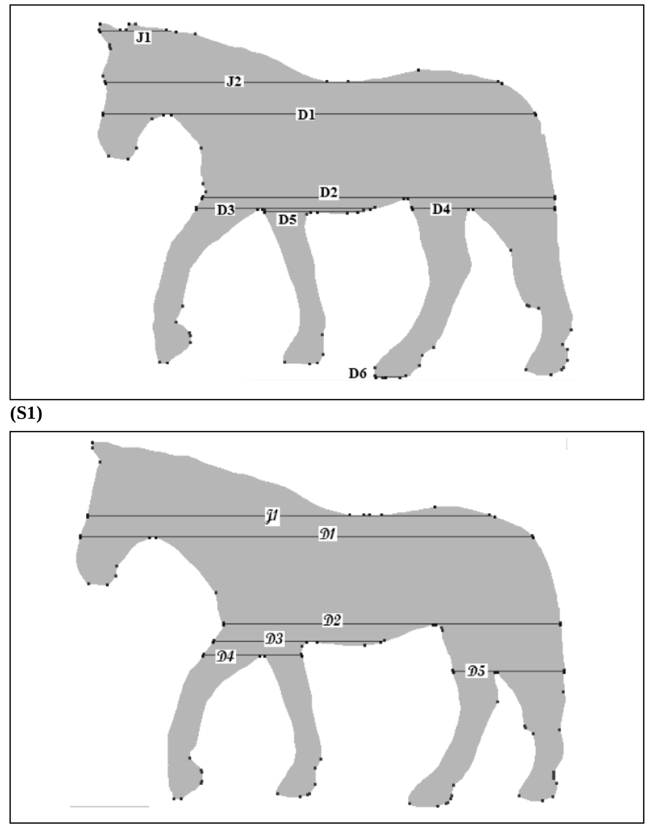
(12, 2, 4, 001111, 3, 3, 3, 3, jwj&jhj&jwj&jhj&jhj&jhjw).



(S1)

(S2)

**Figure 15: Two images of a horse**




(S1)

(S2)

**Figure 16: Decomposition of the two shapes into parts**

The index of the second shape (S2) is:

(13, 1, 5, 011111, 3, 3, 3, 3, 3, jwj&jhj&jhj&jhj&jhj&jhj).

There are no noisy parts in the second shape (no part smaller than the reducing threshold), therefore the second index can not be modified, and the two shapes can not be matched.

### 4.3. Shape Query using Database of Shapes Models

We used the 50% of shapes of the database of B. Leibe and B. Schiele [17] as shapes models and the other shapes as shapes queries. After many experiments, we fixed the threshold L to 3 pixels.

Tables 1 and 2 illustrate examples of obtained results.

**Table 1**
**Results of shape query (Queries cups in blue and models in green)**

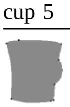
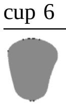
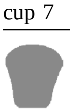
| *Models* <br> *Queries* |  <br> *cup 1* |  <br> *cup 2* |  <br> *cup 3* |  <br> *cup 4* |
|---|---|---|---|---|
|  <br> cup 5 | Matching using 1 iteration of reducing process | Matching using 1 iteration of reducing process | Matching Failure | Matching Failure |
|  <br> cup 6 | Same indexes without reducing | Same indexes without reducing | Matching Failure | Matching Failure |
|  <br> cup 7 | Matching Failure | Matching Failure | Same indexes without reducing | Matching using 1 iteration of reducing process |
|  <br> cup 8 | Matching Failure | Matching Failure | Matching using 1 iteration of reducing process | Same indexes without reducing |

**Table 2**
**Results of shape query (Queries cars in blue and models in green)**

| *Models*<br>*Queries* | <br>*Car 1* | <br>*Car 2* | <br>*Car 3* | <br>*Car 4* |
|---|---|---|---|---|
| <br>Car 5: | Same indexes without reducing | Matching using 2 iterations of reducing process | Matching Failure | Matching Failure |
| <br>Car 6 | Matching using 3 iterations of reducing process | Matching using 1 iteration of reducing process | Matching Failure | Matching Failure |
| <br>Car 7 | Matching Failure | Matching Failure | Matching using 3 iterations of reducing rocess | Matching using 3 iterations of reducing process |
| <br>Car 8 | Matching Failure | Matching Failure | Matching using 2 iterations of reducing process | Same indexes without reducing |

Table 1 shows the following results:

- When the shape query has the same index than the model silhouette (for example cup6 with cup 1), we have matching success, without applying the reducing process.
- For the query cup 5 and the model cup2, we obtain matching success, after applying the reducing process.
- For the query cup 5 and the model cup3, we could not find same indexes while applying the reducing process until reaching the threshold.

Same results are obtained with the other queries (cup 6 until cup 8), and with other objects in table 2.

## 5. CONCLUSION

We presented in this paper a new method for coarse silhouettes comparison and shape indexing. The silhouettes are described according to XLWDOS language. We have proposed a reducing technique allowing the descriptors comparison taking into account the noise on the images. To validate the proposed approach, real image silhouettes have been used. The next step of our work is the full matching and the computation of the similarity measure between two coarsely matched descriptors.

### REFERENCES

[1] P. Aigrain, V. Longville, A connection graph for User Navigation in a Large Image Bank, Proc. RIAO'91, P. 67-84. Barcelona, Spain, Avril, 1991.

[2] B. Bukhardt, S. Siggelkow. Invariant features in pattern recognition –fundamentals and applications. In C. Kotropoulos and I. Pitas, editors, Nonlinear Model-Based Image/ Video processing and Analysis, pp. 269-307. John Wiley & Sons, 2001.

[3] B. Caputo, J. Hornegger, D.Paulus, H. Niemann. A spin-glass markov random field for 3-D object recognition. Technical Report LME-TR-2002-01, Erlangen-Nurnberg University, 2002.

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

[5] 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.

[6] A. P. Dempster, N. M. Laird, D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of Royal Statisical Society*, **39**(1) : 1-38, 1997.

[7] M. D. Ennis. A multidimensional stochastic theory of similarity. *Journal of Mathematical Psycology*, **32**: 449-465, 1988.

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

[9] P. Gros, New descriptors for Image and Video Indexing, Dagstuhl Seminar on Content-Based Image and Video Retrieval, Dagstuhl, Germany, Dec. 1999.

[10] D. Keysers, T.Deselaers, T.M. Breuel, Optimal Geometric Matching for Patch-Based object Detection, *Electronic Letters on Computer Vision and Image Analysis* **6**(1): 44-54, 2007.

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

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

[13] Y. Lamdan, H. J. Wolfson, Geometric hashing: a general and efficient model-based recognition scheme, in *IEEE conference on Computer Vision and Pattern Recognition*, 1988.

[14] S. Larabi, S. Bouagar, F. M. Trespaderne, E. F. Lopez, LWDOS: Language for Writing Descriptors of Outline Shapes, In the LNCS proceeding of Scandinavian Conference on Image Analysis, June 29 - July 02, Gotborg, Sweden, 2003.

[15] S. Larabi, Textual description of images, In the proceeding of International Conference CompImage (Computational Modelling of objects represented in images, Fundamentals and Applications), Coïmbra, Portugal, October 2006.

[16] L. J. Latecki, R. Lakamper, Application of Planar Shape Comparison to object Retrieval in Image Databases. *Pattern Recognition*, **35**, 15-29, 2005.

[17] B. Leibe and B. Schiele, *Analyzing Appearance and Contour Based Methods for Object Categorization.* In International Conference on Computer Vision and Pattern Recognition (CVPR'03), Madison, Wisconsin, June 2003.

[18] J. Li, R. M. Gray, Image Classification by a Two Dimensional Hidden Markov Model, *IEEE Transactions on Signal Processing*, **48**(2), 517-533, 1997.

[19] J. Li, J. Z. Wang, Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach. *IEEE Transactions PAMI* **25**(9), 2003.

[20] T. Lindeberg. Scale-space Theory in Computer Vision. The Kluwer international series in Engineering and Computer Science. Kluwer Academic Publishers, Boston, USA, 1994.

[21] T. Lindeberg. Feauture detection with automatic scale selection. Technical Report ISRN, Dept. of Nmerical Analysis and Computing Science, KTH, May 1996.

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

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

[24] R. Nussinov, H.J.Wolfson, Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques. *Proc. Natl. Acad. Sci. Usa*, **88**, 10495-10499, 1991.

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

[26] J. Posl,, H.Niemann. Wavelet features for statistical object localization without segmentation. ICIP. V. 3, 170-173, Santa Barbara, California, USA, October 1997.

[27] T. R. Read, J. M. Hans du buf. A review of recent texture segmentation and feature extraction techniques. CVGIP: *Image Understanding*, **57**(3): 359-372, 1993.

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

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

[30] M. Shael, S. Siggelkow. Invariant grey-scale features for 3D sensordata. In proceedings of the International Conference of Pattern Recognition, **2**, 531-535, Barcelona, Spain, 2000.

[31] J. Shi, J.Malik. Normalized cuts and image segmentation. In proc. of IEEE CVPR Puerto Rico, 731-737, 1997.

[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] S. Siggelkow, H. Bukhardt. Image Retrieval based on local invariant features. In proceedings of the IASTED International Conference on Signal and Image processing (SIP) 1998, 369-373, Las Vegas, Nevada, USA.

[34] M. J. Swain, D. H. Ballard. Color indexing. *International Journal of Computer Vision*, **7**(1): 11-32, 1991.

[35] J. Z. Wang, J. Li, G. Wiederhold, "SIMPLIcity: semantics-sensitive Integrated Matching for Picture Libraries" *IEEE Transactions PAMI*, **23**(9), 2001.

[36] A. Wardhani, T.Thomson, "content-Based image retrieval using category-based indexing" ICME 783-786 27-30 june Taiwan 2004.

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

[38] X. Zabulis, J.Sporring, S.C.Orphanoudakis. Scal summarized and focused browsing of primitive image content. In Visual 2000, Lyon, France, 269-278, 2000.

[39] S. C. Zhu and A. L. Yuille, FORMS: A flexible object recognition and modeling system, Fifth International Conference on Computer Vision, June 20-23, M.I.T. Cambridge, 1995.