The 13th International Arab Conference on Information Technology ACIT'2012 Dec.10-13
ISSN : 1812-0857

Silhouettes Retrieval based on the Quad-tree Structure following an XML Description

Saliha Aouat and Slimane Larabi

LRIA Laboratory, Computer Science Department, USTHB University, Algiers, Algeria
{saouat, slarabi}@usthb.dz

Abstract.

In this paper we perform image retrieval using efficient similarity measures extracted from textual descriptors of shapes. The features are Geometric Quasi-invariants that vary slightly with the small change in the viewpoint. The quad- tree structure is used to improve the processing time for the indexing process and facilitate the recognition process. Our approach is applied on the outline shapes of 3D objects.

Keywords: *Geometric Quasi-invariants, textual descriptor, quad-tree structure, image retrieval.*

1. Introduction

Geometry-based methods are feature-based methods that extract points from the image (usually edge or corner points) and reduce the problem to point set matching. Users are more interested in matching and retrieval by shape than by color and texture [1-13]. Segments are particularly interesting features [14-16] because of their robustness to noise and their connectedness constraint that reduces the possibility of false matches. They also have the properties to vary slightly with a small change in the viewpoint, and to be invariant under similarity transform of the image [17].
Since these features are widely used to match objects, we need to use geometric invariant features. We consider therefore as 2D image features; the intersecting segments and we transform them to pairs of quasi-invariants features.

Our aim is to develop a complete 3D object recognition system. This system is based, first, on the extraction of pairs of quasi-invariants from a textual description of a shape. Second, it is based on the matching of geometric quasi-invariants between the query and the models. We use also the quad-tree structure to improve the processing time of our recognition process. This induces that recognition and indexing is restricted to 2D-2D matching. Instead of directly interpreting 3D object information, we store several 2D features (quasi-invariants) of a 3D object, and perform the object retrieval in the 2D indexes representation space.

The paper is structured as follows:
In the second section, we present the Geometric Quasi-invariants. The textual description of shapes is given in the third section. In the fourth section we show the decomposition of objects shapes following the Quad-tree structure. The fifth section is about our approach to recognize and retrieve the best model for the given query. In the last experimental section, we use real images of a well known database and discuss the results of our approach on real images

2. Geometric Quasi-invariants

The quasi-invariants (ρ, θ) are defined as : the angle θ between the intersecting segments, and the segments length ratio ρ.

$$\rho = \frac{\overrightarrow{a_0 a_1}}{\overrightarrow{a_0 a_2}} \qquad \theta = \arccos \frac{\overrightarrow{a_0 a_1} . \overrightarrow{a_0 a_2}}{\left\| \overrightarrow{a_0 a_1} \right\| . \left\| \overrightarrow{a_0 a_2} \right\|}$$

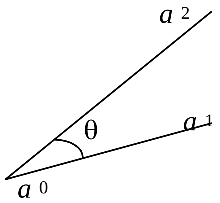

Figure 1: Geometric quasi-invariants (ρ , θ)

The (ρ, θ) pairs found in each image (see Figure 1) vary slightly with a small change in the viewpoint, and are invariant under similarity transform of the image [16,17]. In our approach, the (ρ, θ) pairs will be computed from the textual description of shapes.

3. Textual Description of outline shapes

Let be the silhouette illustrated by Figure 2. This silhouette is partitioned onto parts, junction and disjunction lines [18,19]. Each element is described geometrically by means of contour boundaries and of segments giving the description of a silhouette.

At each junction or disjunction line is associated a composed part defined as a part joined to other parts via junction or disjunction line. This is the case of P1, P2, JL1 and P4 that define a composed part. This composed part is joined to P3, P5 and JL2 to define a new composed part corresponding to the silhouette.

We write the global XML descriptor of this silhouette as follow:

<DS><name>Example of silhouette</name>
<CP><CP> P1 P2 JL1 P4 </CP> P3 JL2 P5 <CP> </DS>

The part is defined by its two boundaries (left and right). The boundaries are segmented into a set of primitives (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 (see Figure 2).

The length of a primitive (segment or curve) refers to the high of the primitive, except when the primitive is horizontal.

The separating lines are decomposed into segments. Three types of segments are possible: Shared (designated by 's') if the segment is common for two parts, Free-High (designated by 'h') if its neighbor is the high part, and Free-Low (designated by 'w') if its neighbor is the low part.

To obtain the full XML descriptor of a shape, we replace in the global descriptor, all parts and lines by their descriptors.

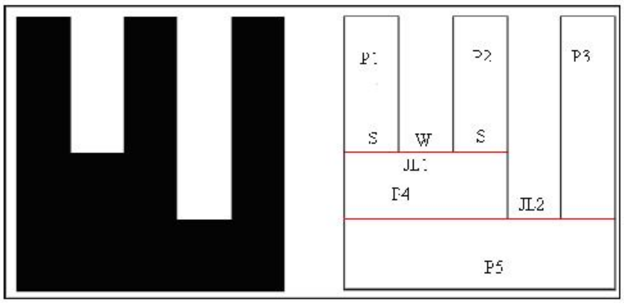

Figure 2: Example of silhouettes and results of its partition.

For the silhouettes in Figure 2, we obtain the following XML descriptor:

<DS><name>Example of silhouette</name>
<CP><CP>
<P1><L>r 90 80</L><R>r 180 40 r 90 80</R></P1>
<P2><L>r 90 80</L><R>r 180 40 r 90 80</R></P2>
<J>s P1 P4 40 w P4 40 s P2 P4 40</J>
<P4><L>r 90 40</L><R>r 90 40</R></P4></CP>
<P3><L>r 90 120</L><R>r 180 40 r 90 120</R></P3>
<J>s P4 P5 120 w P5 40 s P3 P5 40</J>
<P5><L>r 90 40 r 180 200</L>
<R>r 90 40</R> </P5></CP></DS>

In our approach, and in order to reduce the processing time of the recognition process, we perform the indexing process using the quad-tree structure before extracting the textual descriptors, and Geometric quasi-invariants.

4. Decomposition of shapes following the quad-tree structure

When decomposing the image following the quad-tree structure. We divide the picture area into four sections. Those four sections are then further divided into four subsections. We continue this process, repeatedly dividing a square region by four (see Figure 3).

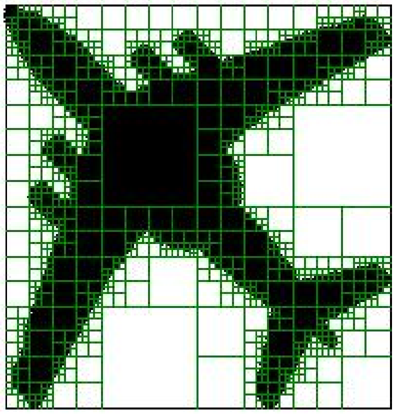

Figure 3: Image decomposition following the Quad-Tree structure

We have demonstrated in a previous work [20] that two levels of the quad-tree (16 quadrants) are sufficient to reduce the processing time and discard many models of the database which are far from the query. So, after this step, only few models (for a given query) are maintained. This step is very important especially for large image databases.

5. Image Retrieval and Recognition

Our recognition process aims to retrieve and recognize the best model for the given query. It is an appearance-based method in which 3D objects are represented by multiple views. The recognition is simplified by performing 2D matching and retrieval between the query and all 2D models of all objects in the database. Finally, the best model for the given query will be found after applying similarity measures.

It is important to precise that in appearance-based methods, the query is theoretically situated between two models of the database (except when the query is a view of an object that is not in the database).

Each step of our approach requires the determination of intervals, or thresholds, allowing the comparison of models and queries. The determination of thresholds must be done in an off-line study by using hundreds of nearby 2D models of many objects. To retrieve the best model, in an on-line study, the comparison between the query and models is done. If the distance between a model and a given query is outside such intervals, the model will be discarded. In opposite, if many models verify the threshold values, we sort those models, following an efficient similarity measure, in order to keep the best one which is close to the query.

5.1 Off-line process

We consider hundredth of objects (See Section 6) and we study the variation of characteristics and features between nearby images in order to find thresholds of similarity values used in our approach.

In order to compute the thresholds, we take into account all pairs of images. Each pair contains nearby models, and then, we compute the difference between extracted features between all nearby images.

A) Variation of the filling rate of the quadrants

We define the filling rate as the percentage of black pixels in a quadrant relatively to the number of all pixels in the same quadrant (see Figure 4).

The filling rate of black quadrants is 100% because the quadrants are inside the silhouette. However, the filling rate of white quadrants is 0% because the quadrants are outside the silhouette.

The filling rate is given by.

$$\text{Rate} = \frac{\text{Number of black pixels}}{\text{Number of black pixels} + \text{Number of white pixels}} * 100$$

Figure 4: Decomposition of a silhouette into 16 quadrants

Evaluations (see experimental section) show that the intervals of filling rate to be maintained for the online study is [0,10].We have chosen 16 quadrants (two levels of the quad-tree), because this reduces the processing time and permits to discard the models that are far from the query [20].

B. Computation of quasi-invariants from the textual description

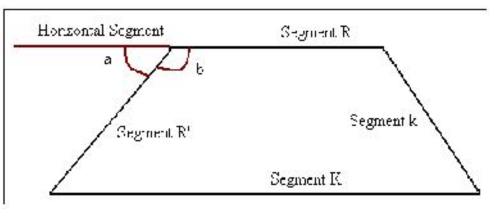

Figure 5: Determination of Quasi-invariants from XLDWOS descriptors

Let us consider the shape in Figure 5. In the textual description curves are approximated by segments and all angles are given between the primitive and the horizontal axis [18]. (see section 3).

In order to find all angles between two successive segments we have to deduce them from the textual descriptor.

For example in Figure 5, angle (a) is given using the textual description [18]; it is situated between the segment R' and the horizontal axis. We compute angle (b), which is between two successive segments (one of our quasi-invariants). Indeed for the quasi-invariants, we need all angles between segments, and the length ratio of such successive segments (see section 2).

5.2 On-line process

In the online process, we perform the recognition process. For a silhouette query we have to find the most similar images in the database, following three steps.

The first step is to decompose the query into 16 quadrants (see Figure 4), and compare the filling rates with those of the models. All models verifying thresholds, determined in the subsection 5.1, will be selected and used for the next step.

The second step is to compute the global descriptor of the query. All models with same descriptors as the query will be selected for the next step.

The last step is to compute quasi-invariants of the query and select, then, all models verifying the thresholds determined in the offline study.

If many models are selected after these steps, the best model is that minimizing the Euclidian distance with the query. This distance is computed for both angle and segment ratio.

$$\sum_{i=1}^{m}(xr - xm)^2$$

where xr represents the quasi-invariants (angle or segment ratio) of the query, and xm represents those of models.

6. Experimentation

Different objects have been used to validate the proposed methods. We use the database of shapes built by B. Leibe and B. Schiele [21]. This database contains 80 different objects. Each object is represented by 41 views spaced evenly over the upper viewing hemisphere. Some objects are shown in Figure 6.

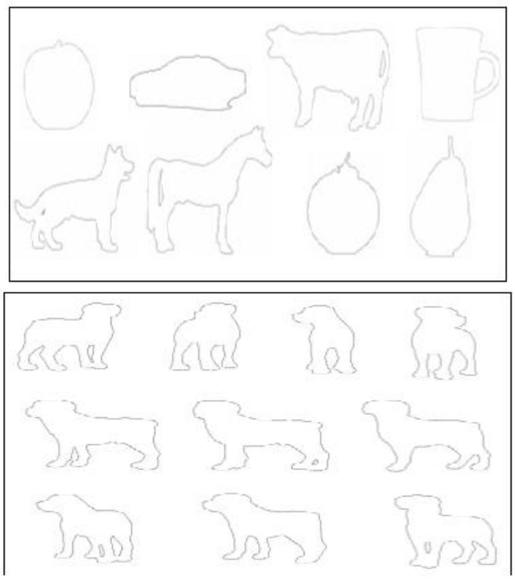

Figure 6: A set of shapes of the database of Leibe and Schiele and some views of an object [21]

6.1 Evaluation of the filling rate of quadrants

We consider all successive images of the database, and we compute the filling rate of all quadrants between all nearby models. Each quadrant k of the first model i is compared with the quadrant k of the model i+1, (same positions of compared quadrant). Experiments show that more than 95 % of nearby images have a difference of filling rate between 0 and 10. (see Figure 7)

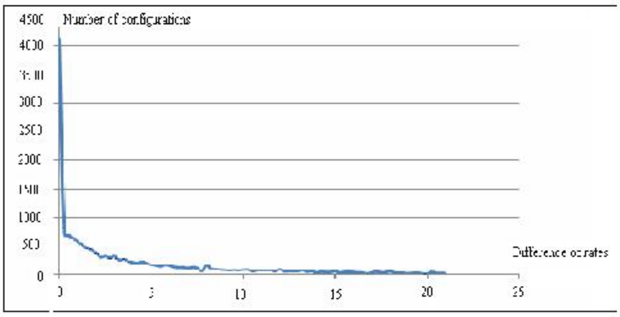

Figure 7: Thresholds of filling rate. (Number of configuration in this graph refers to the number of compared quadrants)

6.2 Evaluation of quasi-invariants differences

We extract all quasi-invariants ρ and θ from models then we compare all those extracted from neighbor views. Experiments (see Figure 8) show that more than 90% of Quasi-invariants configurations are less than (ln(ρ) = 0.21 and θ =17.8).

(we use ln(ρ) instead of ρ, because ln(ρ) follows a uniform distribution [16]).

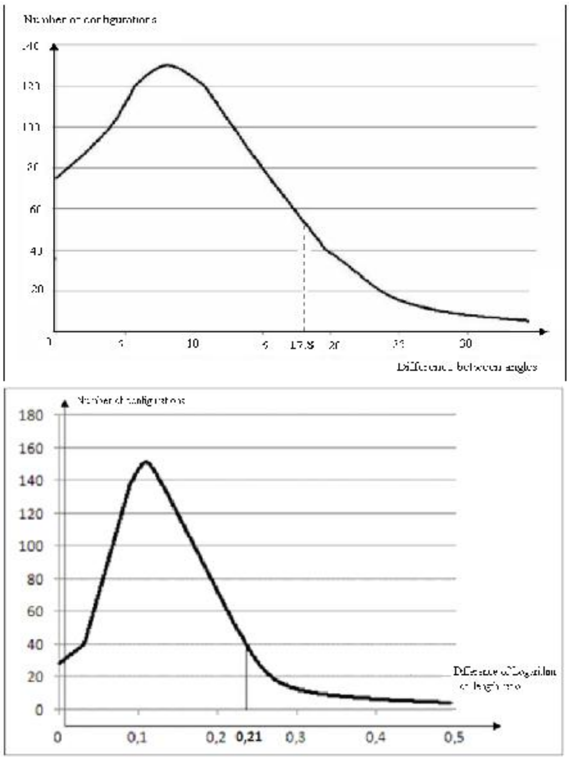

Figure 8: Difference of quasi-invariants between successive images

6.3 Retrieval process

The query is processed as explained in sub-section 5.2, after the three steps of the online process the best model for the query is retrieved. Three examples are given in Figure 9.

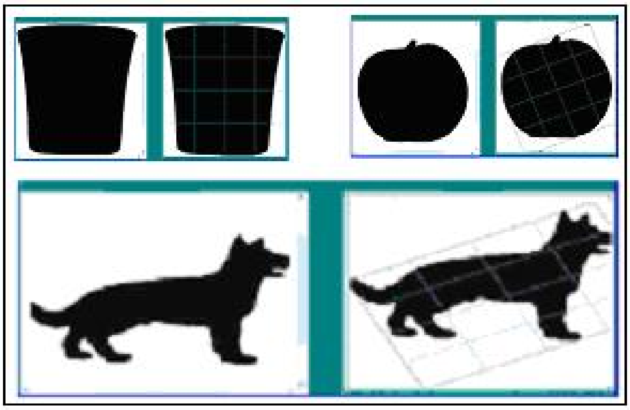

Fig. 9. Examples of the retrieved shapes (The best model in the left, the query in the right)

7. Conclusion

In this paper, we proposed a new method for silhouettes retrieval. The silhouettes are written following Textual Descriptors of shapes.

We have seen the importance of applying efficient similarity measures to achieve this process. The used measures were:

- The filling rates of the query quadrants to reduce the number of models maintained for the following processes.
- The Geometric Quasi-invariants in order to efficiently compare the query silhouettes geometry with the models geometry.

Conducted experiments, performed on a known database, showed the method efficiency and its usefulness to resolve the problem of the retrieval process.

References

[1] Arandjelovic, R., Zisserman, A.:Efficient image retrieval for 3D structures. BMVC: 1-11, 2010

[2] Cronin, T.M.: isualizing concave and convex partitioning of 2D contours. Pattern Recognition Letters. 24 : 429-443. 2003

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

[4] Keysers, D., Deselaers, T., Breuel, T.M.: Optimal geometric matching for patch-based object detection. Electronic Letters on computer vision and image analysis, 6(1): 44-54, 2007.

[5] Latecki, L.J., Lakaemper, R., Wolter, D.: Optimal partial shape similarity. Image and vision computing, 23(2): 227-236, 2005

[6] Ma, T., Latecki, L.J.: From partial matching through local deformation to robust global shape similarity for object detection. CVPR: 1441-1448, 2011

[7] Mokhtarian, F.: Silhouette-Based isolated object recognition through curvature scale space. IEEE Trans. on PAMI, 17(5): 539-544, 1995.

[8] Mokhtarian, F., Mackworth, A.K.: A Theory of Multiscale, Curvature-Based Shape Representation for Planar curves. IEEE Trans. On PAMI, 14(8), 789-805, 1992

[9] Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. Computer Vision and Pattern Recognition, 2007

[10] Trinh, N.H., Kimia, B.B.: Skeleton Search: Category-Specific Object Recognition and Segmentation Using a Skeletal Shape Model. IJCV 94(2): 215-240, 2011

[11] Wang, X., Bai, X., Liu, W., Latecki, L.J.: Feature context for image classification and object detection. CVPR 2011: 961-968, 2011

[12] Yang, X., Bai, X., Latecki, L.J., Tu, Z. : Improving shape retrieval by learning graph transduction. ECCV, In LNCS springer, 2008

[13] Zaeri, N., Mokhtarian, F., Cherri, A.: Binarized eigenphases applied to limited memory recognition systems. Pattern analysis and application, Springer London, 11(3), 2008.

[14] Lamdan, Y., Wolfson, H.J.: Geometric hashing: A general and efficient model based recognition scheme, ICCV'88 Second International Conference on Computer Vision, 1988

[15] Matusiak, S.: New Multiscale planar Shape invariant Representation under a General Affine Transformations, 1999

[16] Lamiroy, B., Gros, P,: Rapid Object Indexing and Recognition Using Enhanced Geometric Hashing, 4th European Conference on Computer Vision, Cambridge, England,1996

[17] Gros, P., Quan, L.: 3D Projective Invariant from Two Images, Proc. of the SPIE Conference on Geometric Methods in Computer Vision II, pp 75-86, San Diego, California, USA, July 1993

[18]Larabi, S. Bouagar, S.,: An XML Language for Writing Descriptors of Silhouettes, paper submitted to GVIP 2005.

[19] Aouat, S., Larabi, S.: Matching Descriptors of Noisy Outline Shapes. Int. Journal of Image and Graphics. World Scientific Publisher. 2010

[20] Aouat, S., Larabi, S.: Indexing binary images using quad-tree Decomposition. International Conference on System, Man, and Cybernetics. Istambul, Turkey, 10-13, 2010

[21] Leibe, B., Schiele, B.,: Analyzing Appearance and Contour Based Methods for Object Categorization. In Int. Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, 2003.