A new shape descriptor using XML language

Noreddine GHERABI
Department of Mathematics and Computer Science,
Université Hassan 1er, FSTS, LABO LITEN
Settat, Morocco

Mohamed BAHAJ
Department of Mathematics and Computer Science,
Université Hassan 1er, FSTS, LABO LITEN
Settat, Morocco

Abstract— **In this paper we present a new method for writing an approximate description of 2D/3D objects in text form using XML language. First, the outline of the shape is divided into parts according to N angular and M radial sectors (or bin), each Bin contains a portion of the contour; this portion is divided at the inflexion points into convex and concave sections and local features are extracted for these partitions, then this features are coded and stored in a XML data structure.**

In this work we study the possibility of using this textual description in the shape recognition by exploiting the variables and the values stored in the XML file. Experiments on natural silhouette database and MPEG-7 shape databases demonstrate the effectiveness of the proposed method relative to the standard assignment problem.

Keywords: Shape descriptor, Similarity Search, ARP, XML

I. Introduction

Searching for images using shape features has attracted much attention. Many shape description techniques and features have been developed in the past and recent years. Some popular descriptors include shape signature, signature histogram, shape invariants, shape context, shape matrix, curvature scale space method, Zernike moments, spectral features [1,2,3,4,5]. These various shape features are often evaluated by how accurately they allow one to retrieve similar shapes from a designated database. There are two major approaches for shape representation in the literature: one approach is boundary based and uses contour information [6, 7], and the second approach needs a holistic representation, requiring general information about the shape. Attalla and Siy [8] have presented a polygonal approximation of shape contours that divide it into equal segments and all segments will serve as local features that will represent the shape. Different search techniques were investigated to retrieve shapes from databases. These research techniques used to extract shape descriptors of each shape that is in the database and use these descriptors as indices in the database.

In this paper, we propose a new approach to describe the shape using XML. With XML approach, most system can perform a search based on the shape information presented in the XML format. Thus, the proposed system will be able to perform searching on most image databases that supports XML querying and returns more accurate results to the users. There is not much work that uses an XML description, this technical description of the shape using XML has been used rarely in recent years despite its efficiency and reliability. Man Hing [9] uses this technique to extract the features information of the shape and represent this information in an XML format.

Our algorithm analyzes the shape and its contour is recovered and represented by a set of points, then the contour of the shape is represented in a surrounding circle partitioning it to a set of sectors. After, the local and global information of the contour are extracted and are stored in an XML file.

II. our apprOach

As will be evident in the detailed description of our method, the proposed approach aims at recovering the basic features of shapes, mixing local and global information. The use of precise geometrical quantities is an important variable as a feature of shape. We assume that the shapes have already been extracted from images and are represented by their bounding contours.

A. Angular radial partitioning (ARP).

The first step is to analyze the contour of the shape to be studied. The contour is retrieved and normalized to a set of points. This normalized contour is for feature extraction.

Our method uses the ARP technique [10] to transform the shape data into a new structure that supports

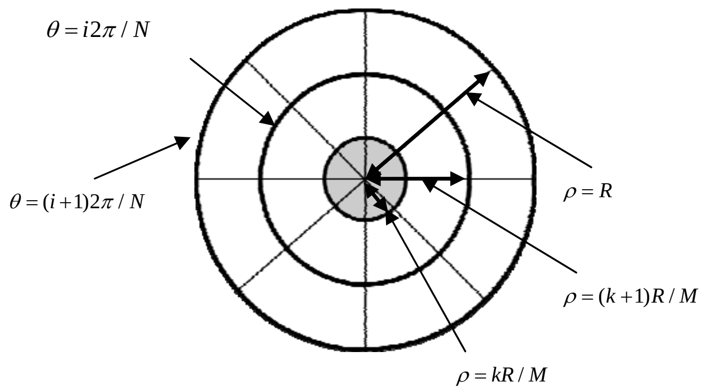


Fig. 1: Angular Radial partitioning of an shape to N angular and M radial sectors where k=0,1,2….M and i=0,1,...N

measuring the similarity between shapes in an efficient manner, where the contour of the shape is represented in a surrounding circle partitioning it to M × N sectors (or Bin), where M is the number of radial partitions and N is the number of angular partitions.

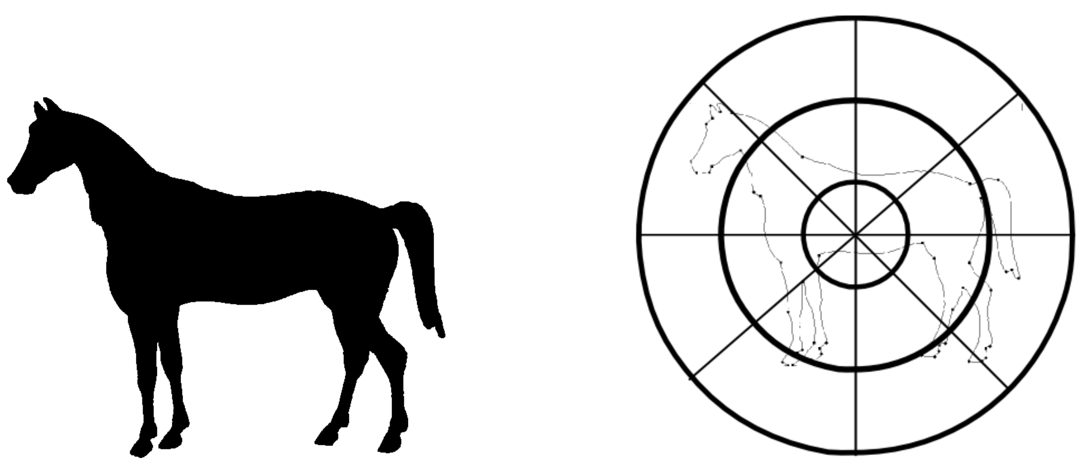
Geometrically, the angle between adjacent angular partitions is $\theta = 2\pi / N$ and the radius of successive concentric circles is $\rho = R / M$; where R is the radius of the surrounding circle of the image. (See Fig.1).

B. Shape Descriptor.

The principle consists in placing the outline in the center of a region with a surrounding circle partitioning it to M × N sectors (or **Bin**) (See Fig.2). This technique is used in shape contexts but in a different way to obtain a good correspondence between a set of points [11,12,13]. To make the method more robust to translations and to achieve scaling invariance, the radius of the largest circle is defined as the radius of the circle that surrounds the shape (M and N are fixed)**.**

The sweep of any shape begins with the first sector (Bin (1)) near the center O and following a direction of clockwise, where O is the center of the surrounding circle.

The description of the shape is the grouping of descriptions of its elements. These elements are each in a specific **Bin** (Fig.3). Each **Bin** contains a portion of the contour; this portion is divided at the inflexion points {Pi} into convex and concave sections and the information about sections are extracted in order to provide a semantic content to the outline shape. We use Chetverikov's algorithm [14] for the detection of the inflection points of the contour. This algorithm is available on the Web.



(a) (b)

Fig 2. (a) Example of the MPEG-7 Shape and (b) its presentation in the surrounding circle with inflexion points.

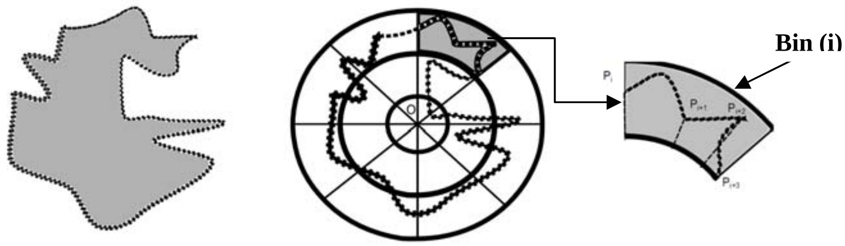
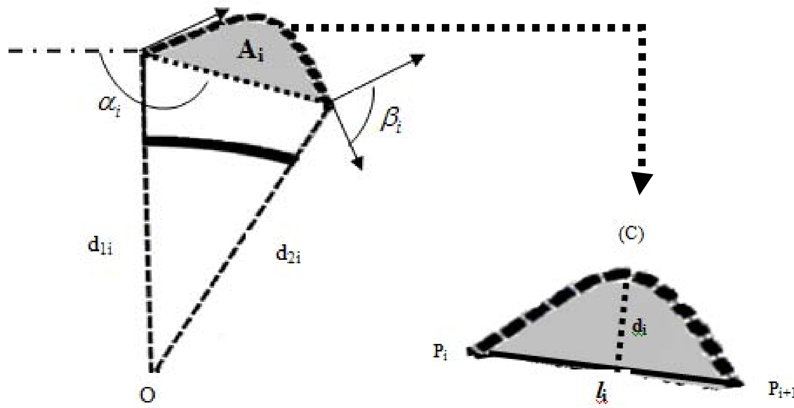


Fig. 3: A scheme of representation of point sets using surrounding circle.



(a) (b)

Fig. 4: (a) Section for contour description.
(b) Convexity and concavity degree.

Using the inflection points, the shape contour is segmented into a set of primitives (line, convex and concave contours) and described by the parameters:

- **Area A_i** is the area enclosed between the chord and the arc between the inflection points Pi and P_{i+1}.
- **Length** l_i is the length of section S_i
- **Type** (line, convex or concave curve).
- **Angle of inclination** α_i (see Fig. 4.a).
- **Degree of concavity or convexity Dg** (see Fig. 4.b).
- **Rotation Angle** β_i is the angle traversed by the tangent to the segment from inflection point Pi to inflection point Pi+1and shows how strongly a section is curved.
- **Length** d_{1i} is the length between the first inflection point Pi and the center O of the section S_i
- **Length** d_{2i} is the length between the second inflection point P_{i+1} and the center O of the section S_i

The convexity or concavity degree ($Dg=d_i / l_i$) of each Section or curve (C) is computed as the ratio of the maximum of distances from points on the curve to associated chord and the distance of the chord of (C) **Fig. 4.(b)**

III. SHAPE DESCRIPTORS USING XML LANGUAGE.

A. Our algorithm to describe the shape contour.

Firstly, the area A_i enclosed between the chord and the arc between the inflection points Pi and P_{i+1} is computed. The same, for each curve of the contour where the two inflection points Pi and Pi+1 are considered and the center O of the surrounding circle is localized and the distances d_{1i} (resp. d_{2i}) between inflection point Pi and the center O (resp. inflection point P_{i+1} and the center O) are computed. Next, the angles α_i **and** β_i are computed with Degree of concavity or convexity Dg. Our algorithm computes these parameters for each section in each Bin and then stores them in a text file.

An iterative process is presented to describe the shape using XML, this process is in the following algorithm:

Algorithm to describe the contour in an XML file:

Begin
Open an XML file F
For j=1 to K // K=N x M
For each Bin B(j)
NP: Compute the number of points in B(j);
L: Compute the number of sections in B(j);
Npi=L+1: // is the number of inflection points
For i=1 to L
S(i) ← current section
If (S(i) is concave) then
Compute the values {A_i, Length l_i, Angle α_i, Angle β_i, Dg_i, Length d_{1i}, Length d_{2i} }
Else If(S(i) is convex) then
Compute the values {A_i, Length l_i, Angle α_i, Angle β_i, Dg_i, Length d_{1i}, Length d_{2i} }
Else // Right Curve
Compute the values {Length l_i, Angle α_i, Length d_{1i}, Length d_{2i} }
End If
Storing values in an XML file (F).
End for(i)
End for(j)
Close the file F // Descriptions of shape is stored in F
End.

From the XML file that contains the encoding geometric outline of the shape, we can draw the outline shape of its description. The use of absolute lengths, angles and the area produces an absolute description of the shape. This coding technique can be used in the description of databases of images such as MPEG-7 database, natural silhouette database, marine database among others. The coding of shapes for their visualization requires a good decomposition of contour

B. How to write an XML descriptors.

In computer vision, the XML syntax is not used in all the descriptions in the literature. In this paper we propose XML language for describing the features of the shape structured in order where each contour is associated one descriptor written in a specific file XML.

To write the shape descriptor, we associate to each Bin (<Bin>) a composed part of the shape, each part may have several sections (concave, convex or right section), these sections are presented by a set of parameters.

We use the syntax of XML to write our outline of the shape as follows:

- - **Bin** : Each Bin is defined by its number, and the Number of its points (NP).
  < Bin Numberofbin="i" >
  - - Number of points in the Bin(i)
  - - Portion
  - - Sections
  </Bin>
- - **Portion** (P): The portion is defined by its sections.
  <P>
  - - Concave Section
  - - Convex Section
  - - Right Section
  </P>
- - **Section (S):** The section is defined by its type and described by the parameters (A_i, l_i, α_i, β_i, Dg_i, d_{1i}, d_{2i})

- Concave/ Convex Section
  - < SCC> or <SCV>
    <A_i> </A_i>
    < l_i > </l_i >
    <alpha_i> </ alpha_i >

<beta_i> </ beta_i >
< Dg_i > </Dg_i >
< d_{1i} > </ d_{1i} >
< d_{2i} > </ d_{2i} >
< /SCC> or <SCV>

- Right Section
  - <SR>
    < l_i > </l_i >
    <alpha_i> </alpha_i>
    < d_{1i} > </ d_{1i} >
    < d_{2i} > </ d_{2i} >
    </SR>

For example, descriptions of the Bin illustrated by figure 5 are writing in XML file as:

<Bin Numberofbin=”i”>
<NP> 31</NP>
<P numberofportion= ’’1’’>
<SCV>
<A> 2.23 </A>
< l > 61.6 </l >
<alpha> 135 </ alpha >
<beta> 90 </ beta >
< Dg > 0.44 </Dg >
< d_1 > 160 </ d_1 >
< d_2 > 132 </ d_2 >
</SCV>
<SR>
< l > 67.5 </l >
<alpha> 180 </alpha>
< d_1> 132 </ d_1>
< d_2 > 136 </ d_2 >
</SR>
<SCC>
<A> 1.92 </A>
< l > 60.5 </l >
<alpha> 110 </ alpha >
<beta> 80 </ beta >
< Dg > 0.23 </Dg >
< d_1 > 136 </d_1>
< d_2> 75.5 </ d_2>
</SCV>
</P>
</Bin>

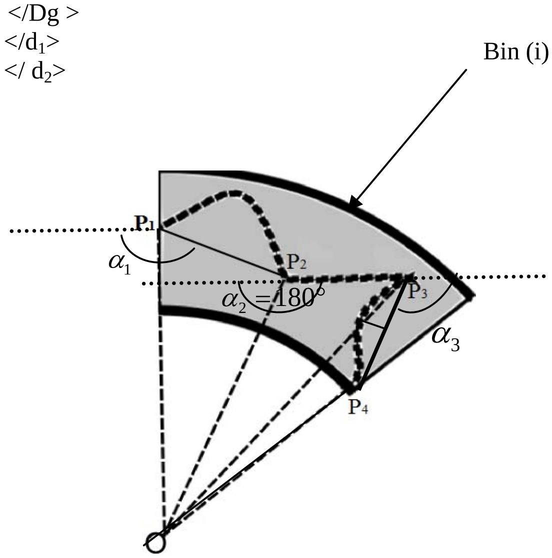


Fig. 5: presentation of Bin number i shown in Figure 3

The Bin illustrated in figure 5 is composed of three sections; the first section is convex and has (135° of inclination , 61.6 pixels as length, 90° as rotation angle, 44.17% of Degree of convexity, 160 pixels as length d1 and 132 pixels for length d2). We compute all parameters of all sections of the contour and then store them in a structured XML file.

C. Storing the shape descriptor in the XML file.

The shape descriptor will be stored in an XML file, the structure of this file must follow the syntax generated by our algorithm. This structure will be generated in this form:

```
<? xml version="1.0" encoding="ISO-8859-1"?>
<SHAPE> <Name> Name of shape </Name>
<Bin Numberofbin ="1 ">
   <P Numberofportion= ''1''>     - - -      </P>
   <P Numberofportion= ''2''>     - - -      </P>
              - - -
</Bin>
              - - -
<Bin Numberofbin ="N ">
   <P Numberofportion= ''1''>     - - -      </P>
              - - -
</Bin>
< /SHAPE>
```

The XML file contains a global description of the information extracted from some shape, so it will be easy to reconstruct the outline of the shape from its XML descriptor.

From the XML descriptor it is easy to extract the following indices:

- Number of Bins: Computed as the number of tag <Bin>.
- The order of Bins.
- Number of Portions: computed as the number of tag <P>.
- Number of Sections: computed as the number of <SCC>, <SCV> and <SR>.

XML file structure can be schematized by a DTD (Document Type Definition) or Schema XSD. A schema shows the element names (XML tags) that can appear in an XML file so we can know the order and names of elements that exist in the XML file. This does make a good correspondence between the shapes descriptors or may be a good recognition of the shape. This will pare the way for our next work leading to the shape recognition.

IV. RESULTS

First, we extract the contour of the shape, (we illustrate in figure 6 an example of shape from MPEG-7 database and its contour presented by inflection points). The inflection points of the contour will be detected using Chetverikov's algorithm [14]. The next step of our program is the computation of different parameters of sections in each Bin, and the descriptor is then stored in text file. The sweep of any shape begins with the first Bin (Bin (1)) near the center O and following a direction of clockwise, where O is the center of the surrounding circle (See Fig.7).

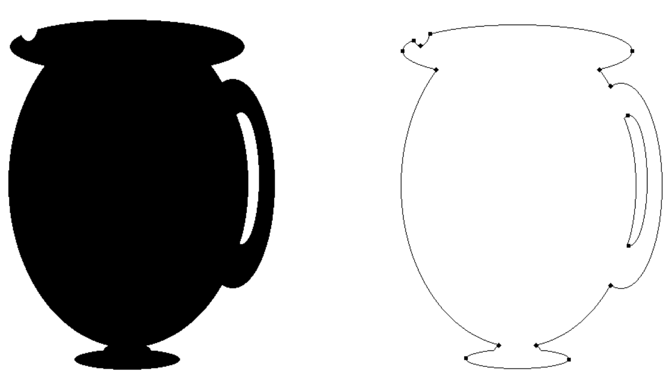


Fig. 6: Simple Shape in MPEG-7 database with detection of inflection points

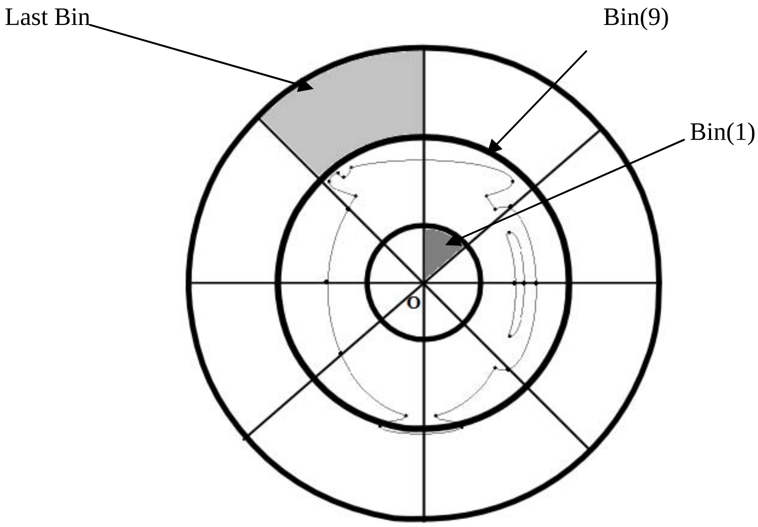
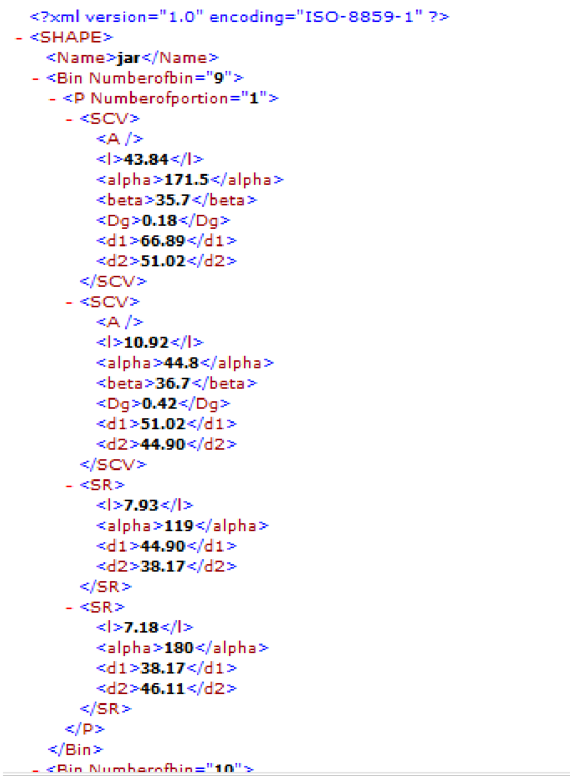


Fig.7: shape descriptor in the surrounding circle

Our algorithm creates an XML file for each shape, for example the XML descriptor computed for the Bin number 9 presented in figure 7 is illustrated by figure 8, there are four sections in the Bin 9 and Three sections in the Bins 10,11,12 and no mapping for Bins 1,2 3,4,5,6,7,8,17,18,19,22,23,24. The Bin number 1 is the starting point of the surrounding circle. (See Fig.7 for the scheme of the shape descriptor in the surrounding circle).



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <SHAPE>
    <Name>jar</Name>
  - <Bin Numberofbin="9">
    - <P Numberofportion="1">
      - <SCV>
          <A />
          <l>43.84</l>
          <alpha>171.5</alpha>
          <beta>35.7</beta>
          <Dg>0.18</Dg>
          <d1>66.89</d1>
          <d2>51.02</d2>
        </SCV>
      - <SCV>
          <A />
          <l>10.92</l>
          <alpha>44.8</alpha>
          <beta>36.7</beta>
          <Dg>0.42</Dg>
          <d1>51.02</d1>
          <d2>44.90</d2>
        </SCV>
      - <SR>
          <l>7.93</l>
          <alpha>119</alpha>
          <d1>44.90</d1>
          <d2>38.17</d2>
        </SR>
      - <SR>
          <l>7.18</l>
          <alpha>180</alpha>
          <d1>38.17</d1>
          <d2>46.11</d2>
        </SR>
      </P>
    </Bin>
  - <Bin Numberofbin="10">
```

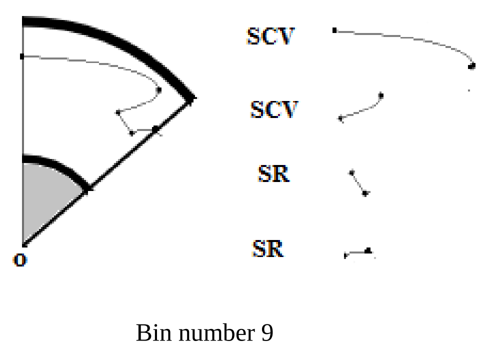



Fig. 8: XML descriptor for a Portion in the bin 9 of the silhouette.

For each shape, one XML descriptor generated by the system, this descriptor can be used to recognize the shape or to make the correspondence between shapes.

V. Conclusion

In this paper, we presented a new method for writing descriptors of outline shapes. This description may be used in Computer vision applications and for the visualization of the coded image in multimedia applications. In addition, using XML format provides the gateway for different systems to perform shape searching with various algorithms. Our algorithm is tested in the databases of MPEG-7, a large number of shapes were used in experiments. The results obtained show that it is advantageous to use the XML descriptor proposed in the recognition problem, because it is easy to extract information about the contours of the shapes in the database and encode them into a text file.

References

[1] Zhang, D., Lu, G.: Review of Shape Representation and Description Techniques. Pattern Recognition. 37, 1–19 (2004)

[2] Cordella, L.P., Vento, M.: Symbol and Shape Recognition. In: Chhabra, A.K., Dori, D. (Eds.) GREC99, LNCS, vol. 1941, pp. 167–182. Springer, Heidelberg (2000)

[3] F. Mokhtarian and A. Mackworth. A theory of multi-scale curvature-based shape representations for planar curves. *PAMI*, 14(8):789–805, 1992.

[4] Miroslaw Bober, “Curvature Scale-Space (CSS) Representation of Shape”, Doc. ISO/IEC JTC1/SC29/WG11 P320, 47th MPEG meeting, Seoul, Mar. 1999.

[5] Whoi-Yul Kim and Yong-Sung Kim, “A Rotation Invariant Geometric Shape Descriptor using Zernike Moment”, Doc. ISO/IEC JTC1/SC29/WG11 P687, 47th MPEG meeting, Seoul, Mar. 1999.

[6] H. Blum, A transformation for extracting new descriptors of shapes, in:W. Wathen-Dunn (Ed.), Models for the Perception of Speech and Visual Form, MIT Press, Cambridge, MA, 1967, pp. 362–380.

[7] K. Arbter, W.E. Snyder, H. Burkhardt, G. Hirzinger, Application of affine-invariant Fourier descriptors to recognition of 3-D objects, IEEE Trans. Pattern Anal. Mach. Intell. 12 (7) (1990) 640–647.

[8] E. Attalla, P. Siy, Robust shape similarity retrieval based on contour segmentation polygonal multiresolution and elastic matching, Pattern Recognition 38 (12) (2005) 2229–2241.

[9] M. H. Yu, C. C. Lim, J. S. Jin, Shape similarity using XML and portal technology, Visual Information Processing VIP , Sydney, Australia 2006

[10] A. Chalechale, A. Mertins and G. Naghdy” *Edge image description using angular radial partitioning* ” IEE Proc.-Vis. *Image* Signal Processing 2004

[11] S. Belongie and J. Malik. "Matching with Shape Contexts". IEEE Workshop on Contentbased Access of Image and Video Libraries (CBAIVL-2000), (2000).

[12] Serge Belongie, Jitendra Malik and Jan Puzicha” Matching Shapes” Eighth IEEE International Conference on Computer Vision (July 2001.

[13] S. Belongie , J. Malik and Jan Puzicha “Shape matching and object recognition using shape context” IEEE transactions on pattern analysis and machine intelligence April 2002.

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