A Novel Star Field Approach for Shape Indexing in CBIR Systems

Oleg Starostenko, Alberto Chávez-Aragón, Gennadiy Burlak, Renán Contreras

***Abstract*—This paper presents a novel hybrid method for content based visual information retrieval (CBIR) that combines shape analysis of objects in image with their automatic indexing by textual descriptions. The principal goal of proposed method is the applying semantic Web approaches for visual information description in systems which use the low-level image characteristics. In the proposed method the user-oriented textual queries are converted to image characteristics which are used for visual information seeking and matching analysis. A decision about similarity between a retrieved image and user queries is taken by computing the shape convergence star field or two-segment turning functions combining them with matching of ontological annotations of objects in image providing in this way the machine-understandable semantics. For analysis of proposed method the image retrieval IRONS (Image Retrieval by Ontological Description of Shapes) system has been designed and evaluated in some specific image-restricted domains**

***Index Terms*— Image retrieval, ontology, semantic web, shapes.**

I. Introduction

A typical approach to automatic indexing and classification of images is based on analysis of the low-level image characteristics, such as color, texture or shape [1], [2], [3] but this type of systems does not provide semantics associated with the content of each image. There are some well-known systems for visual information retrieval (VIR) which may be used as prototypes for a novel approach. One of them is Query by Image Content system (QBIC) provides retrieval of images, graphics and video data from online collections using image features such as color, texture, and shape for computing the similarity between images [4]. AMORE (Advanced Multimedia Oriented Retrieval Engine) and SQUID systems provide image retrieval from the Web using queries formed by keywords specifying similar images, sketches, and SQL predicates [5]. Although the contributions of these systems to field of VIR are important, they do not provide mechanisms to represent the meaning of objects in images.

Manuscript received March, 9, 2007.

O. Starostenko is with the University de las Américas, Cholula, Puebla, 72820, Mexico (phone: 222-229-2654; fax: 222-229-2138; e-mail: oleg.starostenko@udlap.mx).

A. Chávez-Aragón and R. Contreras are with the University de las Américas, Cholula, Puebla, 72820, Mexico (phone: 222-229-2029; fax: 222-229-2138; e-mails: j_chavez_mx@yahoo.com.mx, renan.contrerasgz@ udlap.mx).

G. Burlak is with the Center for Research on Engineering and Applied Sciences, UAEM, Cuernavaca, Mor. Mexico (e-mail: gburlak@uaem.mx).

In order to overcome this problem, we propose to apply the machine-understandable semantics for search, access, and retrieval of multimedia data using ontology [6]. The widely used Grubber's definition permits to describe semantics that establishes a common and shared understanding of a domain and facilitates the implementation of user-oriented vocabulary of terms and their relationship with objects in image [7]. The potential applications of the proposed image retrieval facilities include systems for supporting digital image processing services, high performance exchange of multimedia data in distributed collaborative and learning environments, digital libraries, etc.

II. Proposed Shape Analysis Approach

The proposed method may be described as a combination of specific descriptors based on shape preprocessing for extraction of sub-regions (objects) invariant to scale, rotation, illumination, and application of ontology concepts for definition of machine-understandable semantics for retrieved images.

A. Shape indexing with two-segment turning function

Traditionally, a shape is described as a closed polygon which may be extracted by any well-known method for border estimation. Usually obtained shape may be represented by polygon with a great number of vertices; that require a lot of time for their processing [8], [9]. In order to reduce a quantity of polygon vertices to a subset of vertices containing relevant information about the original outline, the discrete curve evolution technique has been proposed. It is achieved by assigning a relevance measure to each vertex in order to remove the least important vertex. Once a vertex is removed, its neighboring vertices must be connected. This process is repeated until we obtain simplification of desired shape. The relevance measure K is defined as it follows tables:

$$\boldsymbol{K}(s_1, s_2) = \frac{\beta(s_1, s_2) l(s_1) l(s_2)}{2\pi(l(s_1) + l(s_2))} \tag{1}$$

where $\beta(S_1,S_2)$ is the normalized angle in radians between two segments $S_1;S_2$, and $l(S_1)$, $l(S_2)$ are the length functions for segments normalized with respect to the total length of the polygonal curve C. The lowest value of $K(S_1; S_2)$ corresponds to the least contribution to the curve C of arc $S_1 \cup S_2$. The algorithm for curve evolution is presented below.

(Advance online publication: 17 November 2007)

Algorithm 1 Curve evolution algorithm

Input data: a closed polygon P_m, where m and n are the number of segments of input and output polygons, $n < m$

Output data: a closed polygon P_n

1. Find in P_m a pair of segments S_i; S_{i+1} such that $K(S_i; S_{i+1})$ is minimum; $(0 \leq i \leq m-1)$

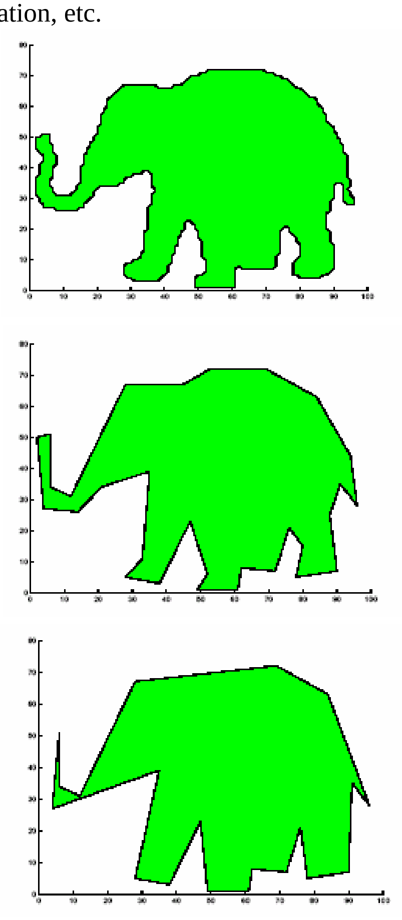
2. Replace two segments S_i; S_{i+1} by one S_j joining initial point of S_i with endpoint of S_{i+1}.

3. $m = m - 1$.

4. Repeat steps from 1 to 4 until m is equal to n.

Fig.1 shows the results of applying this technique that preserves main visual parts of the original polygonal curve and obviously the amount of information to be processed on the step of shape analysis is decreased drastically. The minimum value of *n* is defined empirically by user when this polygon still may be recognized as the particular pattern to be simplified.

However, the polygonal representation of a shape is not a convenient form for calculation how similar is that shape to another. We propose to compute a similarity between shapes using so called two-segment turning function (2STF). Our approach for representing a curve is related to proposal of [10], where calculated from silhouette of object step function is used. This function is called tangent function; however, it has some disadvantages regarding to features such as invariance to scale, rotation, translation, etc.



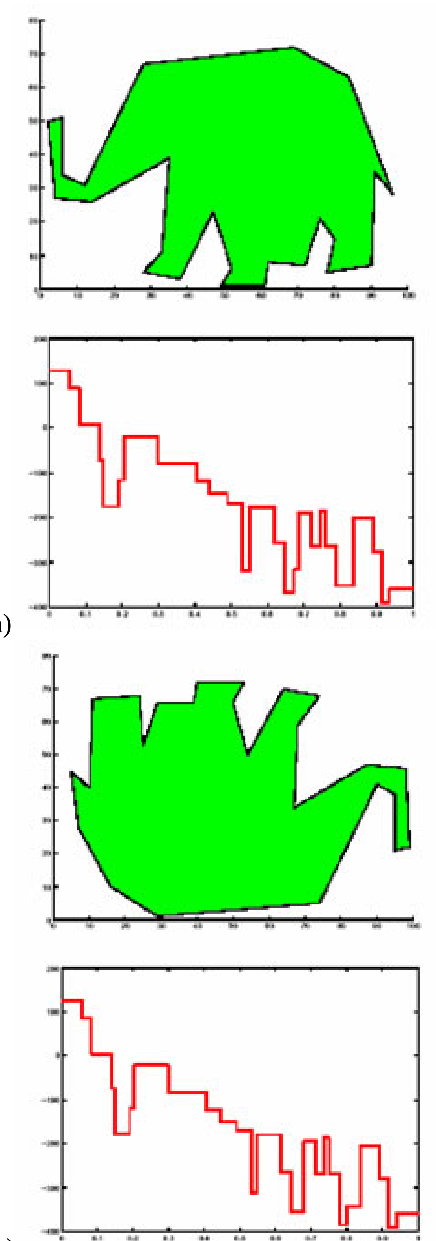
a)

b)

c)

Figure 1. Evolution of the original shape a) to the polygon of 30 segments b), and the polygon of 20 segments c)

2STF solves these problems by a simple strategy which operates with angle between two consecutive segments [11]. Using 2STF a polygonal curve *P* is represented by a step function, the steps on x - axis represents the normalized arc length of each segment in *P* starting from the most left one, and the y - axis represents the angle between plotted segment and next consecutive one in counterclockwise direction. 2SFT has some advantages for shape matching, because this approach is invariant to translation, scaling, reflection, and rotation. 2STF is built taking into account the relative position between consecutive segments. That allows getting the same representation for a set of shapes even though they are placed in different positions or has been scaled or rotated.



a)

b)

Figure 2. The polygon and 2STFs with rotation and scaling

The turning angle obviously is not affected by size of a shape because the angle is the same. Regarding the length of segments, 2STF uses the normalized length with respect to the total perimeter of the shape. The sum of the lengths of the whole curve is equal to 1. As a consequence, this approach is invariant to scaling too. In Fig.2 the computing of 2STF is presented for the same polygonal curve. The polygon on the right has been reflected and scaled by a factor of *0.8.* It is clear that both 2STFs have similar characteristics.

The similarity between two shapes is computed by analysis the differences between obtained 2STFs as it shown in the following algorithm:

Algorithm 2 Matching strategy for comparison of 2STFs

Input: a polygon transposed into curve of 2STF, and 2STF of another reference polygon from data base of prepared forms

Output: compared value

1. Scale two curves to the same length, the scaling factor is $sf = \frac{l(P_1)}{l(P_2)}$ *where* $l(P_i)$ *is the length of curve i and* $l(P_1) > l(P_2)$

2. Curve P_2 *is shifted, equalizing the weighted average of the angle values between* P_1 *and* P_2

3. Compute the area between two 2STF curves called difference D. The difference value D is obtained by multiplying the scaling factor sf by length of the mayor curve

Fig.3 shows the proposed matching strategy. The shaded area represents how similar two shapes are.

The disadvantage of 2STF representation is significant time that it takes to find the best correspondence between two curves which may be reduced by decomposition of a curve *P* into groups *G* in order to obtain the same number of arcs of both curves. This process consists in grouping consecutive largest arcs to form groups of segments covering the whole curve *P.* The idea of joining together largest arcs has the following

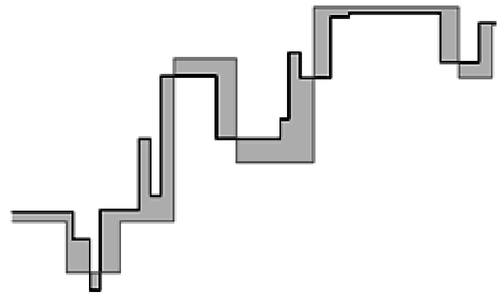


Figure 3. Matching strategy for computing similarity between two polygons.

reasons: curves can have different number of arcs and small arcs can be joined together so they can be compared with a bigger arc of another curve. Formally, a polygonal curve *P* is made up of a set of segments S_i, so we can denote a curve *P* as, $P = (S_1; S_2 ::: S_n)$. Therefore, a curve representing the shape consists of *ns* line segments, where *n* is the number of segments in *P*. The decomposition of the curve into the largest arcs can be obtained in the following way. We denote the largest arc as Ma_i which can be either convex or concave arc. It is a set of segments whose turning angles have the same sense except the first one. The largest arcs are defined as follows. $Ma_i = (S\_start; :::; S\_end)$; $S\_start \geq 1$ and $S\_end \leq n$; as a result, the consecutive segments are fused in one segment Ma_i with initial position at the beginning of *S_start* and final position at the end of *S_end*. The number of group operations for each curve is *NG* = $(2^A)^2$, where A is a number of the largest arcs of the curve. There are some restrictions for grouping either concave or convex arcs which reduce the number of all possible combination to valid ones [12]. Table 1 shows the time that takes the calculation of valid combinations using a personal computer with processor of 2GHz and RAM of 1GB.

Table 1. Time for computing the valid combination of either concave or convex arcs

Maximal arcs	Combinations	Valid Combinations	Time (seconds)
2	16	2	0.031
4	256	18	0.047
6	4096	166	0.078
8	65536	1634	0.375
9	262144	5198	4.375
10	1048576	16646	53.781
11	4194304	53594	581.75
12	16777216	173318	6474.3

Experimentally, we determined that the best correspondence between more than ten-largest-arc polygon takes a lot of time (more than 10 seconds). Therefore, it is possible to conclude that advantage of this approach is independence of shape representation from scale, reflection, translation, and rotation, but it requires a significant time for computing of similarity between shapes. This problem may be solved by another technique called Star Field proposed for representation and indexing of shapes.

B Shapes matching with Star Field

Star Field (SF) is an alternative representation for shapes that allows obtaining more precise comparison because it is not necessary to apply a high grade of evolution of polygonal curves. It means that we able to compare polygons with more than ten arcs (max value acceptable for 2STF). As a result, SF gives easier and faster matching process. Our Star Field method combines approach for computing the similarity among shapes proposed by [13] and its combination with 2STF. Mokhtarian proposed to use the maxima of curvature zero-crossing contours of Curvature Scale Space (CSS) as a feature vector to represent shapes. However, computing CSS is an expensive process and we propose to use 2STF which is easier, faster and more effective [11].

Formally, a SF representation is a set of marks or stars $M_1;M_2:::M_n$, where n is the number of vertices of the polygonal curve that it represents and this number is equivalent to the number of steps in its 2STF. Any star Mi is defined by means of two coordinates ($x;\ y$). The x - coordinates indicate the normalized distance from the starting point to the corresponding vertex, making sure that in the middle of the SF plane there is a star that corresponds to the most important vertex of the polygon. The y- coordinate is the normalized angle between two consecutive segments which share the corresponding point. In other words, y- coordinate of stars correspond to the height of each step in its equivalent 2STF in the range [0,1] where a value of zero represents a $-\pi$ angle and one corresponds to $+\pi$. The principal difference between 2STFs and SF is the grade of evolution of the digital curves they work with. In contrast with the use of a parameter that indicates the number of final vertices of the simplified curve in 2STF, the SF is able to work with a larger number of arcs, consequently with a larger number of vertices without increasing the time for their processing. The curve is simplified until the significance of a vertex has a value more than predefined parameter φ. Thus, the digital curve must be simplified until the least important vertices have been disappeared using threshold φ as it shown in algorithm 3.

Algorithm 3 Curve evolution algorithm with threshold φ

Input: a closed polygon P_m, m is the number of segments; a parameter n, where n is the least number of segments of the output polygon that is acceptable; and a threshold φ;

Output: a closed polygon P_n

1. Find in P_m a pair of segments $S_i;\ S_{i+1}$ such that $K(S_i;\ S_{i+1})$ is minimum $(0 \le i \le m-1)$

2. Replace $S_i;\ S_{i+1}$ by the line S_0 that joins the endpoints of arc $S_1 \cup S_2$.

3. $m = m - 1$.

4. Repeat steps from 1 to 4 until $m > n$ and $|(K(S_i;\ S_{i+1}))| < \varphi$

In the algorithm 3 the threshold φ is used in order to stop the simplification process. Threshold φ is in the interval [$0;\ \frac{1}{4}\pi$]; however, experimentally we have obtained good results using the threshold φ in the interval [$0.4;\ 0.8$]. The SF representation of a curve is made up of a set of points placed on a 2D plane. Each star or point, in the SF represents the vertex that has the same characteristics as its equivalent in 2STF. A particular SF as well as its equivalent 2STF illustration depends on a first point to be drawn. The initial point of the SF diagram is the one with the biggest length. In case of the two biggest segments with the same length the extreme left segment is selected. The way we determine this point is by means of rotating the stars until the star that represents the most important vertex is in the center of the diagram. A SF diagram as 2D plane is divided horizontally into two sections. The upper section holds the stars that represent vertices of concave arcs. The lower part holds vertices of convex arcs. So, a curve can be converted into a star field, and this novel representation can be seen as a cloud of points.

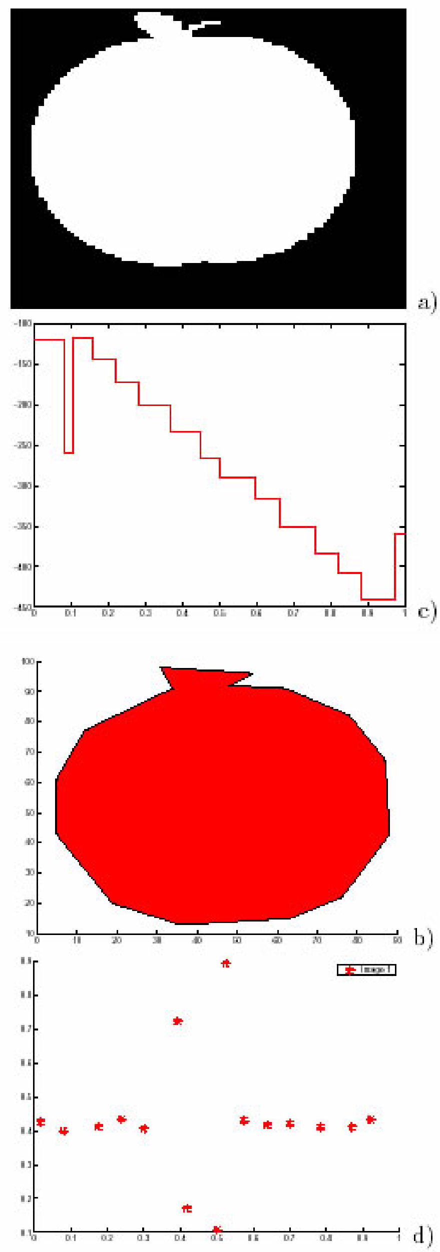


Figure 4. Original image, its 15-segments polygon, and its 2STF and SF representations

In the Fig.4 the original image a) and 15-segments polygon c) obtained from the original image using curve evolution algorithm are presented. The image b) shows 2TSF of polygon and d) shows the star field representation of the same polygon.

So far, a new convenient way for representing a polygonal curve has been presented. For SF a new similarity measure using graph and adjacency matrix will be introduced. Given two polygonal curves P_1 and P_2 and their star field representations SF_1 and SF_2, the graph G that allows us to compute their similarity is defined as $G = (V;E)$ where V and E are finite sets. We call V as the vertex set and E as the edge set of G. Our particular graph G has a set V that consists of two smaller subsets of vertexes υ_1 and υ_2. Set $V = \upsilon_1 \cup \upsilon_2$., where υ_1 is the set of point of SF_1 and υ_2 is the set of points of SF_2. On the other hand, E is the set of pairs $(r; s)$, where $r \in \upsilon_1$ and $s \in \upsilon_2$. Then we propose to use the adjacency matrix for representing the graph, where each cell of that matrix contains the cost for traveling from one column to each row and vice versa. The main idea behind the construction of the matching graph consists in building a connected weighted graph so that an algorithm to find the minimum spanning tree is applied. The minimum spanning tree is a subset of edges that forms a tree of vertexes, where the total weight of all edges in the tree is minimized. Thus, for the more similar shapes we obtain the lower value of corresponding total weight applying very particular mechanism in the searching process as it shown in following algorithm.

Algorithm 4 Matching graph construction

Input: two set of points SF_1 and SF_2 which define two star fields and an increment value Δ

Output: a connected weighted graph

1. Rotate SF_1 and SF_2 so that, the most important star of each SF coincides in the one point

2. For each point sp from the SF_1 do:

3. Look for those points that belong to SF_2, which stay at the most distance d (starting with $d=\Delta$) in all directions from sp, and which have not been connected previously

4. Connect sp with each point found in previous step and assign a weigh equal to the Euclidian distance of two vertices of each edge

5. If there was not any connection, increase d in a value Δ and go to step 3.

6. Select one point of SF_1 and connect the rest of the points from SF_1 with it; finally assign to each edge generated in this step a weigh equal to zero.

The Fig.5 shows a 3D representation of the graph which is the result of applying the Matching graph construction algorithm. All star-like marks are connected single mark with weight equal zero; on the other hand, the star-like marks and the cross-like marks are connected to one with a weight equal to the Euclidian distance between them. For given two identical shapes with the same number of steps of 2TSF the total weight of the spanning tree is equal to zero. This is, because each star is connected with the corresponding one and they have the same value of x- coordinate and y – coordinate.

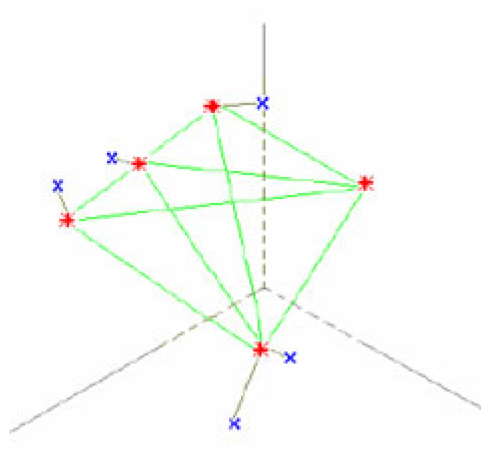


Figure 5 The result of Matching graph construction algorithm

As a result, the values of the path through the spanning tree are equal to zero that means they are identical. The searching the minimum spanning tree may be obtained by applying the Prim's algorithm

Algorithm 5 Prim's algorithm

Input: a connected weighted graph,

Output: a minimum spanning tree

1. Create a tree containing a single vertex chosen arbitrarily from the graph

2. Create a set containing all the edges in the graph

3. Loop until every edge in the set is connected two vertices in the tree

a) Remove from the set the edge with minimum weight that connects a vertex which lies in the tree with a vertex that is not in the tree

b) Add that edge to the tree

Finally, we can define how to calculate the similarity among shapes. The most important part of this calculation is the value of the cumulative weight of the edges that make up the spanning tree. However, the similarity value is also affected by so-called penalty value. It is possible that some stars of the first shape never connected with a star from the second one, a penalty value is added to the final similarity measure. The additional cost is computed as it shown in Fig.6.

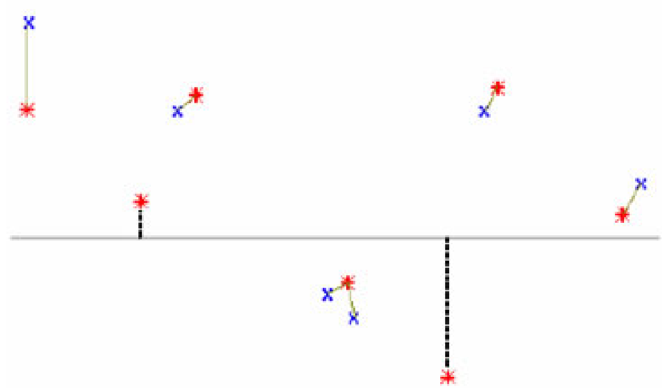


Figure 6. The additional cost defined by distances that are added to the cumulative length

There dotted lines show the distances that are added to the cumulative length obtained from the minimum spanning tree, this is because two stars have not been connected with the corresponding ones.

This new SF approach maintains advantages of 2STF and due to its simplicity allows to computing the complex polygons. SF permits to define a similarity measure based on calculation of a minimum spanning tree from a connected weighted graph which is more fast and accurate.

III. IRONS IMAGE RETRIEVAL SYSTEM

After presentation of two approaches for shape analysis and matching the Image Retrieval by Ontological Description of Shapes (IRONS) system has been implemented. Its block diagram is shown in Fig.7. The input for the system may be an image, its shape, or a keyword, which describes the object with a certain degree of similarity. The retrieved images will be ones with more similarity to the low-level features of a query and will have a high degree of matching with the ontological annotations defining the content of the image. Once the user draws a query, the system uses the SF shape indexing algorithms in order to generate the feature vector for comparison with the other ones in the image database [12]. Then the content-based recognition process is applied to shapes in order to find similar ones in the ontology namespace.

A. Shapes matching with Star Field

The IRONS system consists of four principal modules: query preprocessing, indexing module, feature vector comparison and feedback GUI. The query preprocessing module provides the selection of sub-region containing the relevant objects. Once the sub-region is extracted, the object within that sub-region is found by the CORPAI algorithm.

The Smallest Unvalued Segment Assimilating Nucleus (SUSAN) method [14] for corner detection and Convex Regions Preprocessing Algorithm in Images (CORPAI) [15] have been used for preprocessing of queries by image.

The result of applying SUSAN and CORPAI algorithms is a convex polygon that may be simplified by discrete curve evolution process described early. The algorithm for query preprocessing is show as it follows

Algorithm 6 Query preprocessing algorithm

Input: A color image with luminance of pixels Ic; Output: the feature vector described a shape

1. Ig ⇐ ComputeLuminance (using Ic) // it converts color into gray level image

2. Principal corners ⇐ SUSAN operator (Ig) // detection of object's corners

3. Scs ⇐ SpatialSampling (Ic) // reduction of image size to an 8x8 pixels window4. ColorDescriptor ⇐ ComputeColorDescriptor I1I2I3 (Scs) // descriptors based on I1I2I3 color system model

5. FeaturesVector⇐ ComputeDescriptor (Principal Corners, ColorDescriptor) // the sub-region descriptor includes a color vector and the principal corner's position.

6. Subregion ⇐ CORPAI(Ic,Sp) // applying the CORPAI algorithm over regions.

ConvexHulls (points[ ]) // compute the convex hull {if (query_sub-region(image [][])) // apply boundary detection operator to sub-region (operator(image [][]))}

7. Ic_{NEW} ⇐ TransformationFromSubregionToImage (Subregion) // transformation of the irregular convex sub-region of the original image to a new normalized one8. FeaturesVector ⇐ ComputeDescriptor (Principal Corners, ColorDescriptor, ConvexRegions) // the convex region descriptor is obtained.

9. FeaturesVector ⇐ DiscreteCurveEvolution (Simplified Polygon) // removal of the least important polygon vertexes.

If the query is a keyword, the preprocessing step is not applied.

B. Indexing module

The indexing module generates a feature vector describing shape of objects in image and content-based annotations. The preprocessed polygon is represented by SF using 2STF invariant to scaling, rotation, reflection, and translation. The ontological annotation tool is used for searching matches in the ontology name space. The images with higher matching are retrieved and visualized on GUI with corresponding degree of similarity.

In this work we use hybrid feature vector which defines such low-level image characteristics (shape) and semantic descriptions.

This permits to speed up the matching process as well as reduce the number of iterations with nonsense results. Second vector is formed by ontological description tool which establishes the relationship between the object and its formal explicit definition. In such a way, the meaning of an image may be obtained in textual form as a set of descriptions for each sub-region related to a particular ontology. The Resource Description Framework (RDF) language to support the ontology management has been used in this approach that defines a syntactic convention and a simple data model to implement machine-readable semantics [3]. Using RDF it is possible to describe each Web resource with relations to its object-attributes-value based on metadata standard developed by the World Wide Web Consortium [16]

The ontology is described by a directed acyclic graph; each node has a feature vector that represents the concept associated with that node. Concept inclusion is represented by the IS-A inter-relationship. For example, particular elements of buildings, churches, etc. correspond to specific concepts of shapes defining these buildings, churches. If the query describes an object using this ontology, the system would recover shapes that contain windows, columns, façades, etc. even though, those images have not been labeled as geometric figures for the retrieved object. The feature vectors of each node in the ontology name space consist of keywords linking the previously classified images to the characteristics of the new shape extracted by the SF or 2STF.

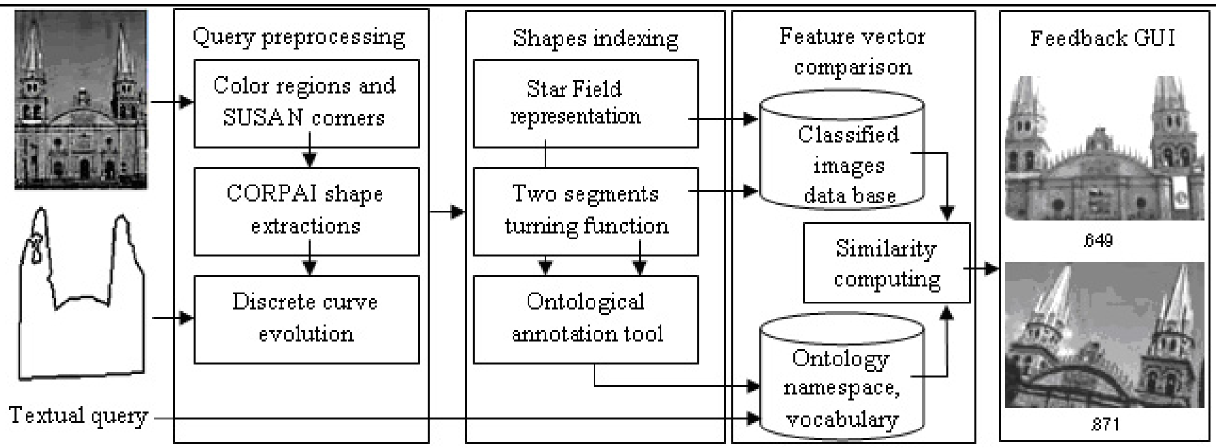


Figure 7. Block diagram of the proposed IRONS system

The indexing and the ontology annotation processes may be described now as:

1. FeaturesVector ⇐ Shape i (Pentagon, Pi,Ci) // Pi is its SF or 2STF representation and Ci is the compactness of the shape computed as a ratio: square root (RegionBorderLength divided by ShapeArea)

2. SaveRelationInOntology(Ic, FeaturesVector of Ic_{NEW}*, Td) // update the ontology namespace*

As it has been mentioned, two kinds of vectors are used for comparison: matching the shapes and definition of similarity in ontological annotations. The computing of similarity is additionally provided by computing the Euclidean distance d_E to compare feature vectors according to the equation:

$$d_E(\mu,\sigma) = \sqrt{\sum(\mu-\sigma)^2} \qquad (2)$$

where μ and σ denote two feature vectors. The query interface of the IRONS system is shown in Fig.8 where the images with high degree of matching are shown in downward order. The user may submit a visual example, a sketch, a keyword or a combination of the above.

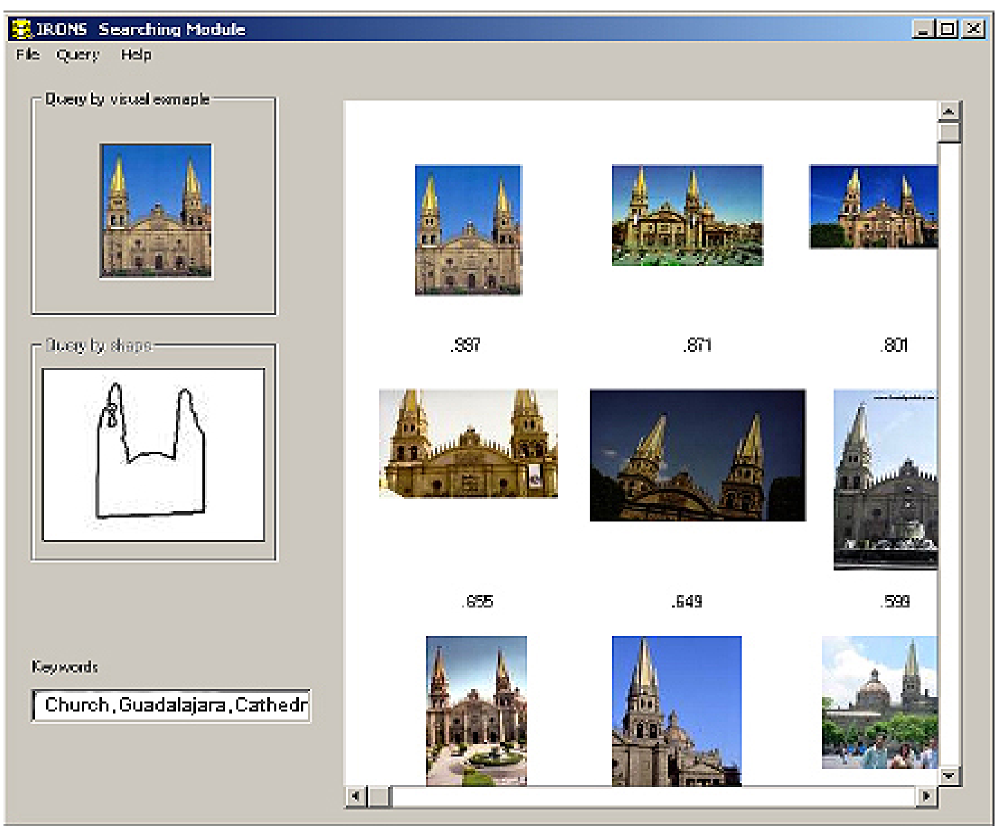


Figure 8. Image retrieval GUI of the IRONS system

IV EXPERIMENTS AND DISCUSSION

We tested the proposed method using the image collection CE-Shape-1. This database has about 1400 images divided into 60 different categories. Each category has about 20 images where there is a single object with well defined contour. This is a restriction of the proposed query preprocessing module which can be improved by using other region selection algorithms. It has been accepted because first of all the hybrid method for image indexing applying semantics approach to analysis of low-level image characteristics must be tested and evaluated. We performed the experiments to verify the role of shape-based and ontology-based indexing in the retrieval process. The performance of the method was evaluated using the precision and recall metrics. The recall means the proportion of relevant images in the entire database that are retrieved for comparison with a query. The precision is proportion of the retrieved images that are relevant to the query from the set of retrieved images. For example, if the IRONS responds with 6 from 10 relevant images in database. In this case the precision is 60%.

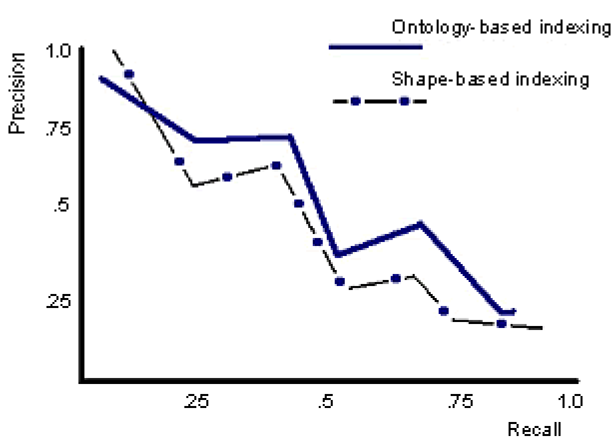


Figure 9. Precision-recall curve for IRONS

The precision of ontology-based indexing was higher as it shown in Fig. 9 where comparison with shape indexing approach has been used. The system performance is better when the image is processed in using ontological descriptions due to the lower number of iterations in the search process.

Some experiments have been done for evaluation of the IRONS average precision in order to detect incrementing the number of retrieved images that are relevant to the query. In this case for each of 20 images within 60 categories the precision without and with ontology-based indexing has been calculated and the average value for each one is presented in Fig. 10 a), b)

Fig. 10 a) shows that with proposed method without ontology the percentage of relevant images is between 0.27 and 0.38. It means that the method is rigorous and does not accept wide range of images as candidates to be retrieval. Fig. 10 b) shows that with ontological descriptions the average percentage is higher for all categories of tested images and amounts to 0.4 - 0.55 confirming that the majority of retrieved images are relevant.

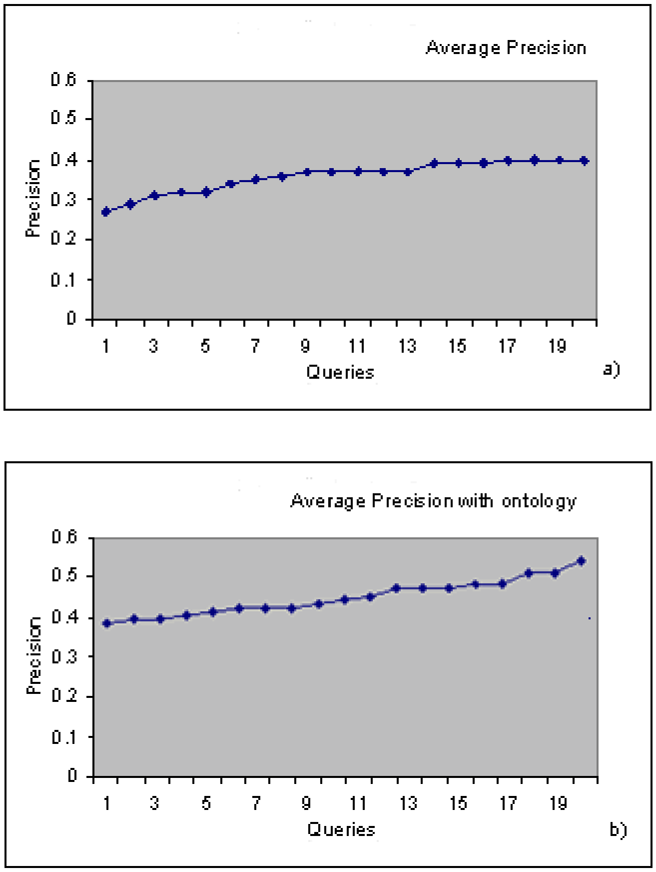


Figure 10. Comparison of average precision for retrieval of images without ontology a) and with ontology b)

V. CONCLUSION

The evaluation of the proposed method and the testing of the implemented system have been done by means of two metrics, precision and recall. In order to use these metrics it necessary to build a reference database which has a set of preprocessed image descriptions. During the comparison of the characteristic vectors of input and preprocessed images the reference database for particular restricted domain is used. The system performance is better when the image is processed in sub-regions; excessive subdivision does not produce good results. Satisfactory retrieval of expected images is achieved faster due to the lower number of iterations in the search process with ontology. The analysis of the indexing approaches shows that SF is in order as fast as 2STF. This occurs because the typical data structures used in indexing tools are hashing tables, which are manipulated with specific keys or signatures representing a shape.

The disadvantages of the system are errors in spatial sampling during generation of the image feature vector as well as the required amount of system memory. Factors like tolerance to occlusion and deformation, robustness against noise, and feasibility of indexing are also considered in our approach.

The most important contribution of this research is the proposed hybrid method combining the advantages of low-level image characteristics extraction with textual description of image semantics. The use of ontological annotations allows simple and fast estimation of the meaning of a sub-region and of the whole image. The proposed image retrieval method is robust to partial occlusion and to small changes in the position of the objects. From the obtained experimental results, we can conclude that the method could be considered as an alternative way for the development of visual information retrieval facilities.

References

[1] T. Gevers, A.W.Smeulders, Combining color and shape invariant features for image retrieval, *IEEE Trans. on Image Processing, 9 (1),* 2000, pp-102-119

[2] J. Chávez, O. Starostenko, Motion estimation algorithms of image processing services for wide community, Baba N, Jain L.C. (Eds.) *Proceedings of Knowledge/Based Intelligent Information Engineering Systems Conference*, September, Japan IOS Press, 2001, pp- 758-763

[3] D. Fensel, The semantic web and its languages, *IEEE Computer Society 15 (6)*, 2000, pp. 67-73

[4] QBIC TM, IBM's Query by image content, 2006, Available: http://wwwqbic.almaden.ibm.com/

[5] Amore, Advance multimedia oriented retrieval engine, 2006 Available:http://www.ariadne.ac.uk/issue9/web-focus/

[6] D. Fensel, *Ontologies: a Silver Bullet for Knowledge Management and Electronic Commerce*, 2-nd Ed. Springer, USA. 2004

[7] T.R. Gruber, A translation approach to portable ontology specifications, *Knowledge Acquisition*, 1993, pp.199-220

[8] O. Starostenko, C. K. Cruz, A. Chávez-Aragon, R. Contreras, A Novel Shape Indexing Method for Automatic Classification of Lepidoptera, *XVII Int. Conf. on Electronics, Communications, and Computers,* Publ. IEEE Computer Society , NJ , USA , February, 2007, Mexico , pp.1-6

[9] M.S. Lew, *Principles of Visual Information Retrieval: Advances in Pattern Recognition*, Springer-Verlag, USA, 2001

[10] . Arkin, , L.P. Chew, An efficiently computable metric for comparing polygonal shapes. *IEEE trans, vol.13,* 1991, pp. 209-206

[11] A. Chávez-Aragón, A Semantic Image Information Retrieval Model, *Ph.D Technical report*, UDLA University, Puebla, Mexico, 2006

[12] O. Starostenko, A. Chávez-Aragón, A Hybrid Approach for Image Retrieval with Ontological Content-Based Indexing, *Lecture Notes, Progress in Pattern recognition,* Springer-Verlag, Germany, vol.3773, 2005, pp.997-1004

[13] F. Mokhtarian, A theory of multiscale, curvature-based shape representation for planar curves, *IEEE Trans. On Pattern Anal. Mach Intell., (14) 8*, 1992, pp.789-805

[14] S.M. Smith, J.M. Brady, A new approach to low-level image processing, *Journal of Computer Vision, (23) 1,* 1997, pp. 45-78

[15] M. A. Medina, A. Chávez-Aragón, O. Starostenko, A Novel hybrid method for image retrieval by ontological descriptions of sub-regions, *Journal WSEAS Transactions on Systems (4) 3,* 2004, pp.1301-1306

[16] D. Beckett, The design and implementation of the Redland RDF application framework, *Proceedings of the 10-th International WWW Conference*, 2001, pp.120-125

Dr. Oleg Starostenko got his BSc and MC degrees at Automation and Telemechanics Department from Lviv State University of Ukraine and Ph.D. degree in Mathematics and Physics from the University Autónoma in Mexico in 1996. Actually, he is Titular Professor at the Computer Science Department. Currently, he participates in several research projects regarding the access, retrieval, and multimedia data processing within distributed environments.

Dr. Alberto Chávez-Aragón obtained the BSc. degree in Computer Engineering from the University Tecnológica de la Mixteca, Mexico, 1998 and the MSc degree in 2001 from the University de las Américas-Puebla in Mexico. Ph.D. degree in Computer Science he obtains from the University de las Americas in 2006. His research interests include visual information retrieval, computer vision and pedagogic robotics.

Dr. Gennadiy N. Burlak got his Ph.D. degree in Physics and Mathematics in 1979 and Dr. of Science in Physics and Mathematics in 1988 from the Kiev University, Ukraine. Actually, he is Research professor in the Center for Research on Engineering and Applied Sciences at Autonomous State University of Morelos, Cuernavaca, Mexico. His current research interests are computer test construction for control and teaching in Client/Server systems and computer simulations.