International Journal of Computer Science and Applications

Vol. 7 No. 1, pp. 45 - 59, 2010

An Object Oriented Graph Approach for Image Representation and Query Based on Content

Eugen Ganea
Software Engineering Department
University of Craiova
Craiova, Romania

Marius Brezovan
Software Engineering Department
University of Craiova
Craiova, Romania brezovan_marius@software.ucv.ro

This paper presents a new method for image representation based on object-oriented graph structure used for formalizing the process of image retrieval based on content considering the information extracted from the segmentation process and the semantic interpretation of this information. We used an image domain ontology for interpreting the image contents and constructed an Object Oriented Graphs (OOG) for describing each image. The instances of the ontology together with the OOG's corresponding to the low level features extracted from the images after segmentation image are stored in an object-oriented database. The object-oriented native query system is used for the retrieval of the images from the database. Our technique, which combines the image low-level descriptors with image domain ontologies, has a good time complexity and the experiments showed that the retrieval can be conducted with good results.

Keywords: object-oriented graph; image segmentation; image representation; domain ontology; object-oriented native query.

1. Introduction

The processing semantics of the images is an open research area, as a support for a series of complex operations in the field of processing images and videos, such as recognizing shapes and objects, understanding video and detect potential risk events in video surveillance, etc.. For semantic retrieval based on image content, it is important to have existing metadata such as an annotation for images. For each domain of application, where image retrieval is desired, semantic relationship needs to be established between the objects (the structure of the working object) present in the images. Our Description Model is based on graph-structured data, implemented in Graph eXchange Language (GXL) [Holt (2000)].For obtaining a successful retrieval of different types of images, there will be combined the low level features extracted from the images (OOG structure) and the high level semantic concepts (ontology structure). We used an image domain ontology based on MPEG-7 shape descriptors and for formulate the semantic query we use the concepts from the ontology; the concepts of ontology are extracted from WordNet lexical resources [Miller (1990)]. In the off-line phase of the retrieval process we designed two stages: a first stage for mapping and storing the domain ontology corresponding to the learning phase and a second stage where the new segmented images are allocate to a concept of ontology. The constructing of the image domain ontology is making with the aid of WordNet plug-in for Protégé [Protege (2009)]. For the translation process from ontology in class hierarchy we used an RDF processor and the generated instances of classes are stored in an object-oriented database (OODB). The implementation of the second step supposes the query of OODB using the features extracted in the segmentation phase. We used the object-oriented native query standard which allows the formalization of queries as class instances. At the end of the learning phase (off-line phase) an OODB for the given domain is populated. In the on-line phase of retrieval process we used the queries expressed in a symbolic form based on the concepts from the ontology, while the target images are found in the database. We implemented a module for transforming the symbolic queries in the object-oriented native queries and for this task the ranked information from the ontology with the synsets extracted from the WordNet lexical resources is very useful. In the following sections there are discussed the next topics: in Subsection 1.2 we describe the method for image segmentation. Section 2 presents the construction of the image domain ontology. Section 3 presents the relationships between the segmented information with their ontology and describes our method for translating the domain ontology into a hierarchy of classes. Section 4 presents the structure of the OODB and the object-oriented native query system used for retrieval. Section 5 describes the results of our experiments and Section 6 concludes the paper.

1.1. *Related Work*

There are many proposed techniques for image retrieval. In this section we consider some of the related work that is most relevant to our approach. Querying visual documents is the subject of many research studies. Two main categories of visual querying approaches can be distinguished in the literature: feature-based querying and semantic querying. Feature-based querying refers to the techniques which focus on low-level visual features (color, texture and shape) such as query-by-example and query-by-sketch. Semantic querying refers to querying more high-level semantics which are closer to users interpretations and the usage contexts. Such descriptions are useful once they can be correctly and easily retrieved through an adapted query language. A new challenge is to provide a query language capable of retrieving visual content described by MPEG-7 and based on the high-level requirements of the users in different application areas. Querying techniques were proposed for image processing, whereas the image content is described by MPEG-7. Two groups resulted: a group that allows the directly using of the XQuery for retrieving the information, and the other group which concentrates on the extraction of semantic information from MPEG-7 documents depending on what the user needs. SVQL, an XQuery adaptation was proposed in order to offer a high-level query language for retrieving visual documents described by MPEG-7 in a TV news production and archiving environment. SVQL (Semantic Views Query Language) is a high level query language which allows different TV news users to express their professional requirements in an abstract and precise way. The project, called "MPEG-7 Audio-Visual Document Indexing System" (MADIS), deals with the indexing and retrieval of video shots and key frames from documentary film archives, based on audio-visual content like face recognition, motion activity, speech recognition and semantic clustering. MADIS is compliant with MPEG-7 standard; it deals both with encoding and search, combines audio, speech and visual modalities and has search capability on the Internet. The ISS (Interactive Shape Selection) Image Database is a state of the art shape and texture based image search engine. Its web based front-end offers a user interface allowing for search by keyword, texture and especially shape, which is given by a hand drawn sketch. The [Gangon (2004)] focus on the use of the technique of the shape recognition module and the associated database access techniques, the main purpose of this paper though are to describe the whole framework of the ISS system. In [Hui, Xu (2009)], the authors describe a MPEG-7 based multimedia retrieval system for a digital museum, called Archaeological Digital Museum of Shandong University (SDU-ADM). For the image retrieval the implemented system uses as descriptors color and shape and simple semantic annotations. This image retrieval system supports the Query-By-Example (QBE) and Query-By-Keyword (QBK) methods. To store the MPEG-7 metadata the proposed system uses an open source native XML database called eXist [eXist (2005)]. This XML database allows index-based XQuery processing, automatic indexing, extensions for full-text search and also integration with the XML development tools. In [Photo Retrieval (2009)] the authors describe a MPEG-7 based Photo Album Management System (MPAMS), which supports semantic photo retrieval using MPEG-7 documents stored in a XML database system. For supporting the search through the photos, the proposed system is designed with an event semantic creation function, which can be divided in two phases. The first phase is the event framework creation and the second one is the event semantic creation. MPAMS system is composed of a photo annotation subsystem, photo retrieval subsystem, and multimedia database. In [Stanescu (2007)] it is approached the content based region query in databases with color images from medical domain with applications in digestive area diseases. Another content based image retrieval system is MIRROR [Relevance feedback (2004)], a MPEG-7 image retrieval refinement based on relevance feedback system, supports both color and texture descriptors and it is based on MPEG-7 Experimentation Mode. It contains three modules: Feature Extraction module, Measure module and Relevance Feedback module. In [Mun Wai Lee (2004)] it is proposed a framework for automatic semantic annotation of visual events. The framework is composed of three parts: the first one is the image parsing engine - is it a stochastic attribute image grammar; the second is the event inference engine - from here it is extracted the descriptive information about visual events. For the semantic representation and also for the grammar-based approach, it is adopted the Video Event Markup Language (VEML). The third component is the text generation engine, where the semantic representation is converted to text description using head-driven phrase structure grammar. In [Tobin (2004)] the authors describe a comprehensive image data management and analysis system, with approaches from three research fields: software agents, geo-referenced data modeling, and content-based image retrieval. The architecture presented in this paper performs automated feature extraction, spatial clustering, and indexing of a large geospatial image library, and it contains three components: the innovative software-agent-driven process, the geo-conformance process for modeling the information for temporal currency and structural consistency to maintain a dynamic data archive, and the third component represented by the image analysis process used for describing and indexing spatial regions from natural and man-made cover types.

1.2. *Segmentation Method*

The low-level system for image segmentation and contours extraction of objects described in this section is used to be integrated in a general framework for storing and semantic retrieval of images in/from the OODB. The segmentation process of an image can be seen as three major steps [Gonzalez (2002)]: preprocessing, feature color extraction and decision. In this subsection we explain only the method used for decision. For image segmentation we use the HSV color space and the following formula for computing the distance between two colors, as proposed in [Smith (1996)]:

$$d_{ij} = 1 - \frac{1}{\sqrt{5 \times (a + b + c)}} \tag{1}$$

where

$$a = (v_i - v_j)^2 \tag{2}$$

$$b = (s_i cosh_i - s_j cosh_j)^2 \tag{3}$$

$$c = (s_i sinh_i - s_j sinh_j)^2 \tag{4}$$

We optimized the running time of segmentation and contours detection algorithms by using a hexagonal structure constructed on the image pixels [Burdescu (2009)] and we used a graph constructed on a hexagonal structure containing half of the image pixels in order to determine the maximum spanning trees for connected component representing salient objects. The hexagonal structure represents a grid-graph, $G = (V; E)$, initially each hexagon h in this structure having a corresponding vertex v from V and the image segmentation is treated as a graph partitioning. The set V of vertices of the graph G is an array of objects of the class $CVertex$. An instance of class $CVertex$ has the following fields associated: $index$, $noComp$, $visited$, $hexagonList$. The field $index$ represents the index of the vertex in the set V from the class $CGraph$; the fields $noComp$ and $visited$ are used in the contraction procedure when is detect a connected components in a graph. The field $hexagonList$ memorize the list of hexagons which correspond of the current vertex. The set E of edges is constructed by connecting pairs of hexagons that are neighbors in a 6-connected sense and it is an array of objects of the class $CEdge$. An instance of class $CEdge$ has the following fields: vi, vj, $wObj$, $origEdge$, $childEdge$. The fields vi and vj are the indices of its two vertices in the set V from the class $CGraph$. The field $wObj$ is an instance of the class $CWeight$ which memorize the weight of the edge. The fields $origEdge$ and $childEdge$ are indices to edges. When a new graph is generate from another graph by contraction procedure, then $origEdge$ represents a reference in the generated graph to the corresponding edge in the initial graph, while $childEdge$ is a reference in the initial graph to the corresponding edge in the generated graph. For building the initial graph G (which is an instance of the class $CGraph$) we cross the hexagonal-grid structure from the left to right and from the top to bottom. Each hexagon corresponding initially to a vertex from V has associated, as attributes, the dominant color, the relative position in the image and its gravity center. For determining these attributes we use eight pixels contained in a hexagon: the six pixels of the frontier, and the two interior pixels. The dominant color of a hexagon is the mean vector color of the all eight colors of its associated pixels. The segmentation is based on maximum spanning tree structure and algorithms and uses the grid-graph for extracting the shapes contained in the image. Each shape is compound by a set of hexagons and has associated six important features: the perimeter, the gravity center, compactness of shape, eccentricity of shape, the list of gravity centers of hexagons from the contour and the syntactic characteristics of the boundary shape. The contours of the extracted visual objects are closed polygons represented by a sequence of the hexagons neighbors. For this list of hexagons on contour we determined the syntactic characteristic which give how there are interconnected three by three the hexagons neighbors from the contour. In Figure 1 there are presented the 5 possible situations. For every possible situation we used for codification a digit from the range $[1 \ldots 5]$. For crossing the boundary, the implemented algorithm considers each hexagon and in this way the length of the syntactic characteristic is equal with the numbers of contour hexagons. In this Figure hc represents the current hexagon, hn represents the hexagon neighbor and r signifies the hexagons belonging to the interior of the current region.

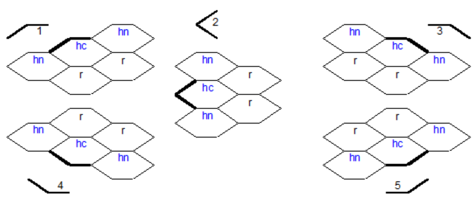

Fig. 1. Possible interconnections between three consecutive hexagons.

We implemented a Java class, $CRegion$, for storing and processing the region attributes. We added a data member to the class for memorizing the unique identification of the image which contains the current shape. It is design other Java class, $CImage$, for memorizing the entire content of the image as a list of objects with type $CRegion$. In the Section 3 we explained how we realized the link between these low level features and the concepts of the image domain ontology.

2. The Domain Ontology Based on Shape Descriptors

In this section, we present the structure and the construction of the domain ontology for describing the image contents in the independent domain. In the first step of the learning phase we consider a set of illustrative images from a domain. These images will be integrated in the ontology based on MPEG-7 features and will result the domain ontology. From the MPEG-7 standard we use only the shape descriptors to describe the low-level features of the image contents. The goal for designing this ontology is to provide a semantic description for MPEG-7 visual descriptors. The segmentation technique, which is presented in the Subsection 1.2, is applied on the set of images used in the learning phase for realizing an interconnection between MPEG-7 shape descriptors and low level features of the images. In order to assign to each image a global concept we consider an associated taxonomy based on WordNet; for transforming the synsets from WordNet in RDF format we used the WordNet plug-in for Protégé. In this phase we designed the domain ontology which will be used also in the query process; the terms used in a query structure corresponds to the entries extracted from WordNet. In the follows subsections we present the MPEG-7 shape descriptors and the structure of the image domain ontology based on WordNet taxonomy.

2.1. *MPEG-7 Contour-Based Shape Descriptors*

The MPEG-7 shape descriptors are invariant to the following affine operations: translation, skewing, scaling and rotation; this property gives a good rate for precision retrieval. The category of shape descriptors are: region-based shape descriptor, contour-based shape descriptor, 3D shape descriptor and 2D-3D shape descriptor. In this paper we use and describe the contour-based shape descriptors which reproduce shape properties of the object contour. The contour-based shape descriptor is based on the Curvature Scale-Space (CSS) representation [Bober (1998)] of the contour. The optimizations of the CSS Descriptor introduced in the MPEG-7 standard are: a) addition of global shape parameters, b) transformation of the feature vector in the parameter space improving retrieval performance, and c) a new quantization scheme supporting a compact representation of the descriptor. We use a hybrid approach with three structural attributes: chain code, polygon approximation, invariants and with three global attributes: perimeter, compactness and eccentricity of shapes. The hierarchy of the classification relating the contour-based shape descriptor is shown in Figure 2:

The value for each specified attributes is determinate by the segmentation module. The chain code is given by the syntactic characteristics of the boundary shape which is computed according to the explanations presented in the Subsection 1.2. The vertices of the polygon approximation of shape are the gravity centers of the hexagons on the contour and as invariant we consider the gravity center of shape. Concerning the global attributes, the perimeter is estimated by the length of the list of gravity centers of hexagons from the contour; for determining the compactness and eccentricity we use the follows formula:

$$compactness = \frac{perimeter^2}{area} \tag{5}$$

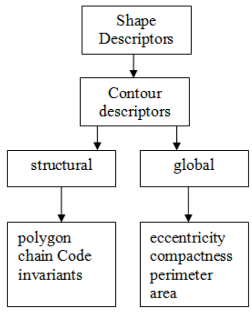

Fig. 2. Classification of the contour-based shape descriptors.

$$eccentricity = \sqrt{\frac{i_{20} + i_{02} + \sqrt{i_{20}^2 + i_{02}^2 - 2 * i_{20} * i_{02} + 4 * i_{11}^2}}{i_{20} + i_{02} - \sqrt{i_{20}^2 + i_{02}^2 - 2 * i_{20} * i_{02} + 4 * i_{11}^2}}} \quad (6)$$

where

$$i_{02} = \sum (y - y_c)^2 \quad (7)$$

$$i_{11} = \sum (x - x_c)(y - y_c) \quad (8)$$

$$i_{20} = \sum (x - x_c)^2 \quad (9)$$

Using the Protégé framework we constructed an ontology based on contour-based shape descriptors extract from the MPEG-7 standard. The format used for saving the ontology is an RDFS schema format based.

2.2. *WordNet Taxonomy for Image Domain Ontology*

The construction of WordNet taxonomy is made with WordNet plug-in for Protégé. We use the noun synsets of WordNet because for image semantic interpretation the nouns can express the entire image content. The hypernym/hyponym relationships among the noun synsets are used for determining the structure of ontology. For a given domain we extract, using Protégé, a list of synsets specifically to keywords (concepts); the keywords from the domain are specified as input. We add as slots for our ontology the six attributes: chain code, polygon approximation, invariants, perimeter, compactness and eccentricity specific of MPEG-7 contour-based shape descriptors. The template of a node from RDF document corresponds to the ontology after these stages is show in Figure 3.

The value type for $chainCode$, $polygonAprox$, $invariants$ and $child - of$ is $Class$ while for other slots the value type is $String$ (for $definition$, $synonym$ and $name$) or $Float$ (for $perimeter$, $compactness$ and $eccentricity$). In the learning phase the training segmented images are allocated to a concept of ontology; the linking of an image to a

```
  <name>wordNet_node</name>
  <type>WordNet</type>
+ <own_slot_value>
  <template_slot>definition</template_slot>
  <template_slot>synonym</template_slot>
  <template_slot>child-of</template_slot>
  <template_slot>name</template_slot>
  <template_slot>chainCode</template_slot>
  <template_slot>polygonAprox</template_slot>
  <template_slot>invariants</template_slot>
  <template_slot>perimeter</template_slot>
  <template_slot>compactness</template_slot>
  <template_slot>eccentricity</template_slot>
```

Fig. 3. The RDF node template.

synset which corresponds to a unique synset number extracted from WordNet, is made by adding a $synset\_id$ attribute at $CImage$ class. In this way for each synset from WordNet considered we attach a list of image and these images are ranked on relationships among the noun synsets (slot $child-of$).

3. Serialization of Domain Ontology into Hierarchy of Classes

The serialization of the domain ontology into hierarchy of classes is made with an RDF processor. The process of translation is realized in two stages which will be presented in the Subsection 3.1. In the Subsection 3.2 we describe how the new segmented images are mapped on the present domain ontology.

3.1. *Translation of the Domain Ontology into Hierarchy of Classes*

The translation process has two phases: the construction of the hierarchy based on the RDFS schema of domain ontology; the extraction of the semantic information from the RDF document and allocation of objects, instances of the classes from hierarchy, with this information. In order to use and process in a unitary way the image information stored in the database, we propose bellow, an example for a simple hierarchy of classes, view Figure 4. As in a pure object-oriented language we propose a class hierarchy with a single and abstract root, called $Object$. The instances of the leaf classes from this hierarchy represent objects, which can be stored into OODB. The four sub-classes of the $Object$ class, $Text$, $Number$, $Character$, and $Image$, represent pre-defined object types used in the information retrieval typically process. Whereas the first three correspond to standard data types, the class $Image$ allows handling the queries concerning the content of images. These classes contain the operations and attribute that aid to the system's functionality.

For the first step of mapping the domain ontology in hierarchy of classes we parse the RDFS schema corresponding to the image domain ontology. We used the Jena framework [Jena (2009)] which enables applications to validate, parse and transform RDF documents. In the second phase of translation we extracted the information from RDF image domain ontology and instantiate the class hierarchy; these instances will be stored in the OODB. The RDF processor creates a class per node of the document, assigning values to attributes from either the data contained within the tag. If the content of the current tag is a reference

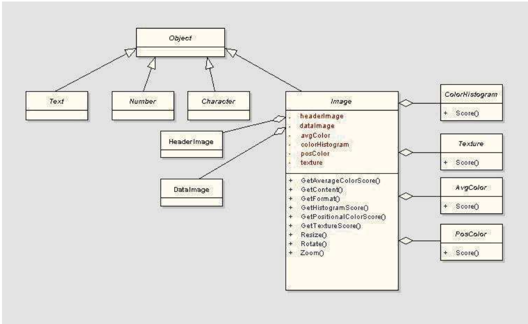

Fig. 4. Simple hierarchy of classes.

to other element, then the processor creates two objects represented the content of the inner tag, respectively the content of the outer tag and establish the relationships between them.

3.2. *The Image Domain Ontology Linkage with Low Level Features*

The relationships between the image domain ontology and the low level attributes extracted in the segmentation process of the images are determinate based on query-by-example of OODB using the relevant images of the domain, which are manually annotated. In this way the image processing ontology is defined. We used the object-oriented native query standard which allows the formalization of queries as class instances. The code template for this type of queries in pseudo-Java language is:

```
final CImage imageQuery =
    new CImage (<<low_level_attributes>>)
List<CImage> list =
    db.query (new Predicate<CImage>()
{
    public boolean match(CImage image)
    {
    return imageQuery.match (image);
    }
});
```

After the execution of one query we have a set of objects which correspond to similar images with the query image. We iterate this set and the synset with the highest frequency given the value of $synset\_id$ attribute for the object corresponding to query image. This mode is used also in the case of query-by-example the OODB is used.

4. Object-Oriented Databases and Native Queries Structure

The advantage of using OODB is the possibility of storing the complex data as image data. On the other hand in the OODB are saved the all low-level features extracted in the segmentation phase (which are used in the retrieval process based on query-by-example) and of the higher-level concepts of ontology assigned in the manual annotation phase. The object-oriented native queries [Cook (2006)] feature is based on the idea that the best language to express a query is the same language that the application is written in. In the next subsections we present how they are stored and how they are indexed the objects in the OODB; we describe also the module which translates the symbolic queries in object-oriented native queries.

4.1. *Storing and Indexing Objects in OODB*

The procedure used for storing a structured object is simple: if we store the composite object then any associated objects will also be stored. The storage process for an object implies the navigation through the object graph starting from that object and storage of all objects that are reached. The instances used when calling the function Insert are only from the classes $CImage$ and $CNode$ attached with a node from the ontology. For the index management of the OODB we build a system of indexes based on the geometric and the semantic attributes of the shapes. The object-oriented database allows creating an index via a specific field or group of fields. The using of indexes improves the query performance, but in the same time the indexes are stored also in the database and the growing of the size which can lead to a decreasing of the storage performance. As a result of our tests we consider two groups of indexes; the first group (geometric group) is used only for the training images in the offline phase of system utilization - learning phase and the second group (semantic group) is used for all other images in the online phase of system utilization - symbolic query phase. At first the group of indexes belongs to the attributes extracted after the image segmentation: the perimeter, the gravity center, compactness of shape, eccentricity of shape, the list of gravity centers of hexagons from the contour and the syntactic characteristics of the boundary shape. This approach drives at a good optimization of the retrieval process for the linking an image to a synset. In this stage the OODB contains only the information corresponding to the ontology so the space taken by the system of indexes has not influence concerning the storage performance. At the end of this phase the first group of indexes is deleted. The code template for the operations of creation/deletion of indexes in pseudo-Java language is:

```
oodb.configure().objectClass(CImage.class).
    objectField(<<attribute_i>>).
    indexed(true)
oodb.configure().objectClass(CImage.class).
    objectField(<<attribute_i>>).
    indexed(false)
```

The second group of indexes is formed by the slots $name$ and $child-of$ from the ontology. Because we create the indexes on a large number of objects we choose to store objects, in the initial stage, without indexes. Then we set the indexes using the template presented above and reopen the OODB again. The experiments showed that this approach allows to obtain a good retrieval time.

4.2. *Translate Symbolic Queries in Object Oriented Native Queries*

The power of the native query is given by the versatility, by the flexibility of the object-oriented languages and by the possibility of using the dynamics queries, which are easily implemented based on the proprieties of the object-oriented languages. In this way the productivity corresponding to the object-oriented programming isn't affected through the utilization of the standard SQL query. The query expressions written in symbolic language must be analyzed and converted to an equivalent native query format. In this process the relationships between the concepts of the ontology on one part and between concepts and classes on the other part are used. The translation supposes two stages: in the first step ir is used the WordNet taxonomy; in the second stage the mapping of concepts on the classes is used. For all the words present in the query expression we search the correspondence with the synsets from the WordNet taxonomy and mark these synsets. In the case when a word hasn't a synset, we use the synonym relation of the taxonomy to retrieve the synset. If it is not found a synset the word it is returned to the user as no relevance for semantic query and it is extracted from expression. After this stage for the list of words from initial query we have a list with synsets. In the second step for each returned synset from the list after the first stage we determine the corresponding class and we make an instance for the class through the call of the constructor which receives the name of the synset. All of these instances and classes are used in the process of matching with the objects stored in the OODB. After the execution a native query is obtained in this mode and we have a list of objects corresponding to the images with semantic content according to the query expression. The name of the file which contains the physique image is formatted based on the unique identified attribute of each attribute. Through the graphical user interface these images are showed to the user and are grouped in clusters according to the input list of synsets.

5. Experiments

A prototype system was designed and implemented in Java, Eclipse Framework, and db4o 7.4 for Java. We tested our system on a Berkeley Segmentation Database dataset (BSDB) [Martin (2001)] and on MPEG7 CE Shape-1 Part B dataset [Leibe (2003)]. The MPEG-7 database consists of 70 classes and 20 shapes per class. The retrieval process implies two categories of experiments: a) the retrieval process based on symbolic language queries, and b) the retrieval process based on query-by-example.

5.1. *Shape Retrieval System*

In the phase of the pre-processing of images we constructed a system containing two modules:

- a segmentation module based on color, which determines super-pixels from the boundary of relevant objects from color images;
- a module which extracts relevant shape features, that can be used for linkage with the image domain ontology.

The relevant information for each determined shape are the syntactic characteristics of the boundary shape; the area, perimeter and center of the gravity of the shape; the circular sequence of the centroid distances and polar coordinates for super-pixels from the contour shape. For experiments on shape retrieval all these features are used. The shape retrieval system uses the following modules:

- the pre-processing subsystem;
- the module for constructing the image domain ontology;
- the translation package from RDF format in object-oriented format;
- the module which serialization the objects data in/from OODB;
- a graphical module which allows users to specify a query in the symbolic language based on the taxonomy extracted form WordNet.

5.2. *Retrieving with symbolic language query*

In this case there were taken into consideration the symbolic queries based on the concepts for the train set of 200 images from BSDB database. In the OODB ”BSDB.dbo” is stored initially the information extracted by the segmentation from the training images. Using the indexes as the perimeter, the gravity center, compactness of shape, eccentricity of shape, the list of gravity centers of hexagons from the contour and the syntactic characteristics of the boundary shape we allocate and store in the BSDB.dbo all images corresponding to the dataset. After the learning and storing phase the OODB is ready to be interrogated. The symbolic query considered is:

$human face$

Using the data from the OOG and from the image ontology we translate this query in equivalent native query:

```
CRegion imageQuery =
    new CRegion (”human face”)
List<CRegion> resultsRegion =
    db.query (new Predicate<CRegion>()
{
    public boolean match(CRegion image)
    {
            return imageQuery.getName().
            Equals (image.getName());
    }
});
```

Figure 5 shows the results for this query applied on our ”BSDB.dbo” databases; we consider only the first 12 retrieval images.

5.3. *Retrieving with query-by-example*

In this case there were taken into consideration the query-by-example based on query image; we used images from 12 classes (MPEG-7 CE Shape-1 Part B) for evaluate the

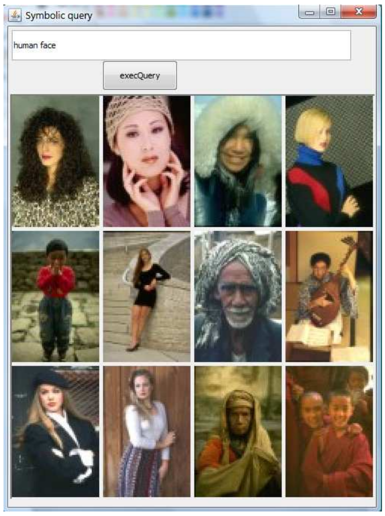

Fig. 5. The results images for the symbolic query.

performance of the shape recognition system based on the retrieval rate.In the OODB ”mpegShape.dbo” is stored initially the information extracted by segmentation from all the images; the stored data are the perimeter, the gravity center, compactness of shape, eccentricity of shape, the list of gravity centers of hexagons from the contour and the syntactic characteristics of the boundary shape. The image considered for retrieval is show in Figure 6.

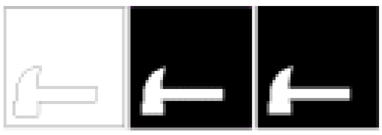

Fig. 6. The image use by the query-by-example. From left to right: contour detection result, segmentation result and the original image.

Figure 7 shows the results for this query applied on our ”mpegShape.dbo” databases; we consider only first 16 retrieval images.

6. Conclusion

In this paper, we propose a method for image retrieval based on content using three key components: (a) a multi-level system for image segmentation and boundary extraction of visual objects based on graph structure and theory; (b) an adaptive visual feature object-oriented representation of image contents; and (c) an image domain ontology for annotation

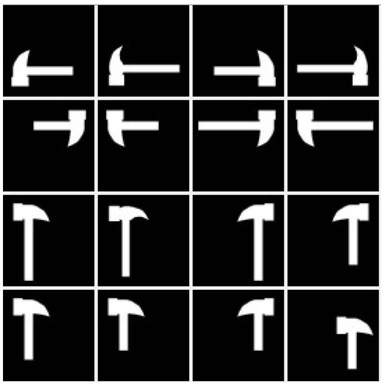

Fig. 7. The results images for the query-by-example.

of the salient objects from images. The proposed techniques allows the creation of an image domain ontology and the population of the OODB with the semantic information from a given domain. Using these two structures the system implements so a query-by-example retrieval process, but especially it implements symbolic language retrieval. The experiments showed that the retrieval process can be conducted with good results regardless of the area that the images come from. The future work implies the description and the using of the attribute grammar system with the goal of searching and retrieving complex images based on the complex query formulated in a symbolic language. Another direction for research will be the description of a lot of ontologies which inherit from each to other and that are organized in an extensible hierarchy of ontologies.

References

Holt, R.C., Winter, A., Schrr, A., *GXL: Towards a Standard Exchange Format*, Proceedings 7th Working Conference on Reverse Engineering, 2000.

Miller, G.A., *Nouns in WordNet: a Lexical Inheritance System*, International Journal of Lexicography, 3,4, pp. 245-264, 1990.

Protégé project, http://protege.stanford.edu/ (consulted 10/03/2009).

Gagnon, L., Foucher, S., Gouaillier, V., Brun, C., Brousseau, J., Boulianne, G., Osterrath, F., Chapdelaine, C., Dutrisac, J., St-Onge, F., Champagne B., Lu X. *MPEG-7 audio-visual indexing test-bed for video retrieval*, Proceedings SPIE Internet Imaging V, 2004.

Xu, H., Xiang, H. *An Image Retrieval System Based on MPEG-7 and XMLDB Query for Digital Museum.*

Open Source Native XML Database Exist, http://exist.sourceforge.net/, 2005.

Ahn, B.T., *Event Semantic Photo Retrieval Management System Based On Mpeg-7.*

Stanescu, L., Burdescu, D.D., Ion, A., Stoica, C, *Content-based region query in databases with color images from digestive area*, Proceedings of Conference on Medical Image Computing and Computer Assisted Intervention, 2007.

MPEG-7 Image Retrieval Refinement based on Relevance feedback, Postgraduate and undergraduate research projects in the Department of Electronic Engineering, City University of Hong Kong, 2004.

Lee, M.W., Hakeem, A., Haering, N., Zhu, S. C. *SAVE: A Framework for Semantic Annotation of Visual Events*,2004.

Tobin, K.W., Bhaduri, B.L., Bright, E.A., Cheriyadat, A. Karnowski, T.P., Palathingal, P.J., Potok, T.E., Price, J.R., *Large-Scale Geospatial Indexing for Image-Based Retrieval and Analysis*,2004.

Gonzalez, R.C., Woods, R.E., *Digital image processing*, 2nd ed., Prentice Hall, 2002.

Smith, J.R., Chang, S.F.,*VisualSEEk: a Fully Automated Content-Based Image Query System*, ACM Multimedia, Boston, MA, November 1996.

Burdescu, D.D., Brezovan, M., Ganea, E., Stanescu L., *A New Method for Segmentation of Images Represented in a HSV Color Space*, Advanced Concepts for Intelligent Vision Systems, Bordeaux, France, 2009.

Bober, M., *Shape descriptor based on Curvature Scale Space*, MPEG-7 proposal P320, Lancaster, UK, February 1998.

Jena A Semantic Web Framework for Java, http://jena.sourceforge.net/, (consulted 15/08/2009).

Cook, W.R., Rosenberger, C., *Native Queries for Persistent Objects, A Design White Paper*, Dr. Dobb's Journal (DDJ), February 2006.

Martin, D., Fowlkes, C., Tal, D., Malik, J., *A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics*, Proceedings of 8th Int'l Conference Computer Vision, 2001.

Leibe, B., Schiele, B., *Analyzing Appearance and Contour Based Methods for Object Categorization*, International Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, June 2003.