Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 695–701
ISBN 978-83-60810-27-9
ISSN 1896-7094

An Hypergraph Object Oriented Model for Image Segmentation and Annotation

Eugen Ganea
Software Engineering Department
University of Craiova
Craiova, Romania Email: ganea_eugen@software.ucv.ro

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

***Abstract*—This paper presents a system for segmentation of images into regions and annotation of these regions for semantic identification of the objects present in the image. The unified method for image segmentation and image annotation uses an hypergraph model constructed on the hexagonal structure. The hypergraph structure is used for representing the initial image, the results of segmentation processus and the annotation information together with the RDF ontology format. Our technique has a time complexity much lower than the methods studied in the specialized literature, and the experimental results on the Berkeley Dataset show that the performance of the method is robust.**

I. Introduction

The hypergraph data model and object-oriented model make join to define the spatial relations between regions from images and the features of them; the composite model is called hypergraph object-oriented model ($HOOM$). Hypergraph data structure inherit the characteristics of the object model. Depending on hypergraph structure the complex spatial relations can be described easily and the attributive data can also be integrated more efficiently. Using hypergraph structures allows more advantages: using a single form for the representation of the combinations of classes and structural inheritance; allowing to avoid data redundancy (inherited values for sub-objects are taken from the super-objects) and to define complex objects (using undirected hyperedges) and functional dependencies (using direct hyperedges); support mechanism for identification through OID's and offers mechanisms for specifying multiple inheritance. In [1] was presented an overview of a hypergraph-based image representation that considered Image Adaptive Neighborhood Hypergraph ($IANH$) model.The proposed segmentation and annotation methods use a virtual graph structure constructed on the image pixels in order to determine the regions from the image and to allocate the labels for these which can give the semantic signature of the each region. Thus the image segmentation is treated as a hypergraph partitioning problem. The predicate for determining the set of nodes of connected components is based on two important features: the color distance and syntactic features [2], that are geometric properties of regions and their spatial configurations. The schema for the prototyping system is presented in Fig. 1:

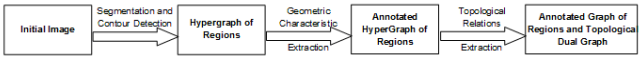

Fig. 1. The image processing system architecture

In the following sections there are discussed the next topics: in Section II we describe the image segmentation based on hypergraph structure; in Section III presents the method of the image annotation; Section IV describes the results of our experiments and Section V concludes the paper.

A. Related Work

In this section we briefly consider some of the related work that is most relevant to our approach. In the image segmentation area, the most graph-based segmentation methods attempt to search a certain structures in the associated edge weighted graph constructed on the image pixels, such as minimum spanning tree [3], or minimum cut [4]. The major concept used in graph-based clustering algorithms is the concept of homogeneity of regions. For color segmentation algorithms, the homogeneity of regions is color-based, and thus the edge weights are based on color distance. For image annotation, an approach based on graph is presented in [5], where the learning model is constructed in a simple manner by exploring the relationship among all images in the feature space and among all annotated keywords. The Nearest Spanning Chain method is proposed to construct the similarity graph that can locally adapt to the complicated data distribution. Object oriented database models [6] are based on the object-oriented techniques and their goal is representing by data as a collection of objects that are organized in hierarchy of classes and have complex values associated with them. The [7] describes the use of the $OODB$ in content-based medical images retrieval and the proposed approach accelerates image retrieval processing by distributing the workload of the image processing methods in the storing time.

II. Image Segmentation Technique Based on Hypergraph Structure

A. The Image Hypergraph Model

The construction of the initial hypergraph is based on an utilization of pixels from the image that are integrated into a network type graph. We used a hexagonal network structure

on the image pixels for representation of the hypergraph $HG = (V, HE)$ and we considered two hyperedges of graph joining the pseudo-gravity centers of the hexagons belongs to hexagonal network as presented in Fig. 2.

The introduction of the hypergraph structure was in [8] and is a generalization of graph theory. The main idea refers to consideration of sets as edges and then a hypergraph is the family of these edges (called hyperedges). This concept represent more general data than graph structure and the theory of hypergraphs has proved to be of a major interest in applications to real-world problems such is image processing. The object model used for storing images, is based on the complex and different structure for each image that does not allow a simple data model using predefined data structures such as those used in relational databases. Relational databases have several limitations in representing an image: from the perspective of data representation model, in the relational database, links between two records are achieved through attributes primary key and foreign key. The records have the same values for foreign keys, primary that are logically related, although they are not physically linked (logical references). The object-oriented model allows to define the methods by which messages are exchanged between objects and to implement the inheritance mechanism which offers classes which have new definitions based on existing definitions.

In an object-oriented database, each object in the real world can be modeled directly as an instance of a class; each instance has an OID and is associated with a simple or complex object. The OID stored in the database is not changed, while other fields associated object can be modified. This identity provides a good support for object sharing updates and simplifies management. Inheritance offered by object-oriented paradigm provides a powerful mechanism for organizing data, it allows to the user to define classes in an incremental way by specializing existing classes. As you can see, is achieved a triangulation of the image, which is in fact a decomposition of the image in a collection of triangles whose edges form the set V of nodes of the hypergraph HG. The condition for achieving a triangulation is satisfied, namely collection of triangles is mutually exclusive (no overlapping triangles) and fully exhaustive (all triangles meeting covers the original image). If it considered the edges which join the gravity pseudo-centers of the hexagons, it obtain a Delaunay triangulation [9]; the grid-graph is a Delaunay graph and based on planarity graph condition ($no\_edge \leq (3 * no\_vertex - 6)$) we demonstrated that the time complexity of segmentation algorithm is $O(nlogn)$. The algorithms for segmentation and the demonstration for complexity are presented in [10]. In the hexagonal structure, for each hexagon h in this structure there exist 6-hexagons that are neighbors in a 6-connected sense and the determination of indexes for 6-hexagons neighbors having as input the index of current hexagon is very simple. The main advantage when using hexagons instead of pixels as elementary piece of information is the reduction of the time complexity of the algorithms. The list of hexagons is stored such as a vector of integers $1 \ldots N$, where N, the number of hexagons, is determined based on the formula:

$$N = \frac{H-1}{2} \times \left(\frac{W - (W\%4)}{4} + \frac{W - (W\%4) - 4}{4}\right) \quad (1)$$

where H represents the height of the image and W represents the width of the image.

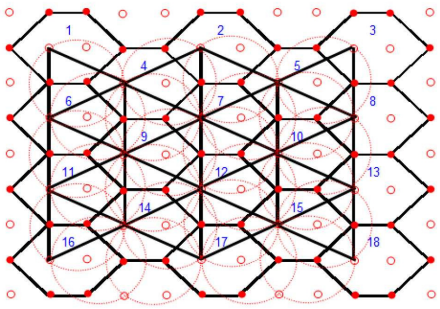

Fig. 2. The hexagonal structure on the image pixels

Each hexagon from the set of hexagons has associated two important attributes representing its dominant color and its pseudo-gravity center. For determining these attributes we use eight pixels: the six pixels of the hexagon frontier, and two interior pixels of the hexagon. For RGB space, the dominant color of a hexagon is determined as follows: is extracted three components (r, g, b) for each pixel of the hexagon; next step is sorting of the three vector components; last phase involves choosing as dominant components of media components located in positions 3 and 4 in the vectors, and dominant color calculation using the formula:

hexagon_color =

$$(r\_d\_color << 16)|(g\_d\_color << 8)|b\_d\_color \quad (2)$$

where r_d_color is the dominant color for the red component, g_d_color is the dominant color for the green component and b_d_color is the dominant color for the blue component of the RGB colorspace. We split the pixels of image into two sets, a set of pixels which represent the vertices of hexagons and a set of complementary pixels; the two lists will be used as inputs for the algorithm which construct the initial hypergraph of the initial image. The mapping of pixels network on the hexagons network is immediately and it is not time consuming in accordance with the following formula which determines the index for the first vertex of hexagon:

$$fv = 2 \times h + \frac{2 \times (h-1)}{columnNb - 1} \quad (3)$$

where fv represent the index of the first vertex, h the index of the hexagon and $columnNb$ represent the column number of the hexagon network. For representing the output of the image segmentation process we used the Attributed Relational Graph (ARG) [11]. The result of segmentation algorithm is stored as a graph where the nodes represent the regions and the edges represent the neighborhood relations: G = (V_r, E_r), where V_r is the set of vertices corresponding regions detected

and E_r is the set of edges that describes the neighborhood relations. The spatial relations between regions are divided into 3 categories: distance relations, direction relations and topological relations. For determining these types of relations we choose for each region the following relevant geometric features: the pseudo-center of gravity; the distance between two neighboring regions; the length of common boundary of two regions and the angle which is formed by two regions.

B. The Image Segmentation Methods

In this subsection there is shown the algorithms for determining the initial hypergraph and the segmentation method. By using as unit element a hexagon, any two neighboring elements have in common two vertexes (an edge). The function $createHG$ which is describes in algorithm 1 produce the initial hypergraph for the initial image.

Algorithm 1: Create the initial hypergraph

Input: The list of color pixels from the hexagonal network: $L = \{p_1, \ldots, p_{6n}\}$
Output: The initial hypergraph HG which correspond to the initial image

```
1  Procedure createHG(L ; HG);
2  * init crtHyperEdge;
3  for i ← 1 to sizeof(L) do
4      hcrt ← L(i);
5      * init indexN as indexes from L for the neighbors
       of hcrt;
6      indexM ← −1;
7      for k ← 1 to 6 do
8          distRef ← colorDist (L[indexN[k]], L[hcrt]);
9          if distRef are minimum value then
10             indexM ← k;
11         end
12     end
13     if !mark(edge[L[indexN[indexM]], L[hcrt]]) then
14         * add (crtHyperEdge,
           edge[L[indexN[indexM]], L[hcrt]];
15         * mark edge[L[indexN[indexM]], L[hcrt]];
16     end
17     else
18         * add crtHyperEdge to HG;
19     end
20 end
```

The output of segmentation algorithm 2 is the hypergraph corresponding to the segmentation image and it is computed based on the hexagonal grid-graph and the distance between the two colors for the RGB color space. The variable $colorsHexagon$ represents the vector colors of the hexagon and the others attributes of the class $Hexagon$ are $indexHexagon$ in the hexagonal structure of the virtual graph and $visitedHexagon$ which is used in the crossing network algorithms.

Algorithm 2: Determination of the segmentation hypergraph

Input: The total number of hexagons from hexagonal grid n
The hypergraph representation for an image (HG), obtained with algorithm 1
Output: The hypergraph with the hexagons of regions $HG = \{\{he_1\}, \ldots, \{he_k\}\}, he_i = \{color, \{r_1, \ldots, r_p\}\}$

```
1  Procedure HGSegmentation (N, HG; HG);
2  * initialize the stack of hyperdges;
3  * get indexH as index of unmarked hexagon
4  while (indexH != -1) do
5      * mark as visited the hexagon indexH;
6      * push (hyperdge[indexH]);
7      HERegionItem ← newHyperEdgeRegion;
8      while !empty(stack) do
9          crtColorHyperedge ← pop();
10         for each crtColorHexagon from the set of
           crtColorHyperedge do
11             if (!mark(crtColorHexagon) then
12                 * add crtColorHexagon to
                   HERegionItem;
13                 * push (hyperdge[crtColorHexagon]);
14                 * mark as visited the hexagon
                   crtColorHexagon;
15             end
16         end
17     end
18     * add object HERegionItem to list HG;
19     * get indexH as index of unmarked hexagon
20 end
```

The procedure $HGSegmentation$ returns the hypergraph with the hexagons of regions; the elements of hypergraph (the hyperedges) are determined for each distinct color from the input image as a list of regions that contain hexagons which have the same dominant color. The $HERegionItem$ is an instance of the class $HyperEdgeRegion$ and represents the data structure corresponding to an item from the output list. The attributes of the class $HyperEdgeRegion$ are: the color of the region and the list of hyperedges which represent the hexagons with the same color.

III. Image Annotation Technique Based on Hypergraph Structure

The management of ontologies, used for annotate of images, has two hierarchical levels that are closely associated. On the one hand, the low-level image contains specific properties as color, texture, shape, and the second level, which contains sematic of image that can be perceived human user. An ontology management system should model the low level that supports retrieval and inference level of their content. One scenario of using such a system can be that a user loads all

ontologies in a given area, and allows selection of different objects in images and their correlation with the concepts of ontologies.

A. Ontologies and RDF format

For specifying the ontologies and the corresponding graph structure of the images segmented and annotated format we used RDF (Resource Description Framework). The RDF is a specification defined $metadata$ processing, providing interoperability between different applications such as an exchange of information, the purpose of understanding the semantics. To use the method of reasoning described in the ontology-based knowledge bases, we used $RDF2Jess$ model, a hybrid model that can be used to fill the gap between RDF and $Jess$. Based on domain knowledge and using Protégé [12] ontology editor, this method turns the RDF format in $Jess$ facts using XSL transformations on XML syntax and additional rules in Jess. For these rules redefined based on RDF semantics $Jess$ inference system is used to implement the reasoning. Predefined rules are used to check consistency and to determine the characteristics of RDF vocabulary. Deducted $Jess$ assertions are helpful for phase domain ontology modeling to assess and refine ontology. According to different levels of expressiveness, $RDF2Jess$ could be extended to new $SWRL2Jess$ where $SWRL$ extends the set of axioms to include Horn rules. The conversion of syntactic and semantic RDF in $Jess$, allows replacement of ontology in RDF format using $Jess$ reasoning engine. Ontology conversion into Jess facts and rules is done in four steps:

- first step is the ontology construction, ontology editor used $Protege$, provides a plug-in RDF ontology to support development. The taxonomy of knowledges into classes, features, restrictions was accepted by experts in the field and by software developers, because this paradigm is very similar to object-oriented modeling (UML). Lately, most ontologies have been formalized using standardized $RDF(S)$ and can be reused to extend the rules, if necessary;
- the second step is to represent the transformation of RDF syntax in $Jess$ syntax using $XSLT$, the output file consists of $Jess$ facts. If support ontology language semantics is specified as $Jess$ rules, matching specific keywords is no longer necessary in the transformation;
- the third step is to combine file $Jess$, including $XSLT$ transformation result, and RDF predefined rules. Moreover, external queries and $Jess$ rules can also be added to the composition similar properties;
- the last step is running the system of rules $Jess$ inference system. The rules defined, is the classification and consistency checking characteristics. The output containing erroneous messages indicating the presence of incorrect syntax in RDF ontology processed.

To implement the RDF, semantics are defined to represent the additional facts and their relations with $RDFS$ primitives such as $rdfs : Class$. This approach prevents duplication of semantic information in the database, and an overview of how to store links between visual concepts/concepts domain and related regions is given by Fig. 3.

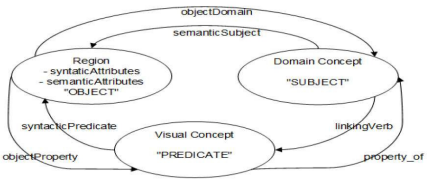

Fig. 3. Triple link: visual concept/semantic concept/Region

In Fig. 4 are presented the classes which are used for specification of the $HOOM$ in the image annotation processus:

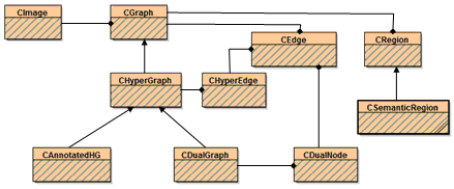

Fig. 4. The hypergraph object data model

B. The Image Query Specification

To specify queries we have expanded the query language $GOQL$ using data structures of type hypergraph - $HGOQL$ (Object Query Language Hyper-Graph). In [13] is proposed $GROOVY$ model (Graphically Represented Object-Oriented Data Model with Values) which represents a proposal to formalize the object-oriented model based on hypergraph structures type. It defines a set of data structures for object model data: (I) Value diagram that defines the attributes that contain a class of objects. Items can be atomic or multi-value, (ii) The court: defines a lot of items under the diagram. An object is a pair $O = < id, v >$, where id is the identifier of the object and v is the value of the object (his properties), (III) Dependency of functional values: are used in the scheme to check that the value of a set of attributes determine in a unique way the value of another attribute; (IV) Object diagram: is a triplet $< N, F, S >$, N - diagram value, F a lot of dependency of functional values for N and S - a lot of subsets of N. This structure is defined using hypergraphs by establishing a corresponding one-to-one between object diagram $< N, F, S >$ and an interpretation where N is the oriented nodes of a hypergraph, F is its direct hyperedges, and S, undirect hypeedges. A manipulation language of hypergraph (HML - Hypergraph Manipulation Language) for query and update them has with two operators querying based on identifier or value, eight operators for inserting and deleting the hypernodes and the hyperedges. During the developed language ($HGOQL$) we implemented only operators who refers to the query. The grouping of the indexes attributes (which includes

enable/disable attributes through facet $index - propagation$) is achieved by Algorithm 3 based on a function implemented in the states of each class. The function $hyperGraphGroup$

Algorithm 3: Grouping the attributes index

Input: The current instance
Output: The index of current instance
Procedure groupIndexObject ($thisObject$;$indexThis$);
$attributeSet \leftarrow atributesOf$ ($thisObject$);
initialize $attributeIndexSet$;
for $attribute \leftarrow attributeSet$ **do**
  if $attribute.pattern\text{-}match\text{-}facet\ is\ reactive$ **then**
    add $attribute$ to $attributeIndexSet$;
  end
end
for $attribute \leftarrow attributeIndexSet$ **do**
  if $attribute.index\text{-}propagation\text{-}facet\ is\ non - inherit$ **then**
    remove $attribute$ to $attributeIndexSet$;
  end
end
initialize $indexThis$;
for $attribute \leftarrow attributeIndexSet$ **do**
  $indexThis \leftarrow hyperGraphGroup(indexThis, attribute)$;
end

Algorithm 4: Function hyperGraphGroup

Input: The index of current instance, the current attribute
Function hyperGraphGroup ($indexThis$, $attribute$);
initialize $currentHEdge$;
if *isSyntactic* ($attribute$) **then**
  $currentHEdge \leftarrow$ addToSyntacticHyperEdge ($attribute$)
end
else
  $currentHEdge \leftarrow$ addToSemanticHyperEdge ($attribute$)
end
$hyperEdgeSet \leftarrow hyperEdges$ ($indexThis$) U $currentHEdge$;
clear $indexThis$;
for $hyperEdge \leftarrow hyperEdgeSet$ **do**
  if *isNonTopological* ($hyperEdge$) **then**
    add $hyperEdges$ to $indexThis$
  end
  else
    add $hyperEdge$ to $indexThis$
  end
end
return $indexThis$

that does clustering indexes, used as support for storing and linking indexes a $hypergraph$ [14], implemented by the $CHypergraph$ class, subject to the previous result that output algorithm ($indexThis$) is an instance of $CHypergraph$. Choosing $hypergraph$ type structure to represent indexes was made because this type of structure is very good for browsing and retrieving images corresponding graph processed. The function $HyperGraphGroup$ properly algorithm is presented in Algorithm 4.

IV. Experiments

In this section experimental results are highlighted demonstrating that the method presented produces a good image segmentation and annotation, and extraction of outlines for visual objects from different images without the need for parametrization of the method depending on image processing. For this purpose, we used human segmentation for color image from Berkeley Segmentation dataset ($BSDB$).

A. Image Segmentation Experiments

After problem examination of validating the quality of segmentation, we proposed an effective evaluation system which shows the comparative results of segmentation both obtained with the algorithm implemented and the three alternatives segmentation methods: the "Mean-Shift" method (MS) [15], the "Local-Variation" method (LV) [3] and the "Normalized-Cuts" method (NC) [4]. In the last part is presented an analysis performed for a set of synthetic images. The $BSDB$ database contains two sets of images: a lot of training with 200 images and a lot of test 100 images. For each image is available a set of human segmentations. This set contains between four and eight segmentations specified in the form of images labeled using a special format. In [16] showed that human segmentation, although vary in detail, are according with each other for the segmented regions by humans are exposed to a finer level of detail and in this situation can merged so as to appear as to extracted regions a coarse level of detail. To assess performance of segmentation method were segmented images of crowds test and were used two measures of quality: the value function harmonic mediation ($F - measure$) and performance value ($P - measure$) [17]. Value of $F - measure$ [18] determined by a combination of precision values (P) and re-call (R) is calculated for each segmentation and number the correct segmentation comparing to the wrong. In Fig. 5 are the results of contour detection for a group images, considering the four methods of segmentation. In the three methods chosen for comparison was considered when the result value is the measure of $F - measure$ is the maximum, namely: MS method, DW = 10 and PW = 8, for the LV method, the value used was k = 900, for NC method, the value used was $nTresh$ = 25. In the second part of experiments was considered a set of synthetic images generated so the RGB colorspace to be presented with non-null value a single channel of the three. In figure Fig. 6 are examples of such images generated. The purpose of generate and use this set of images was to determine the optimal formula for calculating dominant color of a hexagon (formula

Fig. 5. Results for image segmentation: human segmentation, SOD segmentation, MS segmentation, LV segmentation and NC segmentation

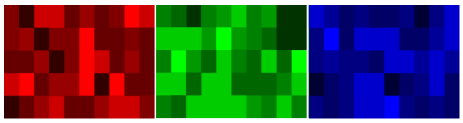

Fig. 6. Samples of synthetics images

2) and on the other side validation formulas used to determine threshold values used in the two stages of segmentation. To achieve these goals we generated for each image, an attached file in XML format. File picture represented in XML as a list of rectangles, for each rectangle specified coordinates of points on the left/top, that right/down, channel number and the value for that channel with values between [0 ...255]. After segmentation a synthetic image is comparing the list of regions obtained with the rectangles list extracted from the XML file by using a parser. As a result of comparison the aim is to modify channel values after considering the dominant color from the level of a hexagon. The closest values were obtained for two experiments that have used the formula 2 for calculation of hexagon color. threshold values used in the two stages of segmentation.

B. Image Annotation Experiments

The following phase after the segmentation step involves the addition of labels to the semantic regions. There are two phases of this stage, manual annotation (training phase) and automatic annotation. For the first phase, representative images are selected for the considered domain. Manual annotation scenario assumes that the first step, loading the domain ontology. In the next step, the user selects the regions of the segmented image and put them in correspondence with the ontology domain concepts. While the regions are manually annotated, an XML file with the annotated information is generated. To identify a region in space of the image is stored in the XML file the hexagons list region which are founded on its border. This information is necessary to establish a visual correlation between the XML node for a region and this; this is done by double selection: when selecting a region already annotated in the tree view of XML corresponding node is selected automatically. Fig. 7 presents the corresponding GUI of the annotation phase.

Based on syntactic and semantic properties of manually annotated images is constructed a decision tree which is translated into a set of rules used for automatic annotation of image regions segmented test. We used to add semantic tags the decoration technique of the hypernodes of the hypergraph in accordance with the RDF format representation. The obtained hypergraph as a result of the processing steps is serialized as a system of objects, which have links in accordance with existing relationships (inheritance or composition) on the all classes which are implemented.

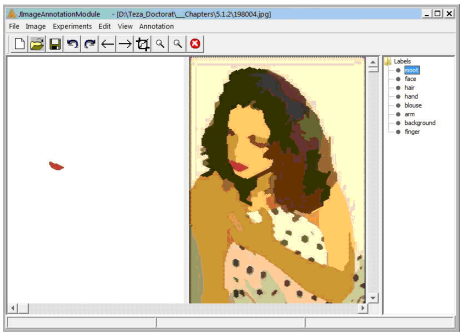

Fig. 7. Graphical user interface subsystem for image annotation

V. Conclusion

In this paper presents an hypergraph object-oriented model for image segmentation and annotation when is considered as the input the information extracted from the image. The unified method for image segmentation and image annotation uses an hypergraph model constructed on the hexagonal structure. The hypergraph structure is used for representing the initial image, the results of segmentation processus and the annotation information together with the RDF ontology format. The hypergraph representation of images is the output of all the phases of system: the segmentation phase and the annotation phase. Our technique, which combines the hypergraph model with the object oriented model, has a good time complexity and the experimental results showed that the method can be yielded with good results regardless of the area of the images that come.The future work implies the using of the hypergraph theory with the goal of searching and retrieving complex images based on the complex query formulated in a symbolic language.

Acknowledgment

The support of the The National University Research Council under Grant CNCSIS IDEI 535 is gratefully acknowledged.

References

[1] Bretto, A. and Gillibert, L., *Hypergraph Based Image Representation*, Graph Based Representations in Pattern Recognition, 2005.
[2] Bennstrom, C.F. and Casas, J.R., *Binary-partition-tree creation using a quasi-inclusion criterion*, Proc. of the Eighth International Conference on Information Visualization, 2004.
[3] Felzenszwalb, P.F. and Huttenlocher, W.D., *Efficient Graph-Based Image Segmentation*, International Journal of Computer Vision, 2004.
[4] Shi, J. and Malik, J., *Normalized cuts and image segmentation*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, 2000.

[5] J. Liu, M. Li, W.-Y. Ma, Q. Liu, H .Lu, *An adaptive graph model for automatic image annotation*, Multimedia Information Retrieval, 61–70, 2006.

[6] W. Kim, *Object-Oriented Databases: Definition and Research Directions*, IEEE Transactions on Knowledge and Data Engineering, 2(3):327–341, 1990.

[7] C.Jr. Traina, A.J.M. Traina, R.A. Ribeiro, E.Y. Senzako *Content-based Medical Images Retrieval in Object Oriented Database*, Proceedings of 10th IEEE Symposium on Computer-Based Medical System - Part II, 67–72, 1997.

[8] Berge, C., *Hypergraphs*, North-Holland Mathematical Library, 1989.

[9] Delaunay, B., *Sur la sphére vide*, Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk, 7:793–800, 1934.

[10] 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.

[11] Hong, P. and Huang, T. S., *Spatial pattern discovery by learning a probabilistic parametric relational graphs*, Discrete Applied Mathematics, 139:113–135, 2004.

[12] *Protégé project*, http://protege.stanford.edu/ (consulted 17/06/2009).

[13] Levene, M. and Poulovassilis A., *An Object-Oriented Data Model Formalised Through Hypergraphs*, Data and Knowledge Engineering (DKE), 1991.

[14] B. Goertzel *Patterns, Hypergraphs and Embodied General Intelligence*, IJCNN, Neural Networks International Joint Conference on, 451–458, 2006.

[15] Comaniciu, D. and Meer, P., *Robust analysis of feature spaces: Color image segmentation*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1997.

[16] Martin, D., Fowlkes, C., Tal, D., and Malik, J., *A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics*, IEEE International Conference on Computer Vision, 2001.

[17] Grigorescu, C., Petkov, N., and Westenberg, M., *Nonclassical receptive field inhibition*, IEEE Transactions on Image Processing, 2003.

[18] Fowlkes, C., Martin, D., d anMalik, J., *Learning affinity functions for image segmentation: combining patch-based and gradient-based approaches*, IEEE Conference on Computer Vision and Pattern Recognition, 2003.