New Method to Detect Salient Objects in Image Segmentation using Hypergraph Structure

Eugen GANEA, Dumitru Dan BURDESCU, Marius BREZOVAN
University of Craiova, 200440, Romania
eugen.ganea@software.ucv.ro

Abstract—**This paper presents a method for detection of salient objects from images. The proposed algorithms for image segmentation and objects detection use a hexagonal representation of the image pixels and a hypergraph structure to process this hierarchal structure. The main goal of the method is to obtain salient regions, which may be associated with semantic labels. The designed algorithms use color characteristic and syntactic features for image segmentation. The object-oriented model used for storing the results of the segmentation and detection allows directly annotation of regions without a processing of these. The experiments showed that the presented method is robust and accurate comparing with others public methods used for salient objects detection.**

Index Terms—**feature extraction, image processing, image segmentation, hypergraph data structures, object detection.**

I. Introduction

The high computation time corresponding to the segmentation algorithms and poor obtained results make that the image segmentation to remain a big challenge for computer science. Segmentation techniques are divided in two categories, segmentation based on region and segmentation based on contours. These approaches can be correlated if we consider the contours as boundaries of regions. In an environment where the segmentation is based on contours [1], the regions from an image can be considered as areas bounded by closed contours. In the case of image segmentation in salient regions, we can consider as goal the determination of closed contours of these regions.

In this article we present a method for image segmentation and extraction of salient visual objects which is designed on three levels. The segmentation method is designed in two levels and implements a computationally efficient treatment which is based on regions and uses local features and global features corresponding to the processed image. The algorithms used as data structures, the hypergraphs, which are built on a hexagonal grid structure over half the pixels of image.

The image representation is dual; each hypergraph corresponds to a forest of trees. The hypergraphs are processed to determine the minimum spanning tree (*MST*), in which each connected component represents a visual object. Thus the image segmentation is transformed in a hypergraph partitioning problem [2]. The condition for determining the nodes of the connected components is evaluated on the basis of two important characteristics: the color distance and syntactic features [3], which are given by geometric features of regions and spatially relations between them.

The structure of the article includes seven sections: the section I, ***Introduction***, shows shortly, the problem of image segmentation in the salient regions; the section II, ***Related Work***, summarizes the main approaches of the image segmentation and salient object detection; in chapter III, ***Image Preprocessing***, is described the hexagonal representation of image and the method for retrieval of the hypergraph data structure; the section IV presents the image segmentation method in two levels. In chapter V, ***Algorithm for Salient Objects Detection***, is shown the post-processing method of the results of segmentation for extraction of the segments which represent the salient objects. In section VI, ***Experimental Results***, are presented the results of the segmentation algorithm and the results of the method for extraction of the salient objects considering the Berkeley image dataset. Chapter VII, ***Conclusion***, describes the main contributions and the future research directions.

II. Related Work

For the phase of image segmentation it have been reviewed relevant approaches and aimed to improve the performances (results and time computing) by using a hexagonal structure which is overlaps on the pixels of processed image. The nearest researches for this segmentation approach are described in the following. The most segmentation methods that are based on graphs using data structures to assign weights to the edges of graph constructed for the pixels, such as “Minimum Spanning Trees” [4] and "Minimum-cut [5]. The main concept used in the classification algorithms which are based on graphs is the homogeneity of regions. For the algorithms that use as feature the color, homogeneity of regions is determined by this characteristic and the weights of edges are given as distance value between two colors. The first method based on graphs uses the fixed thresholds and local features to perform image segmentation. To overcome the problem of fixed thresholds, in [6] is determined the normalized weight for an edge using the lowest value of the pair of vertexes which correspond to that edge. Other methods [4], [7] use an adaptive criterion that depends on the local properties reported to global features. The methods based on the achievement of minimum cuts in graphs are designed to minimize the similarity between the sets of pixels which can be divided [8].

Most approaches use color and texture model to determine the homogeneity of objects extracted, in [3] is presented an additional source of information represented by syntactic features; such properties include homogeneity, solidity, regularity and inclusion.

In [9] it is proposed a function cost composed by the values of syntactic features in order to extract image

Digital Object Identifier 10.4316/AECE.2011.04018

1582-7445 © 2011 AECE

regions, but at the same time, tends to semantic analysis of images. The features used are homogeneity, regularity, solidity and inclusion of the regions. The function cost is integrated into the segmentation algorithm in which regions are extracted and classified. This approach was developed by [10] and uses the descriptors of MPEG-7 "Dominant Color" and "Region Shape". For each region extracted in the initial segmentation, the descriptors are determined and stored as attributes of the node region of the graph. The next step is to calculate the distance to each region for both descriptors in relation, to adjacent regions and determining a composite distance is used in the second stage of segmentation.

III. Image Preprocessing

Many of the color image segmentation problems are solved by choosing an appropriate color space in order to increase the level of differentiation between the color components from an image. We have used the HSV color space because it is the closest color space of human perception. Comparatively with native RGB color space for digital images, the HSV color space is uniform, and the function of calculating the distance between two colors is a uniform function. This property is based on the composition of colors in HSV space, which unlike the native RGB space, allows the definition of colors in accordance with the human perception. We used for determination of the distance between two colors the following Smith formula [11].

To reduce the computational time of segmentation and detection of salient objects algorithms, we have used a hexagonal structure constructed for the image pixels and it is shown in Fig. 1. The representation of the network is a planar graph $G = (V, E)$, in which each hexagon h corresponds to a node v in V. The set of edges is constructed for pairs of neighboring hexagons, taking into account that a hexagon is 6-connected network. The main advantages of using the network of hexagons are to reduce the size of the memory space associated with graph nodes and the development of efficient algorithms for segmentation and detection of salient objects. The number of pixels that are part of the hexagon is equal to (ntp / 2), where ntp is the total number of pixels and the number of edges is reduced, because the number of hexagons is less than ntp / 4 and the 6-connectivity neighborhood relationship is used instead of 8-connectivity.

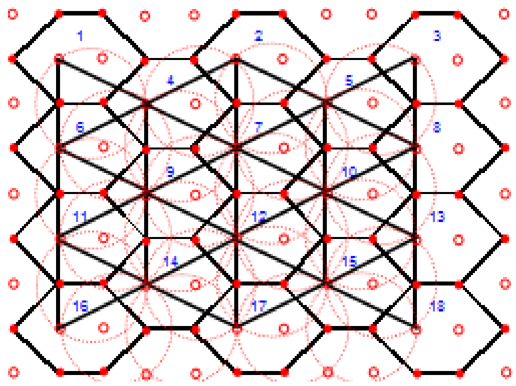

Figure 1. The hexagonal structure of virtual graph on the image pixels

Each hexagon from the set of hexagons is associated with two important attributes representing the dominant color and the center of gravity. To determine these attributes are used all the 8 pixels of a hexagon: six pixels corresponding to the vertexes of hexagon and two pixels on inside. For the *HSV* color space, the dominant color of a hexagon is determined as follows: extract the three components (h, s, v) for each pixel of a hexagon, the next step is sorting the three vector components, and the last phase involves choosing the dominant components as average of the components from the positions *4* and *5*, and determining the dominant color.

The pixels of image are divided into two sets, the set of pixels corresponding of all hexagons and complementary set of pixels. The mapping of pixels which belong to the hexagonal structure is immediate and not time consuming; it is determined the first index of the hexagon, and then determining all indices of the vertexes.

The preprocessing module is used to make the transition from graph model representation to the hypergraph model representation [12].

Definition. A hypergraph HG can be defined as a tuple $<V, HE>$, where V is a set of vertices, and HE is a set of hyperedges between the two or more vertices. Each hyperedge is a set of vertices: $HE \subseteq \{\{v_1, v_2 \ldots v_n\} \in V^n\}$.

An important step of the preprocessing method is the partitioning of the *G.E* based on the neighborhood relationships; within a partition can be found all neighboring hexagons with the color distance not exceeding a specified threshold. For every two neighboring hexagons, the threshold is given by the color distance between two pixels that form the edge common to the two hexagons. These values are determined during the construction phase of the hexagonal network, where for each hexagon are stored the six threshold values corresponding to its edges. By partitioning the initial set of edges of the graph *G*, we realized the transition from the graph representation at hypergraph representation of an image.

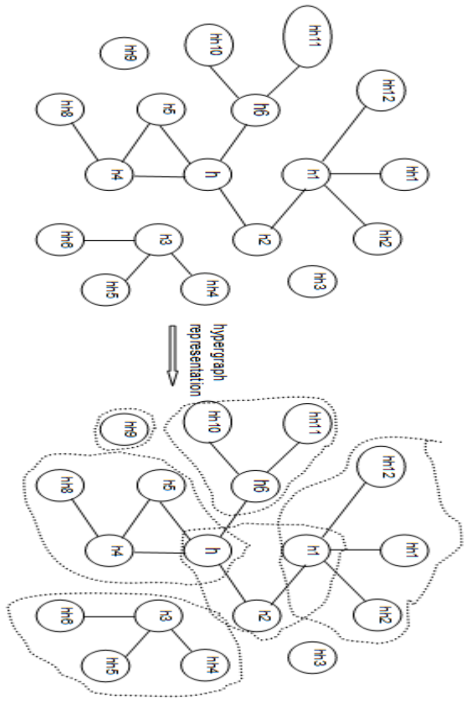

Figure 2. An example of graph and the equivalent hypergraph

This representation is used both for image segmentation, and for salient object detection. A partition of the set of edges, specified by a subset, is a hyperedge of the hypergraph *HG*, so that the initial set of edges, *E*, will be the corresponding set of hyperedges *HE*. The structure of the initial hypergraph is preserved and used in the detection of salient objects, as described in section V.

IV. The Image Segmentation Method

The segmentation module uses the determinate hypergraph in the previous stage of processing and extracts the set of regions that are salient objects present in the image. The last module, the detection of salient objects, determines for each region the corresponding object with semantic properties.

The set of elements used in segmentation, HE, contains hyperedges like nodes, each hyperedge of HE corresponds to a crowd of neighbors hexagons in terms of space and in terms of color. The weight of each edge, denoted by $w(he_i,he_j)$, is the value of the similarity metric between the two neighboring hyperedges he_i and he_j.

The segmentation S of HE represents a partitioning of HE in which each component C of S corresponds to the hyperedge from the GD hypergraph. The goal is to find a good segmentation to represent the salient objects in the image. The following definitions are used for segmentation: "fine" segmentation and "gross" segmentation as in [4], which seeks to formalize the human perception of visual salient objects.

Using an object-oriented model to represent hypergraphs [13] allowed the processing in a dynamic way of low-level features corresponding to segmented regions (to a hyperedge from hypergraph). To represent characteristics of extracted forms and neighborhood relationships between them, it was used decorating technique of the initial hypergraph built after segmentation with spatial information extracted from a set of rules written in a language based on binding rules before to use the Rete algorithm [14].

The proposed segmentation method is hierarchical based on the two levels and will produce a correct segmentation according to the above. The segmentation corresponding to the first-level based on color, builds a minimum spanning tree (MST) based on Kruskal's algorithm [15].

On the second level, there are used syntactic features to determine the weights of hyperedges and the refining of the minimum spanning tree by determining trees forest. The transition from a hypergraph with a lot of hyperedges to one with less hyperedges is based on the Borůvka's algorithm [16].

The general form of segmentation method is presented below:

```
General algorithm for image segmentation
Input: Initial list of pixels that belong to hexagonal network - P
Output: Composed list of regions corresponding to salient objects - R
Procedure Segmentation(P; R)
1.  createHexagonalGrid(P;R)
2.  HG <- createInitialHyperGraph(P;R)
3. createpartitionList (HG.HE; HG.HE.partitionList);
4.  createColorPartition(HG,R)
5.  HG` <- extractGraph(HG)
6.  createSyntacticPartition(HG, HG`)
7.  extractFinalComponents()
end procedure
```

For the first level of segmentation is proposed a modified form of Kruskal's algorithm that determines the minimum spanning tree, in which the hyperedges of hypergraphs generated at each step save the components corresponding to the regions detected in the segmentation process and the links between them.

The criterion used to compact regions in the first phase is based only on color feature. It is determined *k_th* constant value which depends on the standard deviation and on the average value of weights w (he_i, he_j) for all neighboring hyperedges (he_i,he_j) from *HE*.

The input hypergraph of ***createColorPartition*** procedure is an instance of *CHyperGraph* class with the following attributes: [***HE HE.partitionList compNb n m***]. HE attribute allows storage of a hyperedges crowd of hypergraph and a vector of vectors for CEdge class instances is used as a data structure. For hypergraph edges (***HG.HE.partitionList***), is used an object vector of class CEdge. N attributes, respectively m refers to the number of hyperedges respectively to the number of edges from hypergraph. ***CompNb*** field is used in the second level of segmentation to achieve graph compaction. The members of the class *CEdge* are [***index vi vj w inMst origEdge childEdge***].

Field ***inMst*** has truth value ***true*** for those edges that are part of the MST, and the two attributes: origEdge and childEdge are two references to the parent edge from input hypergraph, respectively to the child edge of the resulting hypergraph from the agreement.

To each edge of ***HG.HE.partitionList*** is attached a weight to represent the distance between two colors corresponding to a pair of two neighboring hexagons. For each item from the list of partitions scroll through the list of edges that have been ordered according to increasing share value and check if the distance is less than a threshold value, then is done the union of the components represented by the nodes corresponding to the edge. To avoid repeated unification of the same components we have to verify that if in the forest trees, dual representation, the trees corresponding to the two components have different roots. For a quick retrieval of the root, for each node is used a link to the parent node and the function *findRoot* calls recursive to scroll the tree from a leaf node to root node, which is sought.

Following this initial stage are determined regions, each region corresponding to a parent node in the minimum spanning tree. For the second level of segmentation it is used the same convergence criterion mentioned above, only be considered in addition the distances between the syntactic features.

By sharing the properties of homogeneity within regions, and between regions, it has been taken for reducing the distance between low-level features and the corresponding high-level interpretation of an image. Thus, it was used a new concept, visual syntactic features, which gives a structural description of an image. In [3] there are introduced the following visual syntax features: homogeneity, solidity, regularity, inclusion and symmetry. Because the design and structure are difficult to be determinate to the pixel level, the segmentation in two steps is required. The initial phase provides a simple segmentation (under-segmentation) to allow a determination of the shapes and spatial positions for the initial regions (segmentation based only on color feature). Based on this segmentation and using visual syntactic features, a second segmentation that extracts salient regions present in the image, is performed. Features used in the second level of segmentation, are presented below:

- Homogeneity- objects of interest tend to be consistent with the transition from outer to inner boundaries for defining contours.

- Solidity refers to composite objects which are formed in different parts spatial interconnected.

- Regularity - simple objects tend to have shapes and contours that presents some regularity (their form of complexity is low).

Visual syntactic features that express homogeneity, solidity and regularity are determined considering that basic unit, the hexagon, which represents 8 pixels (6 pixels corresponding to the vertexes of hexagon and the 2 pixels inside it). The area of a region is given by the total number of hexagons that forms the region and the perimeter is determined as being the hexagons which have at least one neighbor hexagon which does not belong to the region. The two values are updated and saved by each step of the segmentation algorithm.

All features described in this chapter are used both in the second stage of the application of segmentation algorithms, as well as the salient object detection algorithm. Their values are stored as attributes in the instances of share's class *(CWeight),* a single instance being attached to each hyperedge from regions hypergraph.

V. Algorithm For Salient Objects Detection

Visual attention is an important technique in many computer vision applications. The proposed algorithm aims to detect multiple salient objects. For this task, we considered as input the initial hypergraph - HG_i, built in the preprocessing phase, and the final hypergraph - HG_f, determined as output of the segmentation method. The hypergraph HG_i is determined based on presumed seed hexagons. These hexagons are selected randomly and if all color distances between them and neighboring hexagons are smaller than the specific threshold then the hexagons are grouped as hyperedges. In the segmentation process, only some of the initial hyperedges are confirmed as seeds. These hyperedges are detected in HG_i, respectively in HG_f. The detection method implements a pattern-matching algorithm on graphs; the initial hyperedges are modeled as queries for the final hypergraph.

The implementation of the match system is based on the graph grammars. Links between symbols of a graph grammar can be more complex than the linear relationship present in other grammars and through a graph structure; a query can be modeled and executed. Grammars are constructed using an inference algorithm that is based on the discovery of frequent sub-graphs of a graph. The induction system for a graph grammar, *SubdueGL* [17] was developed based on *SUBDUE* algorithm [18] and uses an approach of breakthrough sub-graphs, which focuses on the compression of data's crowd of the graph, otherwise seeking the most frequent sub-graphs. Although the approaches based on compression and frequency sub-graphs are closely related, they may produce different results because a sub-graph less frequently can produce a better overall compression for the set of data. Using the growth process of a graph, *SubdueGL* generates candidate sub-structures that can be used to compress data's crowd of graph. The original method uses as a value of compression, the description of a minimum length (*MDL*) to compare the compression of each sub-structures candidate from the set of data. The sub-structure with the highest value for the *MDL* is used in the stage of compression data's crowd. The process is repeated until the crowd of data is totally compressed - remains a single node or any sub-structure can not be found or when it has reached a specified number of iterations. Compression process causes a hierarchical reduction of connected sub-graphs, which corresponds directly with production rules of grammar. *SubdueGL* algorithm uses an approach as bottom-up "for learning graph grammar based on an iterative search in the input graph. To determine each grammar production, the algorithms search in the graph its best sub-structure to enable graph compression. The result sub-graphs are added in a queue structure with priorities built on length. Sub-structures identified so, lead to the right side of each the production rules of grammar.

In the next stage of the learning process, all occurrences of sub-structure identified are removed from the queue structure with priority are replaced by a single non-terminal node (left side) of the current production.

VI. Experimental Results

We have tested our method for image segmentation on a Berkeley Segmentation Dataset (*BSDB*) color image database [19]. The database contains two sets of images: a training set of 200 images and a test set of 100 images.

A. *IMAGE SEGMENTATION EXPERIMENTS*

For evaluating the performance of our segmentation method we have used two quality measures: precision and recall which give us the fraction of detections that are true positives rather than false positives and respectively the fraction of true positives that are detected rather than missed. We include for comparison the segmentation results obtained with other two alternative segmentation algorithms: *ROI-SEG* (Region-Of-Interest SEGmentation) [20] and Normalized-Cuts (*NC*) [5]. These algorithms were chosen because they are representative for methods in image segmentation, and because implementations of the algorithms are available. In Fig. 3 there are shown examples of segmentation images from BSDB.

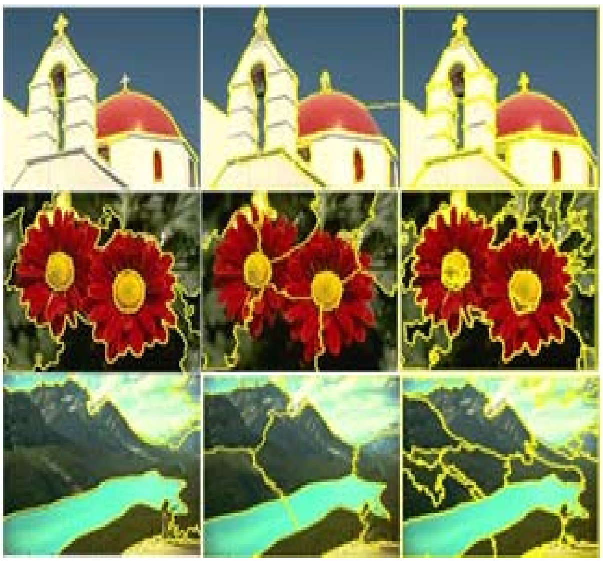

Figure 3. Examples of images segmented. From left to right: ROI-SEG segmentation, Normalized-Cuts segmentation and our segmentation

The evaluation of the proposed segmentation method (*SOD* - Salient Object Detection) is performed with precision/recall metric [21]. We evaluate the performance of the selected algorithms on the Berkeley Segmentation Database. We implemented our precision-recall framework instead of the segmentation benchmark of Berkeley because the contours of the visual objects determined by our method are binary images while Berkeley segmentation benchmark encourage soft boundary representation where pixels have attached a probability of membership to object contours. For combining the quantities of Precision (*P*) and Recall (*R*) in a single quality measure, we have used the harmonic mean function *F*. The maximal value of function *F* gives us the performance of segmentation on a set of images. In Fig. 4 it is shown the Precision/Recall curve of the segmentation methods.

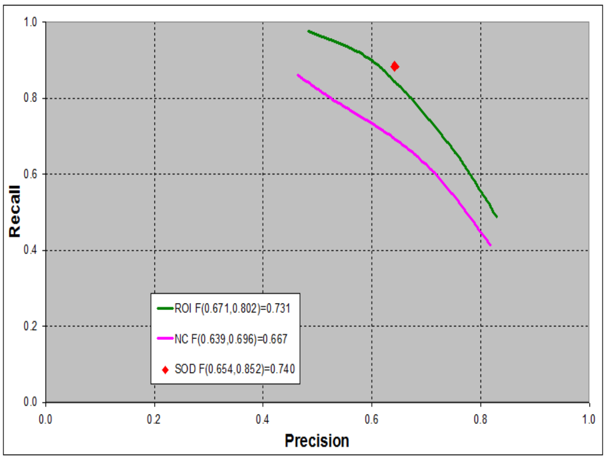

Figure 4. Precision-Recall Curve and F-Value Maximal

B. *SALIENT OBJECT DETECTION EXPERIMENTS*

For evaluating the performance of the salient object detection algorithm we have used Salient Objects Dataset [22]. This dataset is a collection of salient object boundaries based on *BSDB*; seven humans select salient object(s) for each image from the *BSDB*. Each subject is shown randomly a subset of the Berkeley segmentation dataset as boundaries overlapped on the corresponding images. We include for comparison the segmentation results obtained with the alternatively segmentation methods: ROI-SEG algorithm and Lossy data compression algorithm (CTM) [23].

The evaluation of the proposed algorithm for detection of the salient objects is based on Jaccard coefficient *P* of the region coincidence between the segmentation result and the ground truth. Metric (J-measure) based on areas, specifically for evaluating segmentations in terms of extracting salient objects [24]:

$$P(R,A)=\frac{|A\cap R|}{|A\cup R|},$$

where A represents the number of pixels of the salient region in the witness image, and R the number of pixels of salient region detected by the algorithm.

In Fig. 5 there are shown the values of *J-measure* for detection salient objects from BSDB. For our method, we have obtained a single point and for others we have obtained a range of values, corresponding to the input parameters.

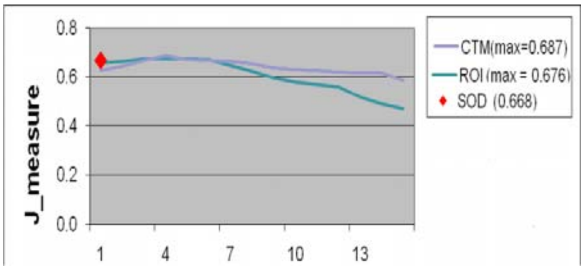

Figure 5. Values of the metric *J-Measure*

VII. CONCLUSION

In this paper we have presented a method for image segmentation and detection of the salient objects. The novelty of our contribution concerns three aspects: (I) we propose a new method to represent pixels, hexagonal representation, and used a hypergraph structure to store this representation; (II) we developed algorithms which were hierarchically structured in order to obtain regions which may be associated with semantic meanings; (III) we have designed and implemented optimal algorithms to detect salient objects, which allows interactive using of syntactic properties in the segmentation. Experimental results prove that our method is faster, robust and accurate comparative with existing salient object detection methods. The object oriented approach used to implement segmentation and detection of salient objects algorithms provides a good correlation with the future phase of the semantic image processing; experiments showed that segmented regions can be used directly in the annotation without the need of the other post-processing phase.

ACKNOWLEDGMENT

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

REFERENCES

[1] D. Jacobs. ”Robust and efficient detection of salient convex groups”, In IEEE Transaction on Pattern Analysis and Machine Intelligence, pp. 23 – 37, 1996.

[2] S. Rital, H. Cherifi and S. Miguet. “Weighted Adaptive Neighborhood Hypergraph Partitioning for Image Segmentation”, Lecture Notes in Computer Science, 3687, pp. 522 – 531, 2005.

[3] C.F. Bennstrom and J.R. Casas. "Binary-partition-tree creation using a quasi-inclusion criterion", In Proceedings of the Eighth International Conference on Information Visualization, London, UK, pp. 259 – 294, 2004.

[4] P.F. Felzenszwalb and W.D. Huttenlocher. "Efficient Graph-Based Image Segmentation", International Journal of Computer Vision, pp. 167 – 181, 2004.

[5] J. Shi and J. Malik. "Normalized cuts and image segmentation", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 731 – 737, 2000.

[6] R. Urquhar. "Graph theoretical clustering based on limited neighborhood sets", In Pattern Recognition Letters, 15, pp. 173 – 187, 1982.

[7] L. Guigues, L.M. Herve and L.-P. Cocquerez. "The hierarchy of the cocoons of a graph and its application to image segmentation", In Pattern Recognition Letters, 24, pp. 1059 – 1066, 2003.

[8] Y. Gdalyahu, D. Weinshall and M. Werman. "Self-organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database organization", In IEEE Transaction on Pattern Analysis and Machine Intelligence, 23, pp. 1053 – 1074, 2001.

[9] T. Adamek, N.E. O'Connor and N. Murphy. "Region-based segmentation of images using syntactic visual features", In WIAMIS 2005 - 6th International Workshop on Image Analysis for Multimedia Interactive Services, 2005.

[10] T. Athanasiadis, V. Tzouvaras Petridis, K.F. Precioso, Y. Avrithis and I. Kompatsiaris. "Using a Multimedia Ontology Infrastructure for Semantic Annotation of Multimedia Content", In The 5th International Workshop on Knowledge Markup and Semantic Annotation at the 4th International Semantic Web Conference, Galway, Ireland, 2005.

[11] J.R. Smith and S.-F. Chang. "VisualSEEk: a Fully Automated Content-Based Image Query System", In ACM Multimedia, Boston, MA, pp. 87 – 98, 1996.

[12] A. Bretto and L. Gillibert. "Hypergraph-based imge representation", In Graph-Based Representations in Pattern Recognition, pp. 1-11, 2005.

[13] E. Ganea and M. Brezovan. "An Hypegraph Object-Oriented Model for Image Segmentation and Annotation", In Proceedings of the International Multiconference on Computer Science and Information Technology, pp. 695 – 701, 2010.

[14] C. Forgy. "Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem", In Artificial Intelligence, 19, pp 17 – 37, 1982.

[15] J. Kruskal. Proceedings of the American Mathematical Society, pp. 48–50, 1956.

[16] O. Borůvka. "O jistém problému minimálním (About a certain minimal problem)", In Práce mor. Přírodověd, pp. 37–58, 1926.

[17] I. Jonyer, L. Holder and D. Cook. "Concept Formation Using Graph Grammars", In Proceedings of the KDD Workshop on Multi-Relational Data Mining, 2002.

[18] L. B. Holder. "Empirical Substructure Discovery", In Proceedings of the Sixth International Workshop on Machine Learning, pp. 133-136, 1989.

[19] D. Martin, C. Fowlkes, D. Tal and J. Malik. "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics", In IEEE International Conference on Computer Vision, pp. 416 – 423, 2001.

[20] M. Donoser and H. Bischof. "ROI-SEG: Unsupervised Color Segmentation by Combining Differently Focused Sub Results", In IEEE Conference on Computer Vision and Pattern Recognition, pp.1-8, 2007.

[21] C. Fowlkes, D. Martin and J. Malik. "Learning affinity functions for image segmentation: combining patch-based and gradient-based approaches", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, pp. 54 – 61 , 2003.

[22] V. Movahedi and J. H. Elder. "Segmenting Salient Objects: How do we measure success?", Poster at CVR09, Centre for Vision Research CVR Conference, 2009.

[23] A.Y Yang., J. Wright, M. Yi and S.S Sastry. "Unsupervised Segmentation of Natural Images via Lossy Data Compression", In Computer Vision and Image Understanding, vol. 110, pp. 212 – 225, 2008.

[24] F. Ge, S. Wang and T. Liu. "New benchmark for image segmentation evaluation", In Journal of Electronic Imaging, vol. 16, 2007.