Estimation of Distribution Algorithm for the Max-Cut Problem

Samuel de Sousa, Yll Haxhimusa, and Walter G. Kropatsch

Vienna University of Technology
Pattern Recognition and Image Processing Group
Vienna, Austria
sam,yll,krw@prip.tuwien.ac.at

Abstract. In this paper, we investigate the Max-Cut problem and propose a probabilistic heuristic to address its classic and weighted version. Our approach is based on the Estimation of Distribution Algorithm (EDA) that creates a population of individuals capable of evolving at each generation towards the global solution. We have applied the Max-Cut problem for image segmentation and defined the edges' weights as a modified function of the L2 norm between the RGB values of nodes. The main goal of this paper is to introduce a heuristic for Max-Cut and additionally to investigate how it can be applied in the segmentation context.

Keywords: max-cut, graph cut, eda, segmentation.

1 Introduction

Many problems in computer vision end up by assigning a certain label (corresponding to a class) to a pixel or a region in the image. Therefore, it is required to choose a proper representation in order to assign such label. Many algorithms that are suitable for graph theoretical problems can also be applied in the computer vision domain if the problem is modeled using the graph formulation. Thus, the choice of representing images as graphs has several advantages over other approaches.

A graph theoretical clustering algorithm consists of searching for a certain combinatorial structure in the edge weighted graph, such as the minimum spanning tree [9, 16] or normalized cut [26, 28]. Among those methods, the complete linkage clustering algorithm [20] reduces the search to the problem of finding a complete subgraph (i.e. the maximal clique [24]) in the image. Also, graph-based spectral methods have been successfully used for clustering [21] as well.

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, Max-Cut is the problem of finding a partition $(T, \bar{T})$ of the nodes V that maximizes the number of edges between T and its complement set $\bar{T}$. This problem belongs to the class $\mathcal{NP}$-Hard [11], therefore, no polynomial time algorithm is able to solve Max-Cut for any arbitrary class of graphs, although several approximations have been proposed. In fact, for planar graphs, it is possible to compute the maximum cut in polynomial time [14].

In this paper, we propose a heuristic for the Max-Cut problem for any arbitrary class of graphs and we model the weighted Max-Cut as the problem of maximizing the sum of weighted edges between two sets of nodes. Nodes of the same set connected by an edge should be merged into one single cluster and nodes of different sets connected by a bichromatic edge should remain separated. Our probabilistic heuristic is based on the Estimation of Distribution Algorithm (EDA) [1]. Moreover, we have used the segmentation task to show the applicability of the problem in the pattern recognition domain. We address the weighted version of Max-Cut whose weights belong to $\mathbb{R}$.

The remainder of this paper is organized as follows: Section 2 provides a literature review on graph-based segmentation. The Max-Cut problem is introduced and explained in Section 3. Our heuristic is disclosed in Section 4. We map the theoretical graph problem into image segmentation in Section 5. Our experiments are described in Section 6. Finally, we present our conclusions and future directions in Section 7.

2 Related Work

Early graph-based clustering methods [29] use fixed thresholds and local measures in computing a cluster, i.e. the minimum spanning tree (MST) is computed. The clustering criterion is to break the MST edges with the largest weight. The work of Urquhart [27] attempts to overcome the problem of fixed threshold by normalizing the weight of an edge using the smallest weight incident on the vertices touching that edge. The methods in [9, 16] use an adaptive criterion that depend on local properties rather than global ones and have the minimum spanning tree as the base algorithm. It is shown in [7] that minimum spanning tree clustering technique, although unsupervised one, approaches the performance of 'Bayes classifier', as the number of sample points from each class increases.

The methods based on minimum cuts [4, 6] in graph are designed to minimize the similarity between pixels that are being split [28, 26]. Authors in [28] define a cut criterion, but it was biased toward finding small components. Shi and Malik [26] developed the normalized cut criterion to address this bias, which takes into consideration self-similarity of regions. These cut-criterion methods capture the non-local properties of the image, in contrast with the simple graph-based methods such as breaking edges in the MST. However they provide only a characterization of such cut rather than of final segmentation as it is provided by Felzenszwalb [9]. Shi and Malik [26] developed an approximation method for computing the minimum normalized cut, closely related to spectral graph methods, e.g [10].

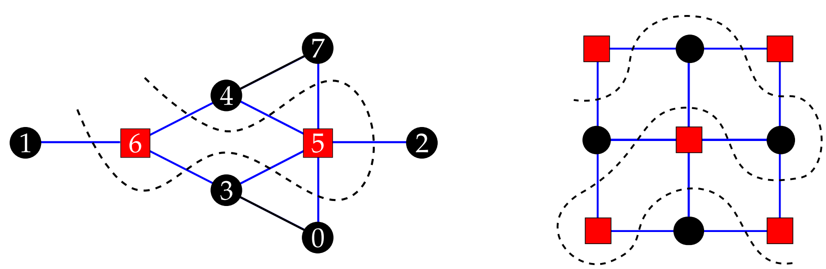
The minimal spanning tree and the minimum cut are explicitly defined on weighted edge graph, whereas the concept of a maximal clique is defined on unweighted edge graphs. As a consequence, maximal clique based clustering algorithms work on unweighted graphs derived from the edge weighted graphs by means of thresholding [17]. Pavan and Pelillo [24] generalized the concept of maximal clique to weighted graphs.

Markov Random Field (MRF) has been used for clustering [12]. However the use of MRF for image clustering usually leads to $\mathcal{NP}$-Hard problems. The graph-based approximation method for MRF problems [5] yields practical solution, if the number of labels for the pixel is small, which limits these methods for use in segmentation and clustering.

A disadvantage of graph theoretical approaches for image segmentation, i.e. clustering, is that these algorithms in some real-time applications are very time consuming.

3 Max-Cut

Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a cut in the graph is a partition of the vertices $\mathcal{V}$ into T and $\bar{T}$. Let $\bar{T} = \mathcal{V} \setminus T$ be the complement set of T and $E(T, \bar{T})$ be the set of edges connecting a vertex in T with another in $\bar{T}$. The Max-Cut problem consists of finding the cut that maximizes $|E(T, \bar{T})|$. It is one of the problems of Karp [19] and it belongs to the $\mathcal{NP}$-Hard class. An example of a maximum cut is shown in Figure 1a.



(a) (b)

Fig. 1: (a) The cut that maximizes the number of bichromatic edges between the T (red squared nodes) and $\bar{T}$ (black circular nodes). (b) The Maximum Cut of a 4-connected representation.

Considering the complexity of $\mathcal{NP}$-Hard problems, common approaches consist of creating *ρ-approximated algorithms*, *i.e.* polynomial algorithms whose solution is ρ times the optimal solution [13]. There is a vast and growing amount of algorithms to deal with $\mathcal{NP}$-Hard problems. Researchers seek to find approaches that are capable of achieving better approximation rates as well as they attempt to demonstrate that there are no better approximations above a certain threshold. For instance, Goemans-Williason [13] proposed an approximated algorithm to the Max-Cut problem whose rate is close to:

$$\alpha = min_{0 \leq \theta \leq \pi} \frac{2}{\pi} \frac{\theta}{1 - \cos\theta} > 0.87856. \quad (1)$$

According to the authors, the approach was a substantial improvement in nearly twenty years. Subsequently, Håstad [15] sets a barrier that unless $\mathcal{P} = \mathcal{NP}$, MAX-CUT can not be approximated by a deterministic algorithm that adheres to a rate strictly exceeding 16/17 [15, 18].

Finally, Kaporis et al. [18] proposed a deterministic algorithm in polynomial time that approximates almost all instances of MAX-CUT with a rate above the Håstad threshold. Their solution became the first improvement of MAX-CUT after a decade [18]. Thus, seeking to break the barrier imposed by Håstad, Kaporis et al. use two strategies: Assuming that the maximum cut is not known, it becomes necessary to (i) find an upper bound for the MAX-CUT as well as (ii) to improve significantly the known lower bounds.

4 Estimation of Distribution Algorithm

Evolutionary Algorithms (EA) and more specifically Estimation of Distribution Algorithms (EDA) consist of an ensemble of individuals (agents) sampling the search space for potential solutions of a given problem. Those individuals have a knowledge about the laws of the environment and a quality measurement that represents how able those individuals are to solve the problem [2]. Those solutions are created based on chromosomes and each chromosome has a probability p of being chosen. Evolutionary Algorithms have been applied to the Max-Cut problem before such as in [8], where the authors propose a hybrid evolutionary algorithm using Variable Neighborhood Search and Memetic Algorithm.

In this paper, chromosomes are represented by the nodes of the graph and the nodes should follow a probability distribution, such as the uniform distribution in which all nodes are likely to be chosen with the same probability.

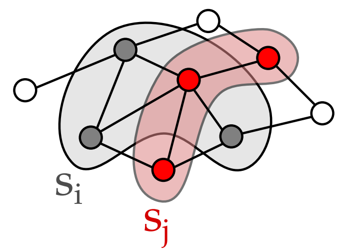
We apply the Population Based Incremented Learning (PBIL) algorithm [3] which takes a vector of probabilities $P = \{p(v_1), p(v_2), \ldots, p(v_n)\}$ associated with how capable a chromosome (v_i) is to provide a solution for the problem. We create a population $S = \{s_1, s_2, \ldots, s_n\}$ of individuals that will choose a subset of nodes to compose the cut T. The best individual (s^g_{best}) of generation g is selected to survive and it is added into generation $g+1$. Hence, we guarantee that $s^{g+1}_{best} \geq s^g_{best}$. The probability of $p^g(v_i)$ is updated as follows:

$$p^{g+1}(v_i) = (1 - \alpha) \times p^g(v_i) + \alpha \times \frac{\sum_{k=1}^{|S|}[v_i \in s_k]}{|S|}, \tag{2}$$

where $\alpha \in [0, 1]$ is a learning rate parameter that weights the impact of both terms of the formula. By using a high α, we decrease the impact of the probability in the previous generation ($(1 - \alpha) \times p^g(v_i)$) and we increase the impact of having this node chosen by many individuals. In our experiments, $\alpha = 0.5$, which means we balance equally the importance of both terms. In this algorithm, each individuals choose one or more nodes to belong to the solution, the number of nodes chosen by each individual is computed randomly in such a way to follow the probability distribution of the nodes.

Data: $\mathcal{G}(\mathcal{V},\mathcal{E}), P, G$
Result: S_{best}
1 **begin**
2     $S_{best} = \varnothing$
3   **for** i = *1* ***to*** G **do**
4         $S^i \leftarrow population(\mathcal{G}(\mathcal{V},\mathcal{E}), P)$;
5         $S^i \leftarrow S^i \cup \{S_{best}\}$;
6         $S^i_{best} \leftarrow evaluate(S^i)$;
7   **if** $S^i_{best} > S_{best}$ **then**
8             $S_{best} \leftarrow S^i_{best}$;
9   **end**
10   **end**
11 **end**

(a) Estimation of Distribution Algorithm: Individuals (S) evolve through generations G until the best individual (s_{best}) is found.



(b) Each individual (e.g. s_i, s_j) selects a cut T of the graph. Probability of nodes in generation $g+1$ is updated using s^g_{best}.

Fig. 2: Estimation of Distribution Algorithm.

Figure 2a shows the Estimation of Distribution Algorithm. Line 4 creates a population based on the current graph and on the nodes' probabilities P. We add our best individual S_{best} into the current population which ensures that the next generation will produce as good results as the current one. Figure 2b shows an example of two individuals selecting a subset of nodes of the graph as their candidate solution for the cut. The best individual survives the current generation and evolves.

5 Max-Cut-based Image Segmentation

In order to segment the image, we first create a graph representation. One approach could consider on assigning each pixel of the image as a node and using a 4-connected neighborhood to create the edges. However, considering that Max-Cut tries to maximize the bichromatic edges, by using this representation, we might end up with a cut such that all edges are bichromatic as displayed in Figure 1b.

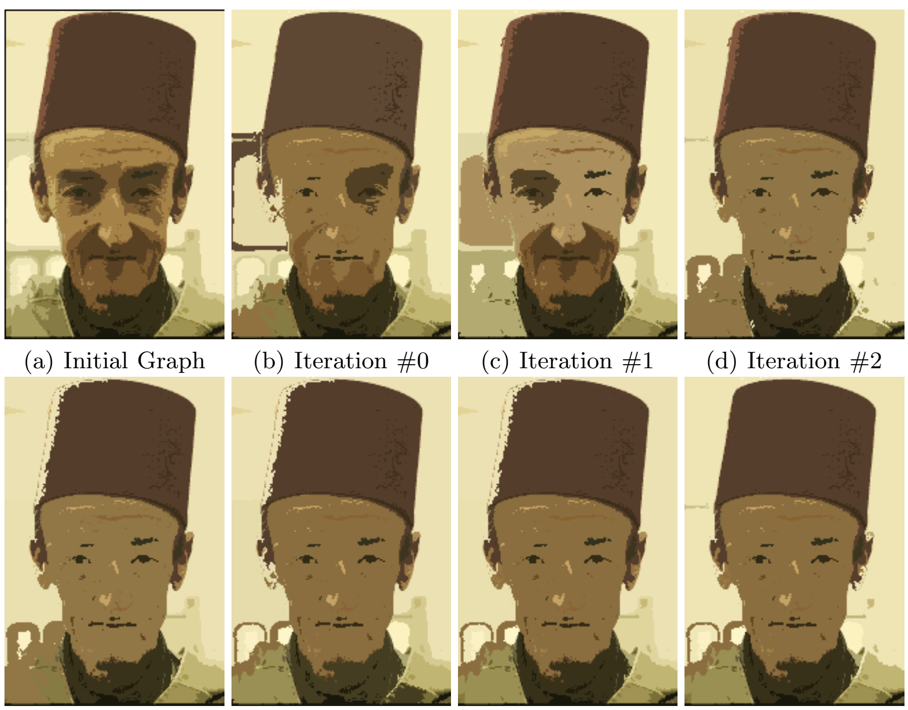
Our graph representation is built as follows: Given a non visited region in the image, we add a seed to that region and we grow this seed by adding pixels whose absolute difference to the seed does not exceed a certain threshold t (in our experiments $t = 40$). For each region we average the intensity components of the RGB of all pixels belonging to that region and we define the weight of an edge (e_w) between two nodes (v_i, v_j) as a modified function of the $L2$ norm between the two regions:

$$e_w(v_i, v_j) = (2 \times [||rgb(v_i) - rgb(v_j)||_2 > t] - 1) \times ||rgb(v_i) - rgb(v_j)||_2. \quad (3)$$

This equation states that based on a threshold t, an edge between two nodes might be either positive or negative. This negative weight assumption prevents the EDA of choosing certain edges in the segmentation process. In the classic Max-Cut problem, an edge has the weight of 1. But in the weighted version we maximize $\sum e_w(v_i, v_j); v_i \in T, v_j \in \bar{T}; e_w(v_i, v_j) \in \mathbb{R}$ which is the sum of the bichromatic edges' weights.

During the construction of the graph, each pixel in the boundary of two regions will produce an edge, which means that nodes modeling bigger regions will have more edges. In this way, our implementation of graph allows multiple edges between two nodes in order to give preference to bigger regions in the segmentation process. The addition of negative edges work as a mechanism to prevent Max-Cut of choosing those edges in the search for the global maxima. The negative edges penalize the cost function in such a way that if the EDA chooses one such edge, the cost will be smaller than not choosing that edge. Hence, we still try to maximize the sum of $e_w(v_i, v_j)$, however, some edges will not be added into the final cut.

We map the nodes' cut into the region segmentation as follows: whenever there is a bichromatic edge in the graph, there will be isolated regions in the image, *i.e.* distinct regions in the final segmented image as generated. However,



(a) Initial Graph (b) Iteration #0 (c) Iteration #1 (d) Iteration #2

(e) Iteration #12 (f) Iteration #65 (g) Iteration #68 (h) Iteration #91

Fig. 3: Evolution of the segmentation through the EDA generations.

when there is an edge between two nodes of the same color, those nodes will be merged into a single region in the final segmentation.

6 Experiments

We have applied our algorithm on images of the Berkeley database [22]. Figure 3 shows an example of the segmentation results during the evolution of the EDA. The initial graph generated by the watershed technique is displayed in Figure 3a and final result of segmentation is available in Figure 3h.

As aforementioned, this paper attempts to apply the Max-Cut into the segmentation problem. We have computed the results for other images (Elephants, Airplane, Church) in Figure 4. For all images, nodes that were merged into a single node succeeded to do so, due to the fact that the distance between regions in the RGB space are relatively small. Hence, those edges were modeled as negative edges and were not selected by the EDA to compose the final result because the addition of negative edges would penalize the cost.

For instance, the segmentation result obtained in the Elephants' picture did not merge the blue sky in the right upper corner considering that the distance between those two regions in RGB was higher than the threshold used during the optimization. On the other hand, the left-most elephant was merged with a piece of sky, which clearly does not produce a correct result. However, many segmentation algorithms, including many graph cuts use interaction with users by adding manual scribbles or regions containing the object of interest to help the segmentation procedure [23, 25]. The negative edges' assumption is an attempt to improve the segmentation results by computing a similarity measure between regions. However, other mechanisms such as brushing or scribbling could be used to map Max-Cut in the segmentation. One could add some knowledge about the spatial location of the regions to the cost function as an attempt to bring color information and spatial configuration together to improve the results.

7 Conclusions

Graph-based representation of an image has many advantages over other representations due to the fact that many problems can be posed in graph theoretical manner. In this paper we investigate the Max-Cut problem which belongs to the class of problems called $\mathcal{NP}$-Hard and use the Algorithm of Estimation of Distribution to compute a solution for it.

The focus of this paper is to show a heuristic for the Max-Cut problem and to show that it can be applied in the segmentation task by assuming, for instance, the negative edges' concept. We are continuously investigating how this problem could be better explored for segmentation.

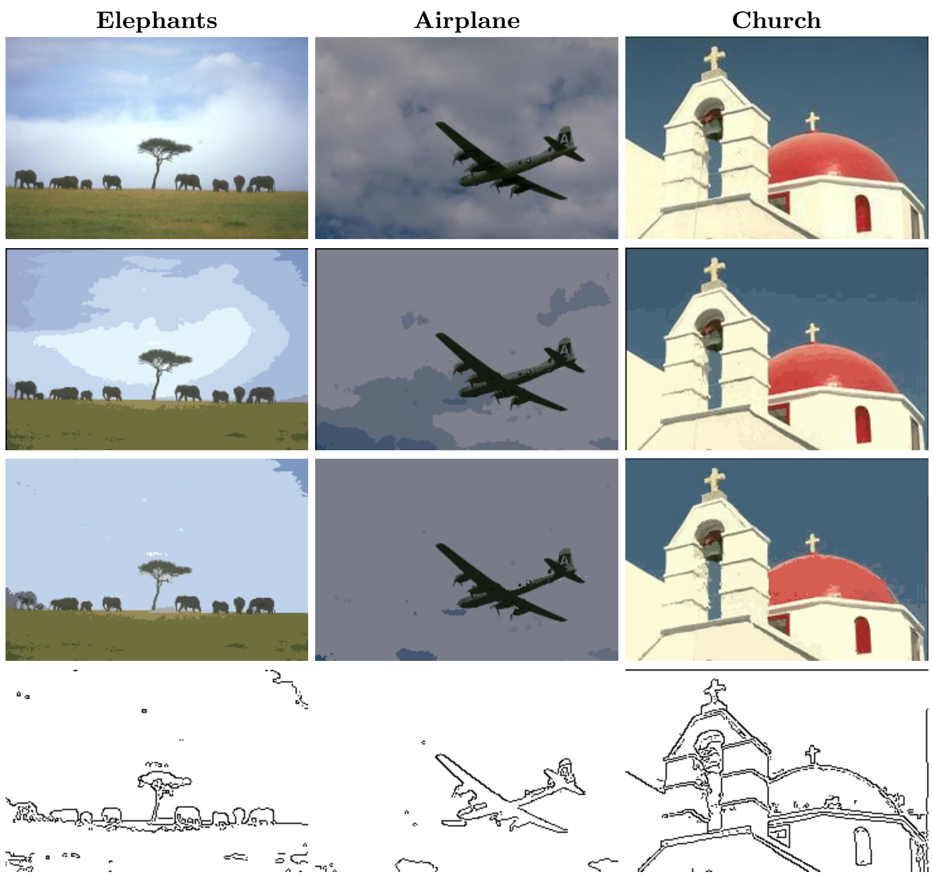


Fig. 4: Segmentation of images from the Berkeley database [22] using the EDA. First row displays the input image. The second row displays the initial graph representation. Our segmentation obtained by Max-Cut is displayed in the third row. We show the edges between regions in the fourth row.

Acknowledgments

Samuel de Sousa acknowledges financial support by the Austrian Agency for International Cooperation in Education & Research (OeAD) within the OeAD Sonderstipendien program, financed by the Vienna PhD School of Informatics.

References

1. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation (Genetic Algorithms and Evolutionary Computation)*. Springer, October 2001.
2. T. Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, Oxford, UK, 1996.
3. S. Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning, 1994.
4. Y. Boykov and G. F. Lea. Graph Cuts and Efficient N-D Image Segmentation. *Int. J. Comput. Vision*, 70(2):109–131, Nov. 2006.
5. Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–655, USA, 1998. IEEE Computer Society. Also as Cornell CS technical report TR97-1658, December 3, 1997.
6. Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. volume 1, pages 105–112, 2001.
7. N. Chowdhury and C. Murhty. Minimal spanning tree based clustering technique: Relationship whith bayes classifier. *Pattern Recognition*, 30(11):1919–1929, 1997.
8. A. Duarte, A. Sánchez, F. Fernández, and R. Cabido. A low-level hybridization between memetic algorithm and vns for the max-cut problem. In *Proceedings of the 2005 conference on Genetic and evolutionary computation*, GECCO '05, pages 999–1006, New York, NY, USA, 2005. ACM.
9. P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.
10. M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Checz Mathematical Journal*, 25(100):619–633, 1975.
11. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
12. S. Geman and D. Geman. Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, November 1984.
13. M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995.
14. F. Hadlock. Finding a Maximum Cut of a Planar Graph in Polynomial Time. *SIAM Journal on Computing*, 4(3):221–225, 1975.
15. J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.
16. Y. Haxhimusa and W. G. Kropatsch. Segmentation Graph Hierarchies. In A. Fred, T. Caelli, R. P. Duin, A. Campilho, and D. de Ridder, editors, *Proceedings of Joint International Workshops on Structural, Syntactic, and Statistical Pattern Recognition S+SSPR 2004*, volume 3138 of *Lecture Notes in Computer Science*, pages 343–351, Lisbon, Portugal, 2004. Springer, Berlin Heidelberg, New York.

17. A. K. Jain and R. Dubes. *Algorithms for Clustering Data.* Prentice Hall, Berlin, 1988.
18. A. C. Kaporis, L. M. Kirousis, and E. C. Stavropoulos. Approximating almost all instances of max-cut within a ratio above the håstad threshold. In *ESA'06: Proceedings of the 14th conference on Annual European Symposium*, pages 432–443, London, UK, 2006. Springer-Verlag.
19. R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.
20. J. Lance and W. Williams. A general theory of classificatory sorting strategies: I hierarchical systems. *Journal on Computing*, 9:373–380, 1967.
21. B. Luo, R. C. Wilson, and E. R. Hancock. Spectral feature vectors for graph clustering. In T. Caelli, A. Amin, R. P. W. Duin, M. S. Kamel, and D. de Ridder, editors, *Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshops SSPR 2002 and SPR 2002, Windsor, Ontario, Canada, August 6-9, 2002, Proceedings*, volume 2396 of *Lecture Notes in Computer Science*, pages 83–93. Springer, 2002.
22. 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 *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.
23. A. Noma, A. B. Graciano, R. M. C. Jr, L. A. Consularo, and I. Bloch. Interactive image segmentation by matching attributed relational graphs. *Pattern Recognition*, 45(3):1159 – 1179, 2012.
24. M. Pavan and M. Pelillo. Graph-theoretic approach to clustring and segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 145–152. IEEE Computer Society, 2003.
25. C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, Aug. 2004.
26. J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
27. R. Urquhart. Graph theoretical clustering based on limited neighborhood sets. *Pattern Recognition*, 15(3):173–187, 1982.
28. Z. Wu and R. M. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, November 1993.
29. C. Zahn. Graph-theoretical methods for detecting and describing gestal clusters. In *IEEE Transaction on Computing*, volume 20, pages 68–86, 1971.