NEIGHBORHOOD HYPERGRAPH PARTITIONING FOR IMAGE SEGMENTATION

Soufiane Rital, Hocine Cherifi
Faculté des Sciences Mirande, Université de Bourgogne
9, Av. Alain Savary, BP 47870, 21078 Dijon, France.

Serge Miguet
LIRIS, Université Lumière Lyon 2
5, avenue Pierre Mendès France, 69500 Bron, France.

Keywords: Graph, Hypergraph, Neighborhood hypergraph, multilevel hypergraph partitioning, image segmentation, edge detection.

Abstract: The aim of this paper is to introduce a multilevel neighborhood hypergraph partitioning for image segmentation. Our proposed approach uses the image neighborhood hypergraph model introduced in our last works and the algorithm of multilevel hypergraph partitioning introduced by George Karypis. To evaluate the algorithm performance, experiments were carried out on a group of gray scale images. The results show that the proposed segmentation approach find the region properly from images as compared to image segmentation algorithm using normalized cut criteria.

1 INTRODUCTION

Image segmentation, whose goal is the partition of the image domain, is a long standing research subject in computer vision (Pal and Pal, 1993). The resulting image subdomains, which can be denoted as image segments, satisfy some condition of homogeneity, e.g., present the same color or some kind of texture. Image segmentation plays a principal role in the realization of computer vision applications, as a previous stage for the recognition of different image elements or objects. Several algorithms have been introduced to tackle this problem. It can be classified into five approaches (Fan et al., 2001)(Navon et al., 2005), namely: (a) Histogram-based methods, (b) boundary-based methods, (c) region-based methods, (d) hybrid-based methods, and (e) graph-based methods. In this paper we briefly consider some of the related work that is most relevant to our approach: graph based methods.

There has been significant interest in graph-based approaches to image segmentation in the past few years (Wu and Leahy, 1993), (Sarkar and Boyer, 1996), (Gdalyahu et al., 2001), (Soundararajan and Sarkar, 2001), (Shi and Malik, 2000), (Soundararajan and Sarkar, 2003), (Wang and Siskind, 2003). The common theme underlying these approaches is the formulation of a weighted graph $G = (X, e)$. The elements in X are pixels and the weight of an edge is some measure of the dissimilarity between the two pixels connected by that edge (e.g., the difference in intensity, color, motion, location or some other local attribute). This graph is partitioned into components in a way that minimizes some specified cost function of the vertices in the components and/or the boundary between those components.

Wu and Leahy (Wu and Leahy, 1993) were the first to introduce the general approach of segmenting images by way of optimally partitioning an undirected graph using a global cost function. They minimized a cost function formulated as a boundary cost metric, the sum of the edge weights along a cut boundary: $cut(A, B) = \sum_{i \in A, j \in B} w(i, j)$, and with the obvious constraints $A \cup B = X$, $A \cap B = \emptyset$, and $A \neq \emptyset$, $B \neq \emptyset$. This cost function has a bias toward finding small components. Cox et al. (Cox et al., 1996) attempted to alleviate this bias by normalizing the boundary-cost metric. They proposed a cost function, ratio regions, formulated as a ratio between a boundary-cost metric and a segmentation-area metric. Shi and Malik (Shi and Malik, 2000) and Sarkar and Soundararajan (Soundararajan and Sarkar, 2003) adopted different cost functions, normalized cut and average cut, formulated as sums of two ratios between boundary-cost and segment-area-related metrics, also in undirected graphs.

The cost function defined by Shi and Malik attempts to rectify the tendency of the cut algorithm to

Rital S., Cherifi H. and Miguet S. (2006).
NEIGHBORHOOD HYPERGRAPH PARTITIONING FOR IMAGE SEGMENTATION.
In *Proceedings of the First International Conference on Computer Vision Theory and Applications*, pages 331-337
DOI: 10.5220/0001376003310337

prefer isolated nodes of the graph. The Normalized cut criterion consists of minimizing:

$$Ncut(A,B) = \frac{cut(A,B)}{Assoc(A,X)} + \frac{cut(A,B)}{Assoc(B,X)} \quad (1)$$

where $assoc(A,X) = \sum_{i \in A, j \in B} w(i,j)$ which intuitively represents the connection cost from the nodes in the sub-graph A to all nodes in the graph X.

An alternative to the graph cut approach is to look for cycles in a graph embedded in the image plane. In this case, the cost function formulated as a ratio of two different boundary-cost metrics in a directed graph. This cost functions can alleviate area-related biases in appropriate circumstances. In (Wang and Siskind, 2003), Wang and Siskind present a cost function, cut ratio, namely, the ratio of the corresponding sums of two different weights associated with edges along the cut boundary in an undirected graph.

In most cases, we usually want to partition (segment) an image into a larger number of parts; i.e., we want a k-way partitioning algorithm which divides our image into k parts. One way of partitioning a graph into more than two components is to recursively bipartition the graph until some termination criterion is met. Often, the termination criterion is based on the same cost function that is used for bipartitioning (Shi and Malik, 2000),(Wu and Leahy, 1993),(Wang and Siskind, 2003). The recursive k-way partitioning algorithm is time consuming because we need to apply the same algorithm at each new iteration of the hierarchy. Ideally, we would like to have a direct k-way algorithm which outputs the k disjoint areas in a single iteration (Hadley et al., 1992). A common solution is to convert the partitioning problem into a clustering problem. Shi and Malik (Shi and Malik, 2000) define a new criterion that can be used in a k-way algorithm.

$$Ncut_k(A_1, A_2, \ldots, A_k) = \frac{cut(A_1, X - A_1)}{Assoc(A_1, X)} \quad (2)$$
$$+ \frac{cut(A_2, X - A_2)}{Assoc(A_2, X)} + \ldots + \frac{cut(A_k, X - A_k)}{Assoc(A_k, X)}$$

where A_i is the ith sub-graph of G. Tal and Malik (Tal and Malik, 2001) used the k-means algorithm to find a pre-selected number of clusters within the space spanned by the non-zero, smallest e eigenvectors. For those cases where the number of clusters is not known, the authors proposed using several values of k and then selecting that k which minimized the criterion:

$$Ncut_k(A_1, \ldots, A_k)/k^2. \quad (3)$$

In (Martinez et al., 2004), Aleix M. Martinez et al. investigate in the other approaches to non-parametric clustering in the eigenspace of the affinity matrix. The authors use the method of Koontz and Fukunaga (Koontz and Fukunaga, 1972) that has the advantage of automatically determining the optimal value of k as the data are grouped into clusters.

The main drawback of proximity graphs is their use of binary neighborhood relations. An image is an organization of objects in a space, and the appropriate relational algebra is not necessarily a binary one. The corresponding representation for images data with higher order relationship is a hypergraph. By regarding each set as a generalized edge one obtains a structure called a hypergraph (Fig. 1). Similarly to graphs, hypergraphs can be used to represent the structure of many applications, such as data dependencies in distributed databases (Koyutrk and Aykanat, 2005), component connectivity in VLSI circuits (Karypis et al., 1999) and image analysis (Rital et al., 2001) (Rital and Cherifi, 2004).

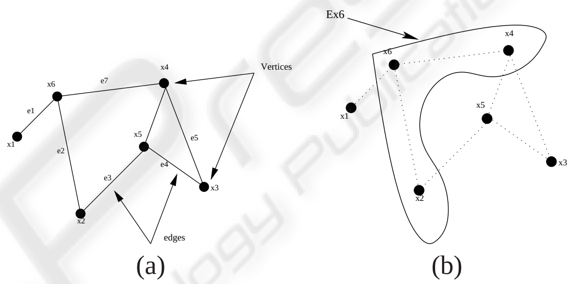


Figure 1: An example of (a) graph and (b) hypergraph.

Also, like graphs, hypergraphs may be partitioned such that a cut metric is minimized. The most extensive and large scale use of hypergraph partitioning algorithms, however, occurs in the field of VLSI design and synthesis. A typical application involves the partitioning of large circuits into k equally sized parts in a manner that minimizes the connectivity between the parts. The circuit elements are the vertices of the hypergraph and the nets that connect these circuit elements are the hyperedges (Alpert and Kahng, 1995). The leading tools for partitioning these hypergraphs are based on two phase multi-level approaches (Karypis et al., 1999). In the first phase, they construct a hierarchy of hypergraphs by incrementally collapsing the hyperedges of the original hypergraph according to some measure of homogeneity. In the second phase, starting from a partitioning of the hypergraph at the coarsest level, the algorithm works its way down the hierarchy and at each stage the partitioning at the level above serves as an initialization for a vertex swap based heuristic that refines the partitioning greedily (Fiduccia and Mattheyses, 1982),(Kernighan and Lin., 1970). The development of these tools is almost entirely heuristic and very little theoretical work exists that analyzes their performance beyond empirical benchmarks. Hypergraph cut metrics provide a more accurate model than graph partitioning in many cases of practical interest. For example, in the row-wise decomposition of a sparse matrix for parallel matrix-vector multiplication, a hypergraph model provides an exact measure of communication cost, whereas a graph model can only provide an upper bound (Trifunovic and Knottenbelt, 2004a) (Catalyurek and Aykanat., 1999). It has been shown that, in general, there does not exist a graph model that correctly represents the cut properties of the corresponding hypergraph (Ihler et al., 1993). Recently, several serial and parallel hypergraph partitioning techniques have been extensively studied (Sanchis, 1989) (Trifunovic and Knottenbelt, 2004a)(Karypis, 2002) and tools support exists (e.g. hMETIS (Karypis and Kumar, 1998), PaToH (Catalyurek and Aykanat., 1999) and Parkway (Trifunovic and Knottenbelt, 2004b)). These partitioning techniques showed a very great efficiency in distributed databases and VLSI circuits fields.

In this paper, we widen the application area of hypergraph partitioning algorithms to image fields and more particularly to the image segmentation. The basic idea of this algorithm can be described as follows and summarize in two steps:

1. It first builds a hypergraph of the image.
2. Then the algorithm partitions this representation into a set of vertices, representing homogeneous regions.

The aim of the first step is to capture all global and local properties of the image data and the whole key information for the segmentation purpose. This model has proved to be extremely useful for solving some applications in image processing fields such as noise removal (Rital et al., 2001) and edge detection (Rital and Cherifi, 2004). While the second step of the proposed approach partition this representation to a homogenous regions. It is done by a fast multilevel programming algorithm. Throughout this paper, we will denote the hypergraph of the image by the Image Neighborhood Hypergraph INH.

In section 2, we briefly review some background on hypergraph theory. The proposed segmentation approach is presented in Section 3 and its performance is illustrated in Section 4. The paper ends with a conclusion in Section 5.

2 BACKGROUND

Our main interest in this paper is to use combinatorial models. We will introduce basic tools that are needed. A hypergraph H on a set X is a family $(E_i)_{i\in I}$ of non-empty subsets of X called *hyperedges* with:

$$\bigcup_{i\in I} E_i = X, \quad I = \{1, 2, \ldots, n\}, \; n \in \mathbb{N}.$$

Given a graph $G(X;e)$, the hypergraph having the vertices of G as vertices and the neighborhood of these vertices as hyperedges (including these vertices) is called the *neighborhood hypergraph* of G. To each graph we can associate a neighborhood hypergraph :

$$H_G = (X, (E_x = \{x\} \cup \Gamma(x))) \quad (4)$$
$$where \;\; \Gamma(x) = \{y \in X, (x,y) \in e\}.$$

2.1 Multilevel Hypergraph Partitioning

The goal of the k-way hypergraph partitioning problem is to partition the vertices of the hypergraph into k disjoint subsets X_i, ($i = 0, \ldots, k-1$), such that a certain objective functions defined over the hyperedges is optimized.

Let us note $H(X,E)$ a hypergraph. We will assume that each vertex and hyperedge has a weight associated with it, and we will use $w(x)$ to denote the weight of a vertex x, and $w(E)$ to denote the weight of a hyperedge E. One of the most commonly used objective functions is to minimize the hyperedge-cut of the partitioning; i.e., the sum of the weights of the hyperedges that span multiple partitions: $cut\{A,B\} = \sum_{E_i\in A, E_j\in B} w(E_i,E_j)$, A, B are two partitions. Another objective that is often used is to minimize the sum of external degrees (SOED) of all hyperedges that span multiple partitions (Karypis et al., 1999).

The most commonly used approach for computing a k-way partitioning is based on recursive bisection. In this approach, the overall k-way partitioning is obtained by initially bisecting the hypergraph to obtain a two-way partitioning. Then, each of these parts is further bisected to obtain a four-way partitioning, and so on. The problem of computing an optimal bisection of a hypergraph is at least NP-hard (Garey and Johnson, 1979); however, many heuristic algorithms have been developed. The survey by Alpert and Kahng (Alpert and Kahng, 1995) provides a detailed description and comparison of various such schemes.

The key idea behind the multilevel approach for hypergraph partitioning is fairly simple and straightforward. Multilevel partitioning algorithm, instead of trying to compute the partitioning directly in the original hypergraph, partition the hypergraph using three process (Fig.2):

Coarsening phase: first obtain a sequence of successive approximations of the original hypergraph. Each one of these approximations represents a problem whose size is smaller than the size of the original hypergraph. This process continues until a level of approximation is reached in which the hypergraph contains only a few tens of vertices (Fig. 3).

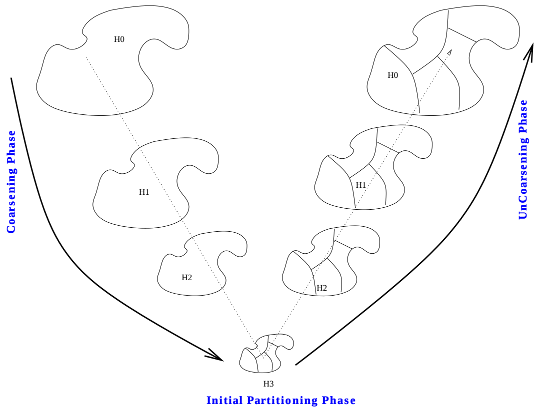


Figure 2: Multilevel Hypergraph Partitioning.

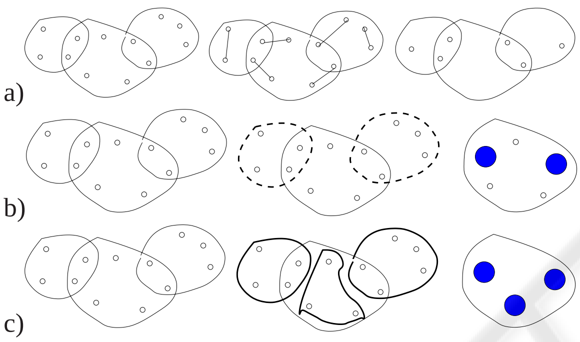


Figure 3: Some coarsening schemes: (a)Edge Coarsening, (b) Hyperedge Coarsening, and (c) Modified Hyperedge Coarsening.

Initial partitioning phase: At this point, these algorithms compute a partitioning of that hypergraph. Since the size of this hypergraph is quite small, even simple algorithms such as Kernighan-Lin (KL) (Kernighan and Lin., 1970) or Fiduccia-Mattheyses (FM) (Fiduccia and Mattheyses, 1982) lead to reasonably good solutions. ***Uncoarsening phase***: final step of these algorithms is to take the partitioning computed at the smallest hypergraph and use it to derive a partitioning of the original hypergraph. This is usually done by propagating the solution through the successive better approximations of the hypergraph and using simple approaches to further refine the solution.

2.2 Image and Neighborhood Relations

In this paper, the image will be represented by the following mapping : $I : X \subseteq \mathbb{Z}^2 \longrightarrow C \subseteq \mathbb{Z}^n$. Vertices of X are called pixels, elements of C are called colors. A distance d on X defines a grid (a connected, regular graph , without both loop and multi-edge). Let d' be a distance on C, we have a neighborhood relation on an image defined by :

$$\forall x \in X, \Gamma_{\lambda,\beta}(x) = \{x' \in X, x' \neq x \mid d'(I(x), I(x') \leq \lambda \ and \ d(x, x') \leq \beta) \quad (5)$$

The neighborhood of x on the grid will be denoted by $\Gamma_{\lambda,\beta}(x)$. To each image we can associate a hypergraph called *Image Neighborhood Hypergraph* (INH) (Rital and Cherifi, 2004):

$$H_{\Gamma_{\lambda,\beta}} = (X, (\{x\} \cup \Gamma_{\lambda,\beta}(x))_{x \in X}). \quad (6)$$

On a grid Γ_β, to each pixel x we can associate a neighborhood $\Gamma_{\lambda,\beta}(x)$, according to a predicate λ. The predicate λ may be completely arbitrary, it is useful for a task domain. It may be defined on a set of points, it may use colors, or some symbolic representation of a set of colors, or it may be a combination of several predicates, etc.

From $H_{\Gamma_{\lambda,\beta}}$, we define a weighted image neighborhood hypergraph (WINH) according to the two maps functions f_{w_v} and f_{w_h}. The first map f_{w_v}, associates an integer weight w_{xi} with every vertex $x_i \in X$. The weight is defined by the color in each pixel. The map function f_{w_h} associates to each hyperedge a weight w_{hi} defined by the mean color in hyperedge. The WINH is defined by :

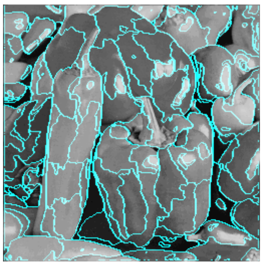
$$H_{\Gamma_{\lambda,\beta}} = (X, E_{\lambda,\beta}, w_v, w_h), \quad (7)$$
$$\forall x \in X, \ \ f_{w_v}(x) = I(x)$$

$$\forall E(x) \in E_{\lambda,\beta}, \ \ f_{w_h}(E(x)) = \frac{1}{|E(x)|} \sum_{i=1}^{|E(x)|} I(x_i)_{x_i \in E(x)} \quad (8)$$

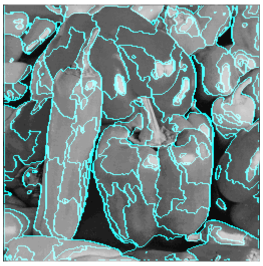
3 SEGMENTATION ALGORITHM

In this section, we describe a segmentation algorithm based on image neighborhood hypergraph representation and multilevel hypergraph partitioning method. It starts with a WINH generation followed by a multilevel hypergraph partitioning. The steps of the algorithm are described below :

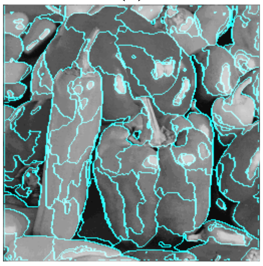
1. Input : Image, thresholds λ, β and μ (the number of regions).
2. Weighted Image Neighborhood Hypergraph (WINH) generation.
3. Multilevel weighted image neighborhood hypergraph partitioning
   (a) the coarsening phase
   (b) the initial partitioning
   (c) the uncoarsening phase
4. Output : segmented Image.



(a)



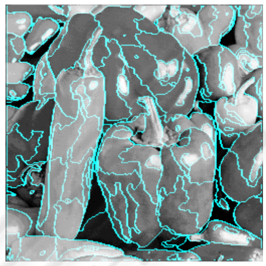
(b)



(c)

Figure 4: The output of the proposed algorithm with WINH (a) using weighted vertices only, (b) using weighted hyperedge only, (c) using weighted both vertices and hyperedges. The parameters algorithm :$\beta = 1$, $\lambda = 15$ and $\mu = 51$.

(a)



(b) $\mu = 51$, $\lambda = 15$, $\beta = 1$ and computing time = 32.23s

(c) $\mu = 47$ and computing time = 402.75s

Figure 5: A comparison between the proposed algorithm and normalized cut. (a) the original image. (b) the output of the proposed algorithm. (c) the output of normalized cut algorithm.

4 EXPERIMENTAL RESULTS

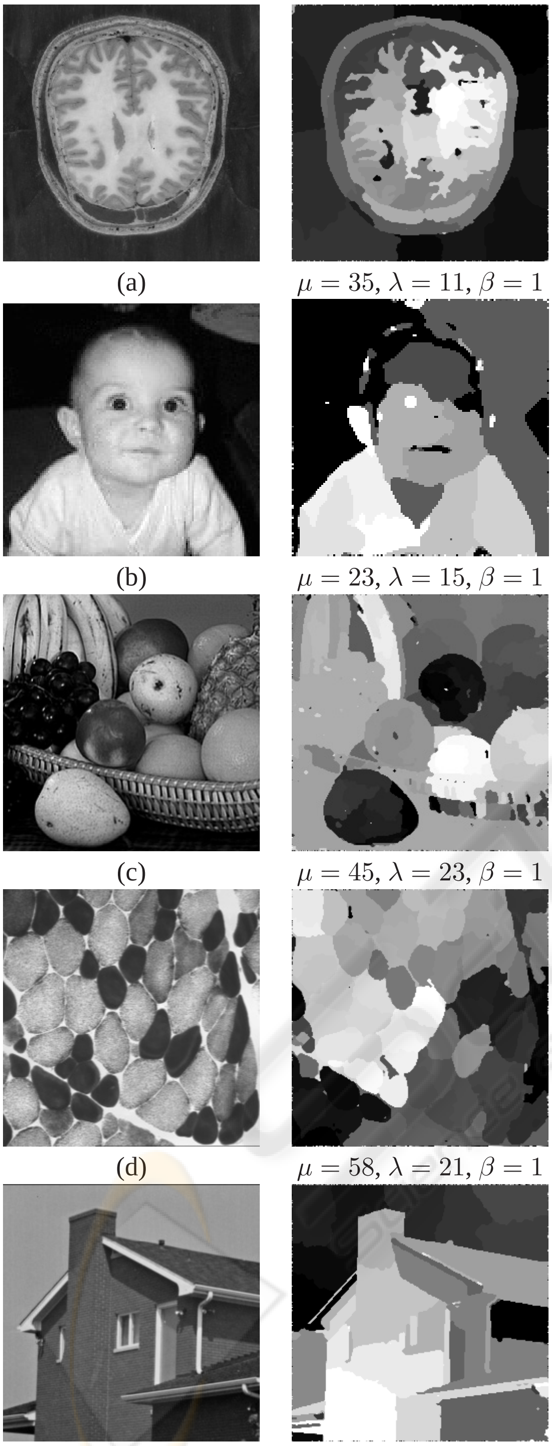
A group of a gray scale images with different homogenous areas were chosen in order to demonstrate the performances of the proposed algorithm. First, we build the WINH. The values of β, λ and μ are adjusted in experiments. The values posted thereafter corresponds to the best results. In the coarsening phase of the second part of the proposed algorithm, we use the hyperedge coarsening method (Fig. 3).

During the initial partitioning phase, a bisection of the coarsened image neighborhood hypergraph is computed. We use multiple random bisections, followed by the Fiduccia-Mattheyses(FM) refinement algorithm. In the last phase (uncoarsening), the partitioning is done by successively projecting the partitioning to the next level finer WINH and using a partitioning refinement algorithm to reduce the cut and thus to improve the quality of the partitioning. For this phase, we use the refinement algorithm integrated in HMETIS package (Karypis and Kumar, 1998).

We first evaluate the performance of the proposed algorithm using WINH. In this experiment, we want to know the best WINH representation allowing to improve the next stage of algorithm (hypergraph partitioning) and consequently the segmentation approach. We implement the algorithm with three types of WINH : (1) using weighted vertices only, (2) using weighted hyperedges only and (3) using weighted both vertices and hyperedges. Figure 4 shows the results of the proposed algorithm using these three types of WINH. We can see that the last representation WINH (using both weighted vertices and weighted hyperedges) gives significant results; especially in the image areas containing many information. Indeed, the third WINH gives more information about the image to neighborhood hypergraph partitioning.

In order to compare our method with an existing one, we have chosen the technique of Malik et al. (Shi and Malik, 2000). We have processed a group of images with our segmentation method and compared the results to normalized cuts. Normalized cuts used the same parameters for all images, namely, the optimal parameters given by authors.

Figure 5 shows a comparison between our algorithm and normalized cut on Peppers image. According to the segmentation results on this image, we note that the proposed algorithm localize better the areas of



(a) $\mu = 35$, $\lambda = 11$, $\beta = 1$

(b) $\mu = 23$, $\lambda = 15$, $\beta = 1$

(c) $\mu = 45$, $\lambda = 23$, $\beta = 1$

(d) $\mu = 58$, $\lambda = 21$, $\beta = 1$

(e) $\mu = 27$, $\lambda = 15$, $\beta = 1$

Figure 6: The outputs of the proposed algorithm on other images. (a),(b),(c) and (d) the original images.

the treated image that the normalized cut algorithm. Figure 6 shows the results of the proposed algorithm on other images.

The strength of our algorithm is that it better detects the regions containing many details. In addition, our algorithm is powerful in computing times. It is ten times inferior comparing to normalized cuts algorithm.

5 CONCLUSIONS

We have presented a weighted image neighborhood hypergraph partitioning for image segmentation. The segmentation is accomplished in two stages. In the first stage, weighted image neighborhood hypergraph is generated. In the second stage, hypergraph partitioning method using HMETIS package is computed. Experimental results demonstrate that our approach performs better than Normalized cut algorithm. Our algorithm represents the first proposition for solving the image segmentation problem. It can be improved in several ways (parameters : the function maps, the colorimetric threshold, the unsupervised region number, etc.).

REFERENCES

Alpert, C. J. and Kahng, A. B. (1995). Recent developments in netlist partitioning: A survey. *Integration: the VLSI Journal*, 19(1-2):1–81.

Catalyurek, U. and Aykanat., C. (1999). Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication. *IEEE Transactions on Parallel and Distributed Systems*, 10(7):673–693.

Cox, I., Rao, S., and Zhong, Y. (1996). Ratio regions: A technique for image segmentation. In *13th International Conference on Pattern Recognition (ICPR'96)*.

Fan, J., Yau, D. K., Elmagarmid, A. K., and Aref, W. G. (2001). Automatic image segmentation by integrating color edge detection and seeded region growing. *IEEE Trans. on Image Processing.*, 10(10):1454–1466.

Fiduccia, C. M. and Mattheyses, R. M. (1982). A linear-time heuristic for improving network partitions. In *Proceedings of the 19th ACM/IEEE Design Automation Conference DAC 82*, pages 175–181.

Garey, M. and Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co.

Gdalyahu, Y., Weinshall, D., and Werman, M. (2001). Self-organization in vision : stochastic clustering for image segmentation, perceptual grouping, and image database organization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(10):1053–1074.

Hadley, S., Mark, B., and Vannelli, A. (1992). An efficient eigenvector approach for finding netlist partitions. *IEEE Trans. CAD*, 11:885–892.

Ihler, E., Wagner, D., and Wagner, F. (1993). Modeling hypergraphs by graphs with the same mincut properties. *Inf. Process. Lett.*, 45(4):171–175.

Karypis, G. (2002). Multilevel hypergraph partitioning. Technical report #02-25, University of Minnesota.

Karypis, G., Aggarwal, R., Kumar, V., and Shekhar, S. (1999). Multilevel hypergraph partitioning: applications in vlsi domain. *IEEE Trans. Very Large Scale Integr. Syst.*, 7(1):69–79.

Karypis, G. and Kumar, V. (1998). hmetis 1.5: A hypergraph partitioning package. *Technical report, University of Minnesota, Available onhttp://www-users.cs.umn.edu/∼karypis/metis/hmetis/index.html*.

Kernighan, B. W. and Lin., S. (1970). An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, 49(1):291–307.

Koontz, W. and Fukunaga, K. (1972). A nonparametric valley-seeking technique for cluster analysis. *IEEE Trans. Comput.*, 21:171–178.

Koyutrk, M. and Aykanat, C. (2005). Iterative-improvement-based declustering heuristics for multidisk databases. *Information Systems*, 30(1):47–70.

Martinez, A. M., Mittrapiyanuruk, P., and Kak, A. C. (2004). On combining graph-partitioning with non-parametric clustering for image segmentation. *Computer Vision and Image Understanding*, 95:72–85.

Navon, E., Miller, O., and Averbuch, A. (2005). Color image segmentation based on adaptive local thresholds. *Image Vision Comput.*, 23(1):69–85.

Pal, N. and Pal, S. (1993). A review on image segmentation techniques. *Pattern Recognition*, 26:1277–1294.

Rital, S., Bretto, A., Aboutajdine, D., and Cherifi, H. (2001). Application of adaptive hypergraph model to impulsive noise detection. *Lecture Notes in Computer Science*, 2124:555–562.

Rital, S. and Cherifi, H. (2004). A combinatorial color edge detector. *Lecture Notes in Computer Science*, 3212:289–297.

Sanchis, L. A. (1989). Multiple-way network partitioning. *IEEE Transactions on Computers*, pages 62–81.

Sarkar, S. and Boyer, K. (1996). Quantitative measures of change based on feature organization: eigenvalues and eigenvectors. *Proc. IEEE Conf. Comput. Vis. Patt. Recogn.*, pages 478–483.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intellignece*, 22(8):888–905.

Soundararajan, P. and Sarkar, S. (2001). Analysis of mincut, average cut, and normalized cut measures. *Proc. Third Workshop Perceptual Organization in Computer Vision*.

Soundararajan, P. and Sarkar, S. (2003). An in-depth study of graph partitioning measures for perceptual organization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(6):642–660.

Tal, D. and Malik, J. (2001). Combining color, texture and contour cues for image segmentation. *Preprint*.

Trifunovic, A. and Knottenbelt, W. (2004a). A parallel algorithm for multilevel k-way hypergraph partitioning. In *Proceedings of 3rd International Symposium on Parallel and Distributed Computing*.

Trifunovic, A. and Knottenbelt, W. (2004b). Parkway 2.0: A parallel multilevel hypergraph partitioning tool. In *Proceedings of 19th International Symposium on Computer and Information Sciences (ISCIS 2004)*, volume 3280, pages 789–800.

Wang, S. and Siskind, J. M. (2003). Image segmentation with ratio cut - supplemental material. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(6):675–690.

Wu, Z. and Leahy, R. (1993). An optimal graph theoretical approach to data clustering: theory and its application to image segmentation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 15:1101–1113.