

# Weighted Adaptive Neighborhood Hypergraph Partitioning for Image Segmentation

Soufiane RITAL<sup>1</sup>, Hocine CHERIFI<sup>2</sup>, and Serge MIGUET<sup>1</sup>

<sup>1</sup> LIRIS CNRS, Lyon II University, Lyon, France

{Soufiane.Rital,Serge.Miguet}@liris.cnrs.fr,

<sup>2</sup> LIRSIA, University of Bourgogne, Dijon, France.

Hocine.Cherifi@u-bourgogne.fr

**Abstract.** The aim of this paper is to present an improvement of a previously published algorithm. The proposed approach is performed in two steps. In the first step, we generate the Weighted Adaptive Neighborhood Hypergraph (WAINH) of the given gray-scale image. In the second step, we partition the WAINH using a multilevel hypergraph partitioning technique. To evaluate the algorithm performances, experiments were carried out on medical and natural images. The results show that the proposed segmentation approach is more accurate than the graph based segmentation algorithm using normalized cut criteria.

**Key words:** hypergraph, neighborhood hypergraph, hypergraph partitioning, image segmentation, edge detection and adaptive thresholding.

## 1 Introduction

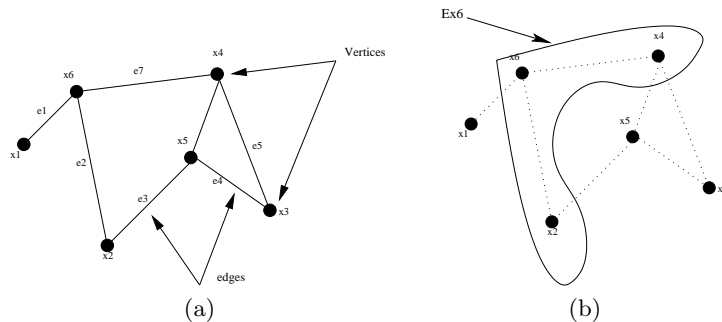
Image segmentation is an important step in computer vision. Several algorithms have been introduced to tackle this problem. Among them are approaches based on graph partitioning [1–6].

The graph approaches carry the appeal of a strong theoretical basis and the advantage of being applicable not only to the segmentation of images, but also to other low, mid, and high level vision tasks. For grouping pixels into regions with a graph-theoretic approach, a graph is usually defined as  $G(X, e)$ , where the nodes  $X$  represent the pixels (one node per pixel) and the edges  $e$  represent the weights  $w(i, j)$  that connect pairs of nodes.

One of the most frequently used techniques to partition a graph is by means of the cut cost function. The goal of the cut algorithm is to find two sub-graphs  $A$  and  $B$  of  $G$  that minimize the value of  $cut\{A, B\} = \sum_{i \in A, j \in B} w(i, j)$  and with the obvious constraints  $A \cup B = X$ ,  $A \cap B = \emptyset$ , and  $A \neq \emptyset, B \neq \emptyset$ .

Several alternatives to the above criterion have been proposed to date [4, 1, 3]. Of particular note is the normalized cut criterion (Ncut) of Shi and Malik [1], which attempts to rectify the tendency of the cut algorithm to favor isolated nodes of the graph.

A hypergraph is an extension of a graph in which edges are allowed to connect arbitrary, non-empty sets of vertices (as shown in Fig. 1). Similarly to graphs, hypergraphs can be used to represent the structure of many applications, such



**Fig. 1.** An example graph and a hypergraph.

as data dependencies in distributed databases, component connectivity in VLSI circuits and image analysis [7–10]. Also, like graphs, hypergraphs may be partitioned such that a cut metric is minimized. However, 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 [11, 12]. It has been shown that, in general, there does not exist a graph model that correctly represents the cut properties of the corresponding hypergraph [13]. Recently, several serial and parallel hypergraph partitioning techniques have been extensively studied [14, 15, 12] and tools support exists (e.g. hMETIS [16], PaToH [11] and Parkway [17]). These partitioning techniques showed a very great efficiency in distributed databases and VLSI circuits fields.

In this paper, we present a new hypergraph-based image segmentation algorithm using hypergraph partitioning techniques. The basic idea of this algorithm can be described as follows. It first builds a weighted hypergraph of the given gray-scale image. Then the algorithm partitions this representation into a set of vertices, representing homogeneous regions. The hypergraph partitioning is done by a fast multilevel programming algorithm. Our contribution consists in presenting an Adaptive Image Neighborhood Hypergraph representation (WAINH). The WAINH is the most significant step in the segmentation algorithm, because it makes it possible to connect the given gray-scale image and the existing hypergraph partitioning techniques.

The adaptive representation captures the local properties of the gray-scale image and the whole key information for the segmentation purpose. This leads to a new hypergraph-based technique which is more relevant to image segmentation than our previous work [9].

The remainder of this paper is organized as follows: in section 2, we introduce the weighted adaptive image neighborhood hypergraph. The hypergraph partitioning for image segmentation is introduced in section 3. In section 4, we

illustrate the performances of the proposed approach. The paper ends with a conclusions and perspectives in section 5.

## 2 Adaptive Image Neighborhood Hypergraph (AINH)

A hypergraph is a pair  $H = (X, E)$ , where  $X = x_1, x_2, \dots, x_n$  is the set of vertices (or nodes) and  $E = E_1, E_2, \dots, E_m$ , with  $E_i \subseteq X$  for  $i = 1, \dots, m$ , is the set of hyperedges.

Let us note  $G(X; e)$  a graph and  $H(X, E)$  a hypergraph. 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 graph  $G$ . To each graph  $G$  we can associate a neighborhood hypergraph (figure 1):

$$H_G = (X, (E_x = \{x\} \cup \Gamma(x))) \text{ where } \Gamma(x) = \{y \in X, (x, y) \in e\} \quad (1)$$

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:

$$\Gamma_{\lambda, \beta}(x) = \{x' \in X, |d'(I(x), I(x')) \leq \lambda \text{ and } d(x, x') \leq \beta\}. \quad (2)$$

The neighborhood of  $x$  on the grid will be denoted by  $\Gamma_{\lambda, \beta}(x)$ . To each image  $I$ , we can associate a hypergraph called *Image Neighborhood Hypergraph*(INH) [8] :

$$H_{\Gamma_{\lambda, \beta}}(I) = (X, (\{x\} \cup \Gamma_{\lambda, \beta}(x))_{x \in X}) \quad (3)$$

The figure 2 illustrates an example of image neighborhood hypergraph representation.

On a grid  $\Gamma_{\beta}$ , to each pixel  $x$  we can associate a neighborhood  $\Gamma_{\lambda, \beta}(x)$ , according to a predicate  $\lambda$ . The threshold  $\lambda$  can be carried out in two ways. In the first way, the  $\lambda$  is given for all the pixels of the image. In the second way, the  $\lambda$  is generated locally and applied in an adaptive way to the image  $I$ .

In this paper, the attribute  $\lambda$  is computed in an adaptive way depending on the local properties of the image. The value of  $\lambda$  will be estimated by :

$$\lambda = \text{Median} \{I(y) - \text{Median}(F(x))\}_{\forall y \in F} \quad (4)$$

$F$  is the window centered in  $x$  with the size  $[2\beta + 1 \times 2\beta + 1]$ .

This  $H_{\Gamma_{\lambda, \beta}}(I)$  combinatorial representation is more relevant than the previous one introduced in [9], because it takes into account both the local and global aspects of the image. Hence  $H_{\Gamma_{\lambda, \beta}}(I)$  offers new facilities for handling the topology and the geometry of the image. Consequently, it gives more information about the nature of the image to analyze.

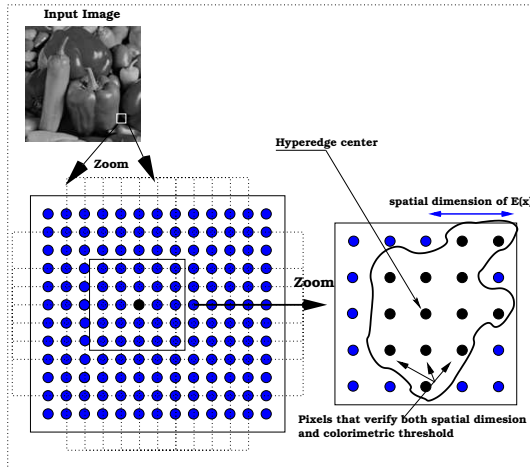


Fig. 2. An example of INH representation.

### 3 Multilevel WAINH Partitioning

From  $H_{\Gamma_{\lambda,\beta}}(I)$ , we define a Weighted Adaptive Image Neighborhood Hypergraph (WAINH) according to the two maps functions  $f_{w_v}$  and  $f_{w_h}$ . The first map  $f_{w_v}$  associates an integer weight  $w_{x_i}$  with every vertex  $x_i \in X$ . The weight is defined by the intensity in each pixel. The map function  $f_{w_h}$  associates to each hyperedge a weight  $w_{h_i}$  defined by the mean intensity in this hyperedge. The WAINH is defined by:

$$H_{\lambda,\beta} = (X, E_{\lambda,\beta}, w_x, w_h), \forall x \in X, f_{w_x}(x) = I(x)$$

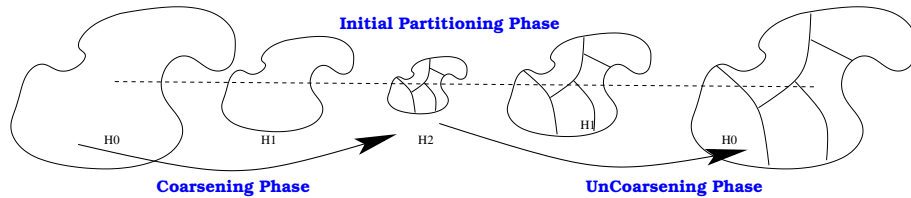
$$\forall E(x) \in E_{\lambda,\beta}, f_{w_h}(E(x)) = \frac{1}{|E(x)|} \sum_{i=1}^{|E(x)|} I(x_i)_{x_i \in E(x)}$$

The formal definition of the  $k$ -way hypergraph partitioning technique is as follows : find  $k$  disjoint subsets  $X_i$ , ( $i = 0, \dots, k-1$ ) of the vertex set  $X$  with part (region) weights  $W_i$  ( $i = 0, \dots, k-1$ )(given by the sum of the constituent vertex weights), such that, given a prescribed balance criterion  $0 < \epsilon < 1$ ,  $W_i < (1 + \epsilon)W_{avg}$  holds  $\forall i = 0, \dots, k-1$  and an objective function over the hyperedges is minimized. The  $W_{avg}$  denotes the average part weight.

If the objective function is the hyperedge cut metric, then the partition cost (or cut-size) is given by the sum of the costs of hyperedges that span more than one part. Alternatively, when the objective function is the  $(k-1)$  metric, the partition cost is given by :  $P_{cost} = \sum_{i=0}^{|E|-1} (\gamma_i - 1)w_{h_i}$

Computing the optimal bisection of a hypergraph under the hyperedge cut metric (and hence the  $(k-1)$  metric since  $k = 2$  for a bisection) is known to be NP-complete [18]. Thus, researches have focused on developing polynomial

time heuristic algorithms resulting in good sub-optimal solutions. Because it scales well in terms of run time and solution quality with increasing problem size, the multilevel paradigm is preferred to direct solution approaches. Below, we describe the main steps of the multilevel paradigm (figure. 3).

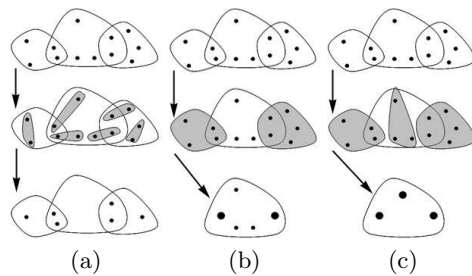


**Fig. 3.** Multilevel Hypergraph Partitioning.

**i. Coarsening phase:**  $H_{\lambda,\beta}$  is approximated via a succession of smaller hypergraphs that maintain its structure as accurately as possible. A single coarsening step is performed by merging the vertices of the original hypergraph together to form vertices of the coarse hypergraph, denoted by a map  $f_{merge} : X \rightarrow X_{coarse}$ , where

$$\frac{|X|}{|X_{coarse}|} = r, \quad r > 1, \quad (5)$$

and  $r$  is the prescribed reduction ratio. The map  $f_{merge}$  is used to transform the hyperedges of the original hypergraph  $H_{\lambda,\beta}$  to the hyperedges of the coarse hypergraph. Single vertex hyperedges in the coarse hypergraph are discarded as they cannot contribute to the cut-size of a partition of the coarse hypergraph. Several  $f_{merge}$  maps functions have been proposed [19] (figure 4): edge coarsening, hyperedge coarsening and modified hyperedge coarsening.

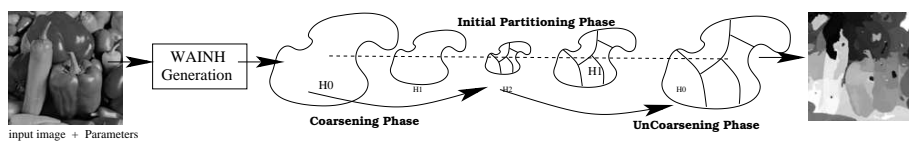


**Fig. 4.** Coarsening phase : (a) Edge coarsening : connected pairs of vertices are matched together. (b) Hyperedge coarsening : all the vertices belonging to a hyperedge are matched together. (c) Modified hyperedge coarsening : we match together all the vertices in a hyperedge as well as all the groups of vertices belonging to a hyperedge.

**ii. Initial partitioning phase:** During the initial partitioning phase, a partitioning of the coarsest hypergraph  $H_{\lambda,\beta}^{coarse}$  is computed, such that it minimizes the cut. Since this hypergraph has a very small number of vertices, the time to find a partitioning using any of the heuristic algorithms tends to be small.

**iii. Uncoarsening phase:** During the uncoarsening phase, a partitioning of the coarser hypergraph is successively projected to the next level finer hypergraph, and a partitioning refinement algorithm is used to reduce the cut-set (and thus to improve the quality of the partitioning). Since the next level finer hypergraph has more degrees of freedom, such refinement algorithms tend to improve the solution quality.

Figure 5 illustrates the proposed algorithm. It starts with a WAINH generation followed by a multilevel hypergraph partitioning.



**Fig. 5.** The two steps of the proposed segmentation algorithm. The input parameters are : intensity threshold  $\lambda$ , spatial threshold  $\beta$  and the number of desired regions  $k$ .

## 4 Experimental Results

A set of gray-scale images with different homogeneous areas was chosen in order to demonstrate the performances of our algorithm. The simulations are grouped in two parts. Firstly, the evaluation of the algorithm according to the WAINH representation, then the evaluation of the proposed algorithm compared to the existing methods.

We will first describe the various stages of implementation of the proposed algorithm.

1. For WAINH generation, we use an adaptive threshold  $\lambda$ . It is estimated using Equation (4) while the parameters value  $(\beta, k)$  are adjusted in experiments.
2. For WAINH partitioning, and in the coarsening phase, we use the hyper-edge coarsening approach. In the initial partitioning phase, we compute the  $k$ -way partitioning of the coarsest hypergraph using the multilevel hypergraph bisection algorithm [15]. In the uncoarsening phase, we use the F.M. refinement algorithm [14].

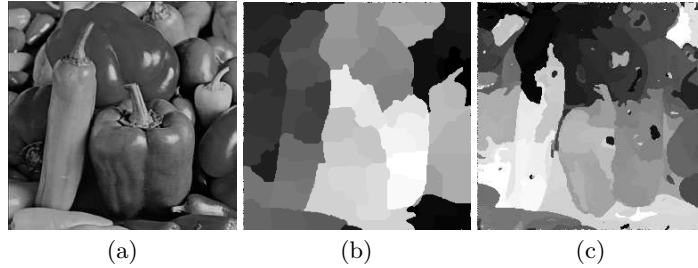
For the coarsening, initial partitioning and uncoarsening phases we use the Hmetis package [16].

We will now show the effect of the weighted hypergraph generation on the quality of the image segmentation results. For this study, we implement two

weighted neighborhood hypergraph representations : the WAINH representation defined in section 3, and the WINH representation used in our previous work [9].

The WINH representation uses a global threshold  $\lambda$ . This means that all the hyperedges  $E_{\lambda,\beta}(x)$  are generated with the same threshold  $\lambda$ .

Figure 6 shows the segmentation results of Peppers image obtained, using WAINH and WINH representations. From this figure, we note that the proposed algorithm using WINH representation tends to divide large constant areas into multiple segments. The reader might see that we could improve the results by using WAINH representation. Indeed, the use of WAINH representation in the proposed algorithm involves the detection of more significant regions with high precision. Consequently, weighted hypergraph representation influences segmentation quality. The improvement of this representation leads to an improvement of the segmentation quality. These remarks can be observed on each image of the set.



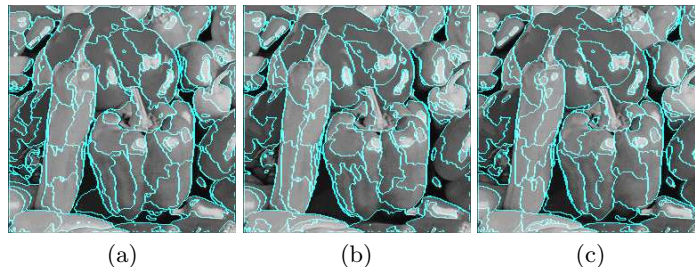
**Fig. 6.** WAINH and WINH comparison. (a) Peppers image. (b,c) Outputs of the proposed algorithm using WINH ( $\lambda = 20$ ,  $\beta = 1$  and  $k = 51$ ) and WAINH ( $\beta = 1$  and  $k = 51$ ) representations respectively.

We will now evaluate the WAINH representation according to the mapping function  $f_{w_v}$  and  $f_{w_h}$ . Independently of these two functions, we can generate three representations. (1) Using weighted vertices only, (2) using weighted hyperedges only and (3) using both weighted vertices and weighted hyperedges.

The goal of this study is to evaluate the proposed algorithm according to these three representations and more particularly to the quantity of information contained in these representations.

Figure 7 shows the results of the proposed algorithm using these three WAINH representations. From this figure, we note that the algorithm with these three representations gives comparable results. But, in some areas containing more details and more useful information, we can see that the last representation WAINH (using both weighted vertices and weighted hyperedges) gives significant results. Indeed, the third WAINH gives more key information about the image used in

multilevel neighborhood hypergraph partitioning technique for the segmentation purpose.



**Fig. 7.** The output of the proposed algorithm with WAINH (a) using weighted vertices only, (b) using weighted hyperedge only, (c) using both weighted vertices and weighted hyperedges. The parameters of the algorithm:  $\beta = 1$  and  $\mu = 51$ .

To evaluate the WAINH representation used in our segmentation approach, the discussion so far used only one Peppers image. In this study, we show several additional results on different types of images. These additional results are compared to the results of Shi and Malik segmentation algorithm [1] (Normalized Cuts detection Ncut). This algorithm use the same parameters for all images, namely, the optimal parameters given by the authors.

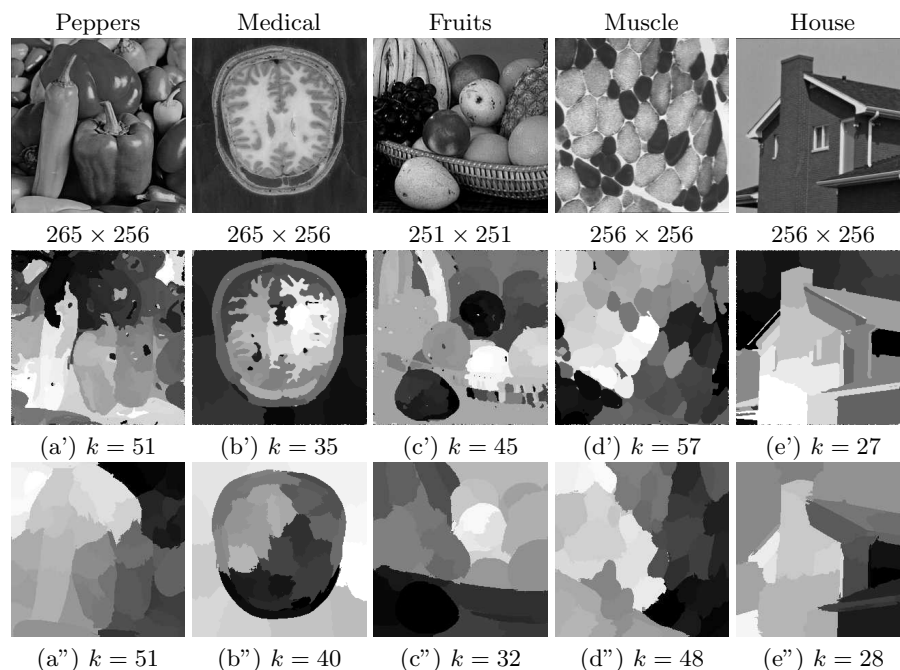
Figure 8 shows a comparison between the proposed and Ncut algorithms on Peppers, Medical, Fruits, Muscle and House images. According to the segmentation results on these images, we note that our algorithm make a better localization of the regions in the processed image compared to the Ncut method. The strength of this algorithm is that it better detects the regions containing many details.

In addition, it results in shorter computing times faster than normalized cuts algorithm. The table 1 describes the computing times of these two algorithms. They have been implemented using C++ language in a notebook with the following characteristics: Pentium Centrino, 1,5GHz, 512 Mo RAM.

**Table 1.** The computing times of the proposed and Malik et al. algorithms for Peppers, Medical, Fruits, Muscle and House images.

Image	proposed (in second)	Ncut (in second)
Peppers	3.22	402.75
Medical	2.90	463.64
Fruits	3.25	447.78
House	2.69	453.62
Muscle	3.78	477.28





**Fig. 8.** A comparison between proposed and normalized cut algorithms. (a',b',c',d',e') The outputs of the proposed algorithm with  $\beta = 1$ . (a'',b'',c'',d'',e'') The output of normalized cut algorithm.

## 5 Conclusions and Perspectives

We have presented a weighted adaptive image neighborhood hypergraph partitioning for image segmentation. The segmentation is accomplished in two stages. In the first stage, a weighted adaptive image neighborhood hypergraph is generated. In the second stage, a hypergraph partitioning is computed using a multilevel technique is computed. Experimental results demonstrate that our approach using the weighted adaptive neighborhood hypergraph performs better than the same algorithm using a global representation and the normalized cut algorithm. Currently, we work on the extension of the proposed algorithm on color images.

## References

1. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22** (2000)
2. Soundararajan, P., Sarkar, S.: An in-depth study of graph partitioning measures for perceptual organization. *IEEE Trans. Pattern Anal. Mach. Intell.* **25** (2003) 642–660

3. Martinez, A.M., Mittrapiyanuruk, P., Kak, A.C.: On combining graph-partitioning with non-parametric clustering for image segmentation. *Computer Vision and Image Understanding* **95** (2004) 72–85
4. Wang, S., Siskind, J.M.: Image segmentation with ratio cut - supplemental material. *IEEE Trans. Pattern Anal. Mach. Intell.* **25** (2003)
5. Soundararajan, P., Sarkar, S.: Analysis of mincut, average cut, and normalized cut measures. *Proc. Third Workshop Perceptual Organization in Computer Vision* (2001)
6. Timothee Cour, Florence Benezit, J.S.: Spectral segmentation with multiscale graph decomposition. *IEEE International Conference on Computer Vision and Pattern Recognition* (to appear) (2005)
7. Rital, S., Bretto, A., Aboutajdine, D., Cherifi, H.: Application of adaptive hypergraph model to impulsive noise detection. *Lecture Notes in Computer Science* **2124** (2001) 555–562
8. Rital, S., Cherifi, H.: A combinatorial color edge detector. *Lecture Notes in Computer Science* **3212** (2004) 289–297
9. Rital, S., Cherifi, H., Miguët, S.: Neighborhood hypergraph partitioning for image segmentation. *First International Conference on Pattern Recognition and Machine Intelligence* (18-22, 2005)
10. Bretto, A., J.Azema, Cherifi, H., Laget, B.: Combinatorics and image processing. *Graphical Models and Image Processing* **5** (1997) 265–372
11. Catalyurek, U., Aykanat, C.: Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication. *IEEE Trans. Parallel Distrib. Syst.* **10** (1999) 673–693
12. Trifunovic, A., Knottenbelt, W.: A parallel algorithm for multilevel k-way hypergraph partitioning. In: *Proceedings of 3rd International Symposium on Parallel and Distributed Computing*. (2004)
13. Ihler, E., Wagner, D., Wagner, F.: Modeling hypergraphs by graphs with the same mincut properties. *Inf. Process. Lett.* **45** (1993) 171–175
14. Sanchis, L.A.: Multiple-way network partitioning. *IEEE Transactions on Computers* (1989) 6281
15. Karypis, G., Aggarwal, R., Kumar, V., Shekhar, S.: Multilevel hypergraph partitioning: applications in vlsi domain. *IEEE Trans. Very Large Scale Integr. Syst.* **7** (1999) 69–79
16. Karypis, G., Kumar, V.: hmetis 1.5: A hypergraph partitioning package. Technical report, University of Minnesota, Available on <http://www.cs.umn.edu/hmetis> (1998)
17. Trifunovic, A., Knottenbelt, W.: Parkway 2.0: A parallel multilevel hypergraph partitioning tool. In: *Proceedings of 19th International Symposium on Computer and Information Sciences (ISCIS 2004)*. Volume 3280. (2004) 789–800
18. Garey, M., Johnson, D.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co. (1979)
19. Karypis, G.: Multilevel hypergraph partitioning. Technical report #02-25, University of Minnesota (2002)