Abstract—The goal of still color image segmentation is to divide the image into homogeneous regions. Object extraction, object recognition and object-based compression are typical applications that use still segmentation as a low-level image processing. In this paper, we present a method for color image segmentation. It formulates a color image segmentation problem as a partition of a Color Image Neighborhood Hypergraph (CINH) representation. Both global and local information are considered when we process this representation. To overcome the computational difficulty of directly solving the CINH partitioning problem, a multilevel hypergraph partitioning has been used. The proposed method is compared with the graph based segmentation algorithm using normalized cut criteria. The experimental results demonstrate the effectiveness of the proposed approach.

I. INTRODUCTION

Image segmentation, whose goal is the partition of the image domain, is a long standing research subject in computer vision [1]. 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. They can be classified into five approaches [2], 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 [3], [4], [5], [6]. 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. 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 bipoartitioning [4],[3],[6]. 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 [7]. A common solution is to convert the partitioning problem into a clustering problem [4][8].

The main drawback of 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 (Fig. 1).

![Graph and Hypergraph](image)

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

Also, like graphs, hypergraphs may be partitioned such that a cut metric is minimized. 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 [9]. It has been shown that, in general, there does not exist a graph model that correctly represents the cut properties of the corresponding hypergraph [10]. Recently, several serial and parallel hypergraph partitioning techniques have been extensively studied and tools support exists (e.g. hMETIS [11], PaToH [12] and Parkway [13]). 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 field and more particularly to the color image segmentation. The basic idea of this algorithm can be described as follows: we first build a hypergraph representation of the color image. Then we partition this representation into a set of vertices, representing homogeneous regions.

The Color Image Neighborhood Hypergraph 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 field such as noise removal.
The color image neighborhood hypergraph partitioning is performed using a fast multilevel programming algorithm. Throughout this paper, we will denote the hypergraph of image by the color image neighborhood hypergraph CINH.

The proposed segmentation algorithm based on the color image neighborhood hypergraph representation and the multilevel hypergraph partitioning technique is presented in section 2. In section 3, we present the color image neighborhood hypergraph presentation. The multilevel k-way color image neighborhood hypergraph partitioning algorithm is illustrated in section 4. In section 5, comparative segmentation results and performance figures are given. For objective assessment, both Berkeley segmentation dataset images and test image in the literature are used. Conclusions and perspectives are presented in section 6.

II. DESCRIPTION OF OVERALL SCHEME

The outline of the segmentation scheme is illustrated in Fig.2 and can be divided in two phases. First, a vectorial image spatial domain representation is produced using an Color Image Neighborhood Hypergraph (CINH) followed by a multilevel CINH partitioning algorithm. The multilevel algorithm for partitioning the CINH representation is based on three phase: (1) coarsening phase in which the CINH size is reduced, (2) initial partition, (3) successive refinement of the initial partition to compute the final segmentation map. In the following sections, we detail the CINH generation, CINH coarsening, initial partition, Uncorsening phase.

![Fig. 2. Block diagram of the presented method.](image)

III. IMAGE NEIGHBORHOOD HYPERGRAPH REPRESENTATION

A hypergraph $H$ on a set $X$ is a family $(E_i)_{i \in I}$ of nonempty subsets of $X$ called hyperedges with: $\bigcup_{i \in I} E_i = X$, and $\forall i \in I, I = \{1,2,\ldots,n\}, n \in N$.

Through this paper, the image will be represented by the following mapping: $I: X \subseteq \mathbb{Z}^2 \rightarrow C \subseteq \mathbb{Z}^n$. Vertices of $X$ are called pixels, elements of $C$ are called colors. The image dimension is specified using the value of $n$. In this paper, we work with a color image thus the value of $n$ is equal to 3.

A digital image is defined on a grid. A grid is a loopless regular graph, associated with a regular lattice $L$ of $\mathbb{R}^n$, which is triangular, square, or hexagonal. The most commonly used grids are the 4-connected grids and the 8-connected grids defined on a square lattice of $\mathbb{R}^2$. Throughout this paper we will be concerned only with square lattices and more particularly with 8-connected grids.

A distance $d$ on $X$ defines a grid. To define the 8-neighborhood system we use the following distance: $d_{\text{nc}}(x,x') = \max\{|i-k|,|j-l|\}$.

Let $\rho$ be the minimum number of edges between any two vertices of a grid; we can call a neighborhood system of order $\beta$ associated with the vertex $x$ the set of vertices defined by:

$$\Gamma_{\beta}(x) = \{x' \in \mathbb{R}, \ 1 \leq \rho(x,x') \leq \beta\}$$

On a grid $\Gamma_{\beta}(x)$, we can also choose the pixels which shares a given information, such as color, texture, motion, ... etc. Let $\mu$ be a similarity measure and $F$ the feature space. We have a neighborhood relation on an image defined by:

$$\Gamma_{\lambda,\beta}(x) = \{x' \in X, \mu(F(x),F(x')) \geq \lambda \ \text{and} \ x' \in \Gamma_{\beta}(x)\}$$

Using a neighborhood system, a similarity measure $\mu$, a threshold $\lambda$, and a feature space $F$, we can associate for each image $I$, a hypergraph called Image Neighborhood Hypergraph (INH):

$$H_{\lambda,\beta}(I) = (X, E_{\lambda,\beta}(x))$$

where the hyperedge centred in $x$ is defined by:

$$E_{\lambda,\beta}(x) = (x \cup \Gamma_{\lambda,\beta}(x))_{x \in X}$$

IV. MULTILEVEL K-WAY INH 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: $\text{cut}(A,B) = \sum_{E_i \in A, E_j \in B} w(E_i)$. $A$, $B$ 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 [16].

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 [17]; however, many heuristic algorithms have been developed. The survey by Alpert and Kahng [18] 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, it they partition the hypergraph using three process (Fig.3):
**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. 4).

**Initial partitioning phase:** At this point, the algorithm computes a partitioning of this hypergraph. Since the size of this hypergraph is quite small, even simple algorithms such as Kernighan-Lin (KL) [19] or Fiduccia-Mattheyses (FM) [20] lead to reasonably good solutions. **Uncoarsening phase:** final step of the algorithm 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.

### V. Experimental Results

In this section, we shall present a set of experiments in order to assess the performance of the segmentation approach we have discussed so far. The performance is carried out on a collection of images [21] and informally compares the results with the techniques of Malik et al. [4].

The proposed approach starts with a INH generation followed by a multilevel hypergraph partitioning. The steps of the proposed algorithm are described below:

- **Input:** Image \( I \), similarity measure \( \mu \), feature space \( F \), neighborhood order \( \beta \), and \( k \) desired regions.
- **Step 1:** INH generation.
- **Step 2:** Multilevel INH partitioning:
  - coarsening phase
  - initial partitioning
  - uncoarsening phase
  - multi phase refinement
- **Output:** segmented Image.

Before starting the simulations, let us present first some parameters and tools used in the proposed algorithm.

To compute the CINH representation, we can use several features that can be extracted from an image pixels and be useful for segmentation purposes, such as color, texture, statistical characteristics, etc. Even so, color is usually the most dominant and distinguishing visual feature and quite adequate for many segmentation tasks. As a result, the color feature can be considered as a sufficient feature descriptor and is also adopted in this work. The similarity measure depends on the feature spaces. Throughout this paper, we use a dissimilarity measure or colorimetric distance \( d'(I(x), I(x')) \) in color feature space and more particularly in both CIELab [22] and RGB color spaces. The thresholding can be carried out in two ways. In the first way, only one \( \lambda \) value is given for all the hyperedge center of the CINH representation. In this case, the \( \lambda \) value can be given by the users or estimated using global information of the image such as the histogram. In the second way, the \( \lambda \) is generated locally and applied in an adaptive way to the image \( I \). For each hyperedge, we calculate the \( \lambda \) value. In this contribution, we use only a global colorimetric threshold \( \lambda \).

In the CINH partitioning, we use the Hmetis package [11]. First, the CINH representation is transformed to a Weighted CINH representation. In this contribution, each vertex and hyperedge has a weight associated with it: \( w(x) = I(x) \) and \( w(E) = 1 \). \( I(x) \) is the color vector in pixel \( x \). In the coarsening phase, we use the hyperedge coarsening approach (Fig. 4b). However, in the initial partitioning phase, we use one objective function: minimize the hyperedge-cut of the partitioning. In the uncoarsening phase, we use the FM refinement algorithm [20].

The experimental results contain two steps. In the first step, we show the effect of the CINH generation on the quality of the image segmentation results. For this study, we implement the color image neighborhood hypergraph representation in two color space: RGB and CIELab. The color difference in RGB and CIELab color space used for CINH generation is defined as the Euclidean distance between two colors in this color space. In the Berkeley image database, the parameters values \( \beta \), \( \lambda \), and \( k \) are adjusted in experiments. In the second step, we compare our segmentation results with Shi and Malik algorithm.

Figure 5 shows the segmentation results obtained, using CINH representation in RGB and CIELab color spaces. From this figure, we can see that using the CINH representation in CIELab color space, we obtain significant and better

---

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

**Fig. 4.** Some coarsening schemes: (a) Edge Coarsening, (b) Hyperedge Coarsening, and (c) Modified Hyperedge Coarsening.
results. Indeed, using CINH in CIELab color space, we detect more significant regions compared to segmentation approach using CINH representation in RGB color space. The results demonstrate the improvement introduced in terms of performance using a Euclidean distance to compute CINH representation and superiority of CIELab space consequently.

In order to compare our method with an existing one, we have chosen the technique of Shi and Malik (Ncut). We have processed a group of images with our segmentation method and compared the results to Ncut algorithm. The Ncut algorithm use the optimal parameters given by authors [23]. Figure 6 shows a comparison between the proposed and Ncut algorithms. 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.

VI. CONCLUSIONS

We have presented a segmentation algorithm for color images. The segmentation is accomplished in two steps. In the first step, a image neighborhood hypergraph is generated. In the second stage, a hypergraph partitioning method is computed using a multilevel technique. Experimental results demonstrate that our approach performs better than Ncut algorithm. It can be improved in several ways (parameters: the colorimetric threshold, the unsupervised region number, etc.).

REFERENCES