

Research Article

A Fast Color Image Segmentation Approach Using GDF with Improved Region-Level Ncut

Ying Li ¹, Shuliang Wang ², Caoyuan Li,² Zhenkuan Pan,¹ and Weizhong Zhang¹

¹College of Computer Science and Technology, Qingdao University, Qingdao 266071, China

²School of Software, Beijing Institute of Technology, Beijing 100081, China

Correspondence should be addressed to Shuliang Wang; slwang2011@bit.edu.cn

Received 4 August 2017; Revised 17 November 2017; Accepted 29 November 2017; Published 2 January 2018

Academic Editor: Marco Mussetta

Copyright © 2018 Ying Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Color image segmentation is fundamental in image processing and computer vision. A novel approach, GDF-Ncut, is proposed to segment color images by integrating generalized data field (GDF) and improved normalized cuts (Ncut). To start with, the hierarchy-grid structure is constructed in the color feature space of an image in an attempt to reduce the time complexity but preserve the quality of image segmentation. Then a fast hierarchy-grid clustering is performed under GDF potential estimation and therefore image pixels are merged into disjoint oversegmented but meaningful initial regions. Finally, these regions are presented as a weighted undirected graph, upon which Ncut algorithm merges homogenous initial regions to achieve final image segmentation. The use of the fast clustering improves the effectiveness of Ncut because regions-based graph is constructed instead of pixel-based graph. Meanwhile, during the processes of Ncut matrix computation, oversegmented regions are grouped into homogeneous parts for greatly ameliorating the intermediate problems from GDF and accordingly decreasing the sensitivity to noise. Experimental results on a variety of color images demonstrate that the proposed method significantly reduces the time complexity while partitioning image into meaningful and physically connected regions. The method is potentially beneficial to serve object extraction and pattern recognition.

1. Introduction

Image segmentation [1–3] is a process of partitioning an image into meaningful disjoint regions such that each region is nearly homogenous with no intersection. It has become a basic issue of image processing and computer vision. For example, object detection [4, 5], object recognition [6, 7], knowledge inference [8, 9], image understanding [10], and medical image processing [11, 12] are all dependent on image segmentation, whose accuracy determines the quality of image analysis and interpretation.

Segmentation problem is essentially equivalent to a clustering issue, which aims at grouping the pixels into local homogenous regions. Many existing methods consider image segmentation as the clustering problem and have succeeded in dealing with this problem, such as k -means [13], region-based merging [14], mean shift [15], model-based clustering [16, 17], and parameter-independent clustering [18]. Of these methods, k -means [13] is a parametric method, requiring a

prior knowledge of the number of clustering centers. Hou et al. [18] further introduces a parameter-independent clustering method for image segmentation in attempt to remove the heavy dependence on the user-specified parameters. Comparatively, region-based merging [14], mean shift [15], and model-based clustering [16, 17] are all nonparametric approaches, which require no prior assumptions of the number of clusters, spatial distribution, and so on.

Nonparametric clustering has been advantageously used in image segmentation. Mean shift (MS) [15] is a typical segmentation approach of nonparametric density estimation. The principle behind it is that dataset characteristics are depicted with empirical probability density distribution in the feature space. However, three difficulties of MS algorithm are usually hard to tackle. First, a single MS is sensitive to the bandwidth selection, producing quite different segmented results because of the choice of different parameters. Second, MS suffers from oversegmentation. It preferably results in massive fragments and possible erroneous partitions,

especially when processing those images with only subtle distinctions between different clusters. Third, MS is very time-consuming. The high time complexity is due to execute large amounts of iterative filtering and subsequent clustering computations on every single pixel.

To overcome the oversegmentation problem of MS algorithm, the authors [19] have combined MS with graph-based Ncut method [20] for segmenting images. In [21], Bo et al. propose the use of dynamic region merging for automatic image segmentation. This approach first employs MS to generate an initially oversegmented image, on which final segmentation is achieved by iteratively merging the regions according to a statistical test. The experimental results of these two MS-based segmentation approaches demonstrate that the inaccurate clusters of some regions are indeed improved to a certain extent. However, the whole time complexity was further increased because of Ncut implementation dynamic region merging procedure in [21]. In view of this situation, a new algorithm, GDF-Ncut, is here proposed to partition color images into meaningful regions by incorporating GDF (generalized data field) and Ncut (normalized cuts). GDF-Ncut is beneficial to improve the quality of image segmentation, as well as reducing the computational complexity.

The rest of this paper is organized as follows: Section 2 briefly introduces the related backgrounds, including the idea of data field and graph-based Ncut partition. The proposed GDF-Ncut is elaborated in Section 3 in detail. Section 4 presents the experimental study. Finally, Section 5 concludes the paper.

2. Background

In this section, we briefly review the theories of generalized data field and normalized Ncut, which constructs the crucial components of the proposed GDF-Ncut.

2.1. Extending Physical Field to Data Field. The physical field is initially defined by Michael Faraday, the famous British physicist, as a kind of media, which is able to transmit the noncontact interactions between objects, such as gravitation, electrostatic force, and magnetic force. And the theories of interaction and field have been successfully used for the description of the objective world in physics at different scales and levels, in particular for mechanics, thermal physics, electromagnetism physics, and modern physics. Particularly, potential field is a type of a time-independent vector field with desirable mathematic properties, explicitly including gravitational field, electric field, and nuclear field. We here select electrostatic field as an example for a detailed illustration of field theory.

In reality, electrostatic field can be generated by a point charge Q [22]. Assuming that the electronic potential is zero at infinite point, the potential value of point x is

$$\hat{\varphi}(r) = \frac{Q}{4\pi\epsilon_0 \|r\|}, \quad (1)$$

where x is a two-dimensional coordinate in the electrostatic field, r is the radial coordinate of point x with the point charge

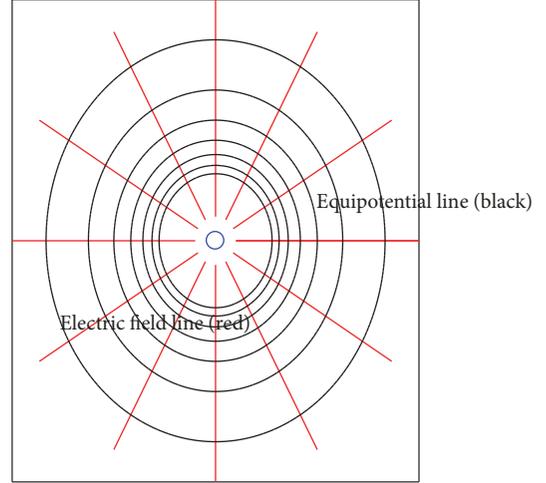


FIGURE 1: The distribution of equipotential lines in the electrostatic field.

as the origin of spherical coordinates, and ϵ_0 is the vacuum permittivity. Figure 1 is the distribution of an electrostatic field. It can be easily seen that the potential is higher with denser equipotential lines while being close to the center of the charge. Moreover, the potential of any point in field is isotropic and single-valued with respect to its spatial location, which is positively proportional to the electricity of the point charge but negatively correlates with the distance between the point and field source.

Suppose that there are n point charges x_1, x_2, \dots, x_n with individual charges Q_1, Q_2, \dots, Q_n in space. The potential of any point x is subject to the field jointly generated by all of the point charges,

$$\hat{\varphi}(r) = \frac{1}{4\pi\epsilon_0} \sum_{i=1}^n \frac{Q_i}{\|r - r_i\|}, \quad (2)$$

where $\|r - r_i\|$ is the distance from point x to the particle x_i . Figure 2 visualizes the distribution of electrostatic field resulting from the superposition of two point charges.

Depending on the idea of physical field, the concept of interaction between physical particles and the way of field description are introduced into the abstract number field space in an aim to discover the hidden but valuable information. Each data object in the space is viewed as a particle with certain mass. A virtual field is thus generated owing to the joint interactions among these data objects, which is called data field [23, 24]. Assuming that data field is generated by a data object y in space Ω , $\forall x \in \Omega$, the potential is accordingly defined as

$$\varphi_y(x) = m \cdot \varphi\left(\frac{\|x - y\|}{\sigma}\right), \quad (3)$$

where m is the mass of object y and σ is an single-valued impact factor indicating the interaction range of object y . The function $\varphi(\cdot)$ is defined as a unit potential function, satisfying the following criterion:

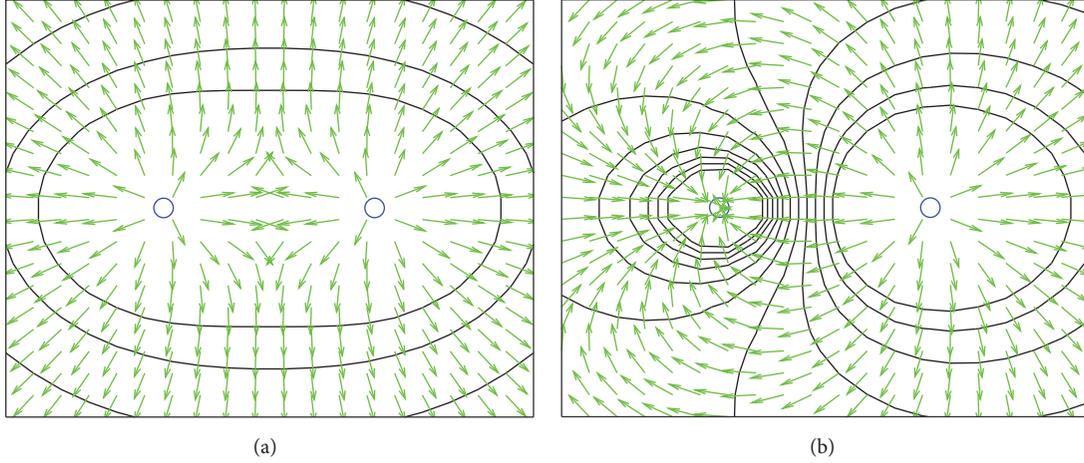


FIGURE 2: The electrostatic field distribution generated by two point charges. (a) Electrostatic field generated by two positive charges; (b) electrostatic field generated by a positive charge and a negative charge.

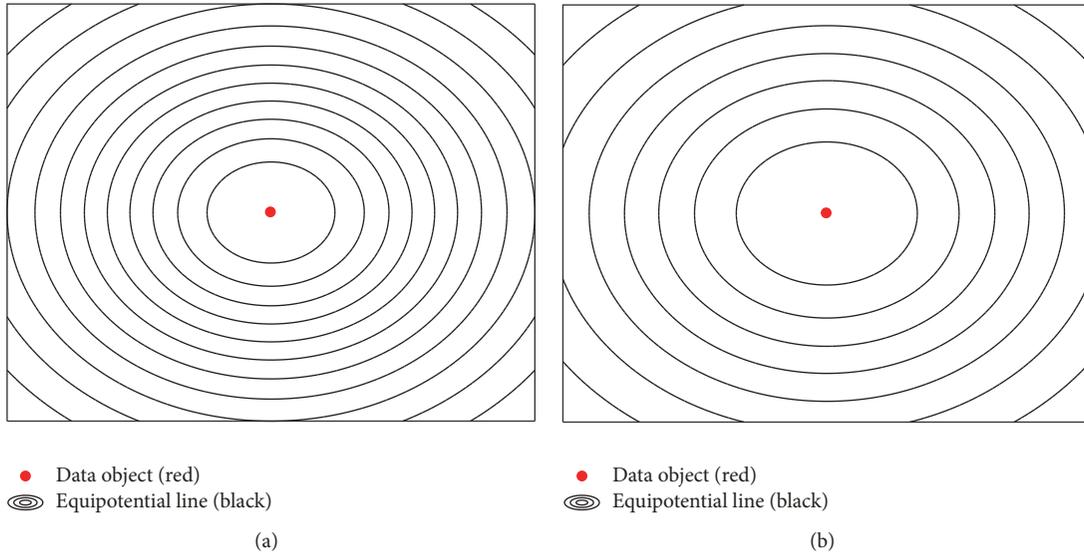


FIGURE 3: The distribution of data field with different unit potential functions. (a) The unit potential function is in accordance with (4); (b) the unit potential function is in accordance with (5).

- (a) $\varphi(\cdot)$ is a continuous and bounded function defined in the space Ω ;
- (b) $\varphi(\cdot)$ is symmetric in the space Ω ;
- (c) $\varphi(\cdot)$ is a strictly decreasing function in the space Ω .

The unit potential function can be defined on the basis of different physical fields, such as electrostatic field and nuclear field. We, for illustration, exhibit two specific definitions of $\varphi(\cdot)$. Considering the electrostatic field, the unit potential function is given,

$$\varphi(u) = \frac{1}{1 + u^k}. \quad (4)$$

Corresponding to the nuclear field in physics [25], the unit potential function can be written as

$$\varphi(u) = e^{-u^k}, \quad (5)$$

where $u = \|x - y\|/\sigma$, $m \geq 0$ is the mass of data object, and $k \in \mathbb{N}$ is the distance index. The parameter k is commonly set to 2 for convenient calculation of $\|x - y\|$. Assuming that the parameter $\sigma = 1$, the field distribution is generated by a single data object in Figure 3. In detail, Figures 3(a) and 3(b) separately visualize the distribution with different unit functions. Seen from Figure 3, the energy of data object disperses evenly in all directions, which highly accord with that in physical field.

Given a dataset $D = \{x_1, x_2, \dots, x_n\}$ in space Ω , n is the number of data object, these data objects interact with each other and jointly generate a data field. The potential of any given point x in this data field is written,

$$\hat{\varphi}(x) = \sum_{i=1}^n m_i \cdot \varphi\left(\frac{\|x - y\|}{\sigma}\right), \quad (6)$$

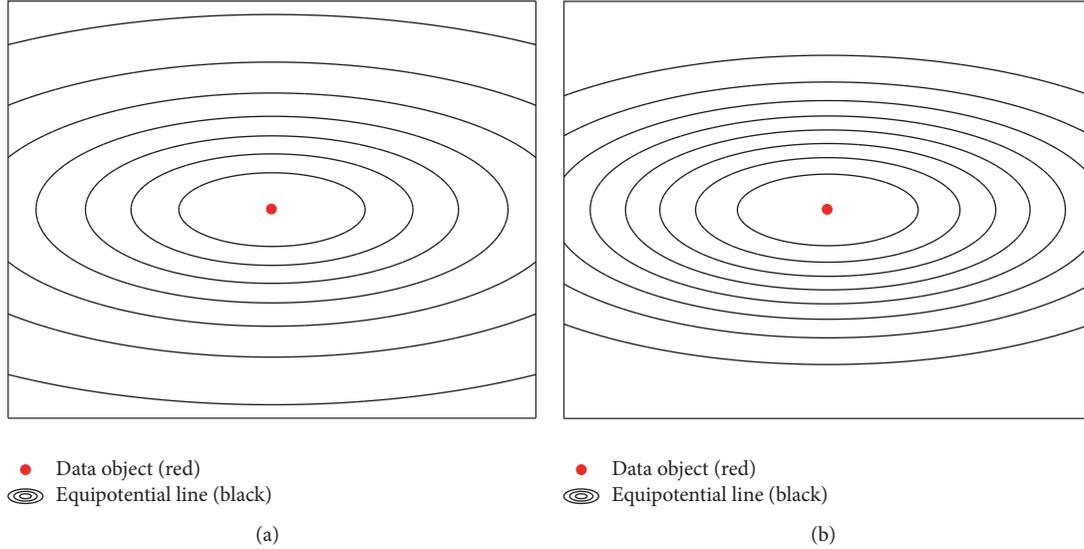


FIGURE 4: The distribution of generalized data field with different unit potential functions. (a) The unit potential function accords with (4); (b) the unit potential function accords with (5).

where $\varphi(\cdot)$ is the unit potential function, σ is a single-valued impact factor, and m_i is the mass of object x_i with $i = 1, \dots, n$. The mass m_i , $i = 1, \dots, n$ satisfies

$$\sum_{i=1}^n m_i = 1, \quad m_i \geq 0, \quad (7)$$

$$\lim_{n \rightarrow \infty} n \sup_{1 \leq i \leq n} \{m_i\} = 1.$$

For multidimensional space, the parameter σ is defined to be the same value in different dimensions according to (6). In this case, the potential estimation is probably unable to represent the truthful distribution of data objects since data objects usually own different prosperity in each dimension. In response to this issue, we extend data field to generalized data field, in which the impact factor is anisotropic; that is to say, data have different impact factors along different dimensions. At this point, (3) is rewritten,

$$\varphi_y(x) = m \cdot \varphi(\|H^{-1}(x - x_i)\|), \quad (8)$$

where H is a diagonal matrix. The diagonal elements of matrix H are $\text{diag}(H) = (\sigma_1, \dots, \sigma_p)^T$, $\sigma_i > 0$, $i = 1, \dots, p$.

Equation (6) also becomes

$$\hat{\varphi}(x) = \sum_{i=1}^n m_i \cdot \varphi(\|H^{-1}(x - x_i)\|). \quad (9)$$

For clear illustration of the difference between data field and generalized data field, we exhibit the distribution of single-object data field in two-dimensional space, as shown in Figure 4. The parameters H and m are set to $\text{diag}(H) = (1, 0.5)^T$ and $m = 1$, respectively. In comparison with Figure 3, the generalized data field is representative of variation among dimensions.

We are then motivated to analyze the characteristics of any dataset in data field or in generalized data field. For example, Figure 5(a) is the test set consisting of 600 data generated for three Gaussian Models and the mass of each data is $1/600$. The potential distribution of data set is displayed as Figure 5(b) after calculating the potential of each data points, where $\sigma = 1$ (see (9)). It can be easily seen that data objects collected around three centers based on the distribution of the equipotential lines. Equipotential lines are distributed densely while being close to centers, indicating that the potential achieves higher value with close distance to centers and finally achieve the maxima in the centers. Under this situation, data objects can be assigned into three clusters. This result conforms to the intrinsic distribution of data points, which are generated by three Gaussian models. Therefore, we can utilize the potential distribution of data field to get the natural clustering of data objects. In detail, the realization of clustering chiefly consists of two procedures. The clustering centers are first obtained by the detection of the maxima of potential distribution and then objects are grouped into clusters by arranging them to the nearest centers.

2.2. Graph-Based Partition. Graph-based methods are mainly involved in normalized cuts (Ncut) [20], average association [26], and minimum cut [27]. These methods are used for image segmentation by constructing a weighed graph for describing relationships between pixels. Specifically, each pixel is regarded as a vertex, two adjacent pixels are connected with an edge, and the dissimilarity between such two pixels is computed as the weight of the edge. We here simply introduce the idea of well-known normalized cuts since it is applied in our proposed algorithm.

Graph-based partition assumes that a graph is characterized as $G = (V, E, W)$, where V is the set of nodes, E is the set of edges connecting nodes, and W is the weight matrix. The

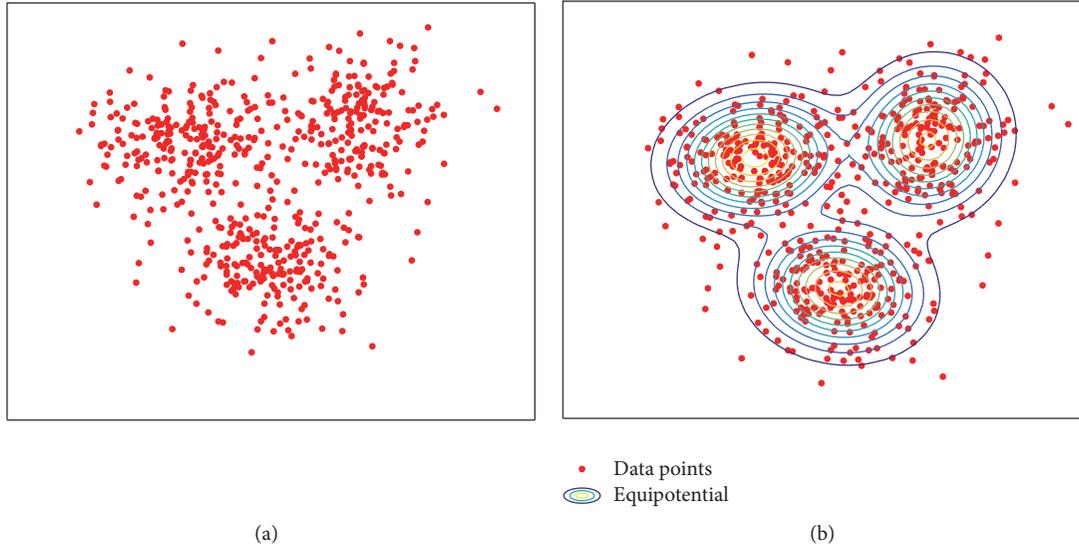


FIGURE 5: Potential distribution of test set. (a) Test set; (b) equipotential lines of test set.

edge weight, $w(i, j)$, is defined as a function of the similarity between nodes i and j . We can partition the graph into two disjoint sets, that is, A and B ($A \cap B = \emptyset$, $A \cup B = V$) by simply removing the edges connecting A and B . And the total weight of removed edges is thus called a cut in graph theoretic language [20],

$$\text{cut}(A, B) = \sum_{u \in A, v \in B} w(u, v). \quad (10)$$

According to (10), the cut measures the degree of the dissimilarity between A and B . Therefore, many algorithms have been proposed to discover the minimum cut of a graph [19]. However, the minimum cut has a bias for partitioning small sets of isolated nodes in the graph since the cut in (10) increases with the rise of the number of edges connecting the two partitioned parts. To avoid this problem, normalized cuts (Ncut) are defined for a given partition:

$$\text{Ncut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)}, \quad (11)$$

where $\text{assoc}(A, V)$ is the total number of connections from the nodes in A to all nodes in the graph and $\text{assoc}(B, V)$ is similarly defined. In this case, the cut criterion indeed avoids a bias in partitioning out small isolated points.

Assume that a partition of nodes V leads to two disjoint sets A and B . Denote $|V|$ as the number of nodes in V . Let x be a $|V|$ dimensional vector, $x_i = 1$ if node i is in A and -1 otherwise. Let $d(i) = \sum_j w(i, j)$ be the total connections from node i to all other nodes. Thus $\text{Ncut}(A, B)$ can be rewritten as

$$\text{Ncut}(A, B) = \frac{\sum_{(x_i > 0, x_j < 0)} -w_{ij} x_i x_j}{\sum_{x_i > 0} d_i} + \frac{\sum_{(x_i < 0, x_j > 0)} -w_{ij} x_i x_j}{\sum_{x_i < 0} d_i}. \quad (12)$$

Let D be a $|V| \times |V|$ diagonal matrix with d on its diagonal and W be a symmetrical matrix with $W(i, j) = w_{ij}$. Minimizing $\text{Ncut}(A, B)$ in (12) is deduced as

$$\min_x \text{Ncut}(x) = \min_y \frac{y^T (D - W) y}{y^T D y}, \quad (13)$$

with the condition

$$y(i) \in \left\{ 1, -\frac{\sum_{x_i > 0} d_i}{\sum_{x_i < 0} d_i} \right\}, \quad (14)$$

$$y^T D 1 = 0.$$

If y is relaxed to take on any real value, (13) can be minimized by solving the generalized eigenvalue system,

$$(D - W) y = \lambda D y. \quad (15)$$

The vector corresponding to the second smallest eigenvalue is the solution to Ncut problem, which is called the second smallest eigenvector of (15).

3. GDF-Ncut Principles

GDF-Ncut is proposed for fast color image segmentation by combining generalized data field with improved normalized cuts. The implementation procedures of GDF-Ncut are specifically displayed in Figure 6.

Considering that image segmentation can be performed under various color spaces, it is essential to choose the most suitable space before describing the details of the algorithm. The spaces $L^* u^* v^*$ and $L^* a^* b^*$ are commonly applied to image segmentation since the color difference is in accordance with the Euclidean distance in either feature space [28, 29]. L^* represents the lightness coordinate in both cases. The chromaticity coordinates are defined differently

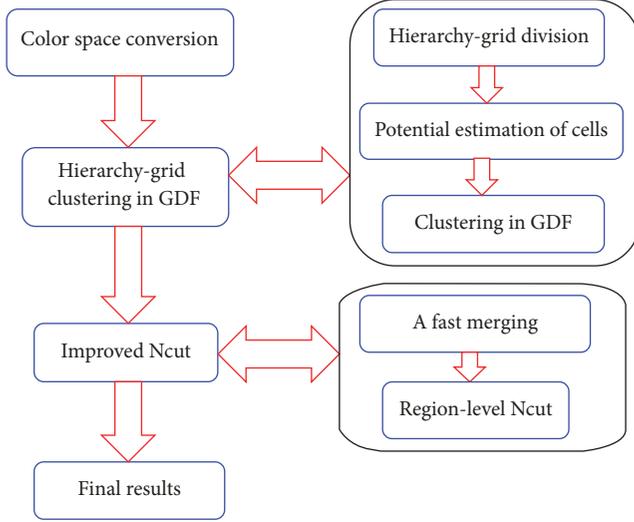


FIGURE 6: Implementation process for the proposed GDF-Ncut.

for $L^*u^*v^*$ and $L^*a^*b^*$. We practically find no obvious distinction on segmentation results while implementing the proposed algorithm for image segmentation based on the two color spaces. In this paper, the space $L^*u^*v^*$ is selected as the feature space because it retains linear mapping property during the process of image segmentation.

After color space conversion, we are motivated to hire the theory of generalized data field for naturally grouping all pixels into clusters in three-dimensional $L^*u^*v^*$ space. Each pixel is viewed as a data object for nonparametric clustering in GDF; however, it would be computationally expensive for dealing with larger image. To overcome this shortcoming, we implement hierarchy-grid clustering in GDF. To this end, it mainly involves the three crucial parts: hierarchy-grid division, cell potential estimation, and cell-based clustering. Then the pixel clusters are projected to the original images domain and yield disjoint regions. Unfortunately, these regions are vulnerable to contain massive fragments such that the image is oversegmented. In response to this issue, the improved Ncut is further used to merge the initial oversegmented regions and achieve final segmentation results.

3.1. Hierarchy-Grid Clustering in GDF. In this section, the idea of hierarchy-grid clustering in GDF is illustrated in detail for producing initial image segmentation. We first construct the two-level hierarchical grid division in L^*u^*v feature space. To do this, we partition $2N \times 2N \times 2N$ uniform cells in L^*u^*v space as the first-level grid division and then repartition $N \times N \times N$ uniform cells as the second-level grid division. Actually, each cell in the second-level grid division can also be expediently created by merging every eight-neighborhood first-level cells. For example, in Figure 7, the left image is a profile of eight-neighborhood cells in the first-level grid division and the right one is its corresponding profile by merging the left cells.

Based on the two-level grid division in feature space, we conduct the clustering of cell objects in the first-level

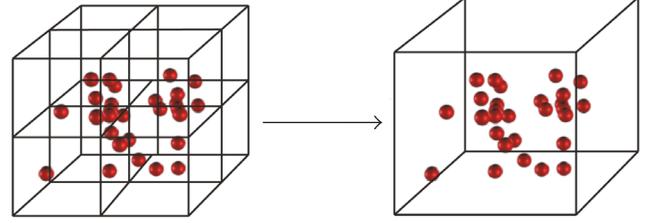


FIGURE 7: Profile changes from eight-neighborhood cells to a cell.

division rather than the pixel-based objects. More precisely, the potential of each cell in the first-level division is calculated relying on the contribution of all the discrete cells in the second-level division. The time complexity is consequently reduced to $O(8N^6)$, which is independent of the size of the image and is only determined by the division level in the feature space.

Assume that dataset S denotes cells in the first-level division and dataset B corresponds to cells in the second-level division. We define $f_{i_s j_s k_s}$ to present the average colorful feature of pixels within a cell of the first-level grid division,

$$f_{i_s j_s k_s} = (f_{i_s j_s k_s}^{(1)}, f_{i_s j_s k_s}^{(2)}, f_{i_s j_s k_s}^{(3)})^T, \quad (16)$$

where $(i_s, j_s, k_s) \in S, i_s, j_s, k_s \in Z^*$. Similar to cells in the second-level grid division, $f_{i_B j_B k_B}$ is defined as follows:

$$f_{i_B j_B k_B} = (f_{i_B j_B k_B}^{(1)}, f_{i_B j_B k_B}^{(2)}, f_{i_B j_B k_B}^{(3)})^T, \quad (17)$$

where $(i_B, j_B, k_B) \in B, i_B, j_B, k_B \in Z^*$. Thus, the potential of each cell in the first-level grid division is calculated based on the contributions of cells in the second-level grid division,

$$\begin{aligned} \hat{\varphi}(f_{i_s j_s k_s})_G &= \sum_{f_{i_B j_B k_B} \in B} w_{i_B j_B k_B} \cdot \varphi\left(\left\|\frac{f_{i_s j_s k_s} - f_{i_B j_B k_B}}{H}\right\|\right), \end{aligned} \quad (18)$$

where $w_{i_B j_B k_B}$ is the number of pixels located in the (i_B, j_B, k_B) cell and H is a diagonal matrix of impact factors. Let $\varphi(\cdot)$ be the unit potential function defined in (5); (18) thus becomes

$$\begin{aligned} \hat{\varphi}(f_{i_s j_s k_s})_G &= \sum_{f_{i_B j_B k_B} \in B} \prod_{j=1}^3 w_{i_B j_B k_B} \\ &\cdot \exp\left(-\left(\frac{f_{i_s j_s k_s}^{(j)} - f_{i_B j_B k_B}^{(j)}}{\sigma_j}\right)^2\right), \end{aligned} \quad (19)$$

where $\exp(\cdot) = e^{(\cdot)}$. As the estimation of potential is similar to kernel density estimation, the diagonal value of the impact factor H , $\text{diag}(H)$, can be easily set to a multiple of the window width h , that is, $\text{diag}(H) = ch$, where c is the proportionality coefficient and $h = (h_1, h_2, h_3)^T$ is the window width of the kernel estimation. The parameter c can be self-tuned to obtain different levels of image segmentation and h is assigned with Rules of Thumb [30, 31]. The gradient of potential function in (19) can be further calculated,

$$\begin{aligned} \nabla \hat{\varphi}(f_{i_s j_s k_s})_G &= \left(\frac{\partial \hat{\varphi}(f_{i_s j_s k_s})}{\partial f_{i_s j_s k_s}^{(1)}}, \frac{\partial \hat{\varphi}(f_{i_s j_s k_s})}{\partial f_{i_s j_s k_s}^{(2)}}, \frac{\partial \hat{\varphi}(f_{i_s j_s k_s})}{\partial f_{i_s j_s k_s}^{(3)}} \right)^T \\ &= \begin{pmatrix} \frac{w_{i_u j_u k_u}}{\sigma_1} (f_{i_s j_s k_s} - f_{i_B j_B k_B}^{(1)}) \sum_B \exp' \left(\frac{f_{i_s j_s k_s} - f_{i_B j_B k_B}^{(1)}}{\sigma_1} \right) \exp \left(\frac{f_{i_s j_s k_s} - f_{i_B j_B k_B}^{(2)}}{\sigma_2} \right) \exp \left(\frac{f_{i_s j_s k_s} - f_{i_B j_B k_B}^{(3)}}{\sigma_3} \right) \\ \frac{w_{i_u j_u k_u}}{\sigma_2} (f_{i_s j_s k_s} - f_{i_B j_B k_B}^{(2)}) \sum_B \exp \left(\frac{f_{i_s j_s k_s} - f_{i_B j_B k_B}^{(1)}}{\sigma_1} \right) \exp' \left(\frac{f_{i_s j_s k_s} - f_{i_B j_B k_B}^{(2)}}{\sigma_2} \right) \exp \left(\frac{f_{i_s j_s k_s} - f_{i_B j_B k_B}^{(3)}}{\sigma_3} \right) \\ \frac{w_{i_u j_u k_u}}{\sigma_3} (f_{i_s j_s k_s} - f_{i_B j_B k_B}^{(3)}) \sum_B \exp \left(\frac{f_{i_s j_s k_s} - f_{i_B j_B k_B}^{(1)}}{\sigma_1} \right) \exp \left(\frac{f_{i_s j_s k_s} - f_{i_B j_B k_B}^{(2)}}{\sigma_2} \right) \exp' \left(\frac{f_{i_s j_s k_s} - f_{i_B j_B k_B}^{(3)}}{\sigma_3} \right) \end{pmatrix}, \end{aligned} \quad (20)$$

where ∇ is the vector differential operator and $\partial \hat{\varphi}(f_{i_s j_s k_s}) / \partial f_{i_s j_s k_s}^{(1)}$, $\partial \hat{\varphi}(f_{i_s j_s k_s}) / \partial f_{i_s j_s k_s}^{(2)}$, and $\partial \hat{\varphi}(f_{i_s j_s k_s}) / \partial f_{i_s j_s k_s}^{(3)}$ are the partial derivatives of function $\hat{\varphi}(\cdot)$.

The specific implementation process of this part is depicted in form of pseudocodes as follows.

Algorithm 1 (cell initialization with their gradient calculation).

Preset. N .

Input. All image data objects with their color features.

Output. Cell set B and S initialization.

Process.

(1) Construct 3-dimensional grid cells with side length of $2N \times 2N \times 2N$, where N is set by the experimenter; for example, $N = 8$. If the cell has data objects, its feature is the mean of features of data objects; otherwise, it is the center of the cell.

(2) In a similar way, obtain the second grid with side length of $N \times N \times N$.

(3) Initialize the datasets S and B .

(4) Calculate the impact factor in formula (19) and then calculate the gradients of all the cells in S according to formula (20).

Based on the potential estimation of cells in the first-level grid division above, we proceed with the clustering of these cells. The first step of clustering in data field is to detect the clustering centers, namely, maximal points of potential distribution. The maximal points of potential are located at the zero of the gradient; that is,

$$\nabla \hat{\varphi}(f) = 0. \quad (21)$$

In practice, it is so in theory but impracticable owing to difficulties of solving (21) despite the fact that the maximal points are of great significance group cell objects into well-defined clusters. In view of the fact, candidate cluster centers are proposed as cluster centers instead of maximal points. Each candidate center meets the condition that it consists of

one cell at least. Algorithm 2 describes the process of finding the candidates cluster centers as follows.

Algorithm 2 (detection of candidate cluster centers).

Input. The gradient of each cell in datasets S .

Output. Candidate cluster centers.

Process.

(1) For cell $(i_s, j_s, k_s) \in S$.

(2) Add cell (i_s, j_s, k_s) and cell $(i_s - 1, j_s, k_s)$ into the initial set L_G if $\partial \hat{\varphi}(f_{i_s j_s k_s}) / \partial f_{i_s j_s k_s}^{(1)}$ satisfies one of the following conditions: (a) $\partial \hat{\varphi}(f_{i_s j_s k_s}) / \partial f_{i_s j_s k_s}^{(1)} = 0$; (b) $\partial \hat{\varphi}(f_{(i_s-1) j_s k_s}) / \partial f_{(i_s-1) j_s k_s}^{(1)} > 0$ and $\partial \hat{\varphi}(f_{i_s j_s k_s}) / \partial f_{i_s j_s k_s}^{(1)} < 0$.

(3) Go back to step (1) until all the cells in S are traversed.

(4) Accordingly, initialize sets u_G and v_G for the dimensions u and v .

(5) Calculate the intersection of the initial sets L_G , u_G , and v_G as the candidate cluster set.

(6) Merge every two candidate clusters if there are common cell elements in their content lists until all clusters have been processed. Update the merged clusters as the candidate centers and ensure that no duplicate cell elements are involved.

To do this, we start with sequentially detecting their adjacent cells outward around candidate centers on the basis of the "gradient criterion." Given a cell of candidate center, we find its 6-neighbor cells and compare the gradients between them. The cells whose $\|\nabla \hat{\varphi}(f)\|$ are larger than that of the center are assigned to the cluster. Then we continue to recursively compare the gradients between the added cells and their 6 neighbor cells. The qualified cells are assigned into the cluster until no new cell increases in the cluster. Every candidate center repeats the same work to group different clusters. All cell objects are eventually arranged to be one of the clusters. We accordingly classify each pixel into the cluster that its located cell belongs to. The clustering pixels are mapped to the spatial image, which yield continuous but disjoint regions in the image. We eliminate the smaller regions with less than forty pixels; however, the initial segmentation results are still

oversegmented. The core execution of this part is depicted as follows.

Algorithm 3 (clustering in GDF and segments generating).

Input. The gradient of each cell in datasets S .

Output. Initial segmentation.

Process.

(1) For each cluster in the merged result set, complete the cluster by searching for all cells that confirm to the “gradient criterion” described along the direction of coordinate axis.

(2) According to the generated clusters, assign cluster label for all cells and pixels.

(3) Eliminate small fragments whose size is less than forty by adjusting the cluster label of each point with respect to the “smooth standard.” That is, when the size of the neighbor region is less than forty, revise the label value of the point so that it is equal to the label of the largest periphery of the neighbor region.

(4) Generate all segments with respective segment labels. Thus S is the initial segmentation.

3.2. Merging Criteria Based on Improved Region-Level Ncut. Construct a graph $G = (V, E, W)$ by applying GDF-partitioned regions as the nodes and connecting each pair of regions via an edge. The weight on each pair of nodes should signify the similarity that two regions belong to one object. Suppose that an image is segmented into R disjoint regions Ω_i ($i = 1, 2, \dots, R$), which contain n_i pixels. We define $F_i = (F_i^{(1)}, F_i^{(2)}, F_i^{(3)})$ ($i = 1, 2, \dots, N$) to be the mean vector of pixels in each region Ω_i . The weight of each edge is measured by the similarity between each pair of regions,

$$w(i, j) = \begin{cases} e^{-\|F_i - F_j\|/\sigma_f} & \text{if } \Omega_i, \Omega_j \text{ are adjacent} \\ 0 & \text{otherwise,} \end{cases} \quad (22)$$

where $\|\cdot\|$ is the vector norm operator and σ_f is the fixed scaling factor.

An example of the graph structure is depicted in Figure 8. The segmented image is composed of six regions, as shown in Figure 8(a). Each region is represented by a node and adjacent regions are connected by an edge, which generate the weighted graph in Figure 8(b).

According to (22), the weight matrix is expressed as

$$W = \begin{bmatrix} 1 & w(1,2) & 0 & w(1,4) & w(1,5) & 0 \\ w(1,2) & 1 & w(2,3) & w(2,4) & 0 & 0 \\ 0 & w(2,3) & 1 & w(3,4) & 0 & w(3,6) \\ w(1,4) & w(2,4) & w(3,4) & 1 & w(4,5) & w(4,6) \\ w(1,5) & 0 & 0 & w(4,5) & 1 & w(5,6) \\ 0 & 0 & w(3,6) & w(4,6) & w(5,6) & 1 \end{bmatrix}. \quad (23)$$

The core procedures of this part are depicted in form of pseudocodes as follows.

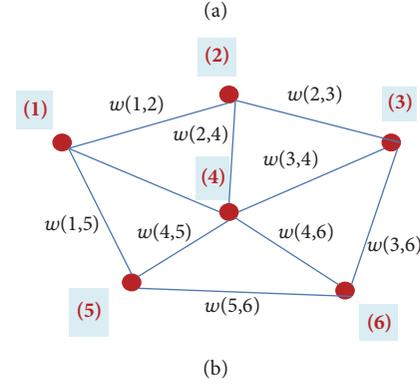
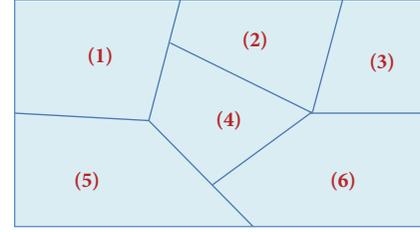


FIGURE 8: Example of the partitioning graph. (a) Segmented image. (b) Weighted graph.

Algorithm 4 (building feature space matrix and normalized cuts).

Preset. cut.

Input. Oversegmented regions.

Output. Final segmentation.

Process.

(1) Form the initial feature space matrix W in terms of (22) and let $W' = W$.

(2) Bipartition the current segment set-father segment set. First, the feature space matrixes W' and D for the current set should be acquired by selecting corresponding values stored in W . Second, solve the generalized eigenvalue system $(D-W)y = \lambda D$, then obtain the eigenvalues and eigenvectors.

(3) By computing and comparing the *cut* values for all possible division schemes, determine optimal partition pattern and generate two child segment sets.

(4) Recursively execute steps (2) and (3) until the current father segment set could not be bipartitioned, scilicet, the size of set is 1, or all *cut* values of possible division schemes surpass the threshold.

(5) Arrange the *cut* result. Set group label for each segment and obtain final segmentation.

The region-level Ncut reduces the size of the weight matrix in comparison with pixel-level Ncut, which significantly results in lower time complexity. However, for some images, the initial segmentations still have many smaller regions. Under this situation, the Ncut procedure still consumes much time to group initial regions and furthermore has difficulty in good performance. In [19], the authors use

multiple nodes rather than a single one for each region to improve segmentation quality at the expense of time. Instead, we propose an effective solution to the problems. A fast merging is achieved to reduce the number of the initial regions before implementing the Ncut procedure.

With initial segmentation generated by GDF, the distance between adjacent regions is calculated based on the criterion of Euclidean distance. Then we define threshold T_0 and merge the regions if their distance is less than T_0 with the precondition that the number of all the regions is more than N_0 . We iteratively execute this merging until all the adjacent regions cannot satisfy the distance condition. Evidently, the threshold plays a significant role in the process of merging, which determines the quality of segmentation. Here, we calculate the threshold as follows. For any region Ω_k , $k = 1 \cdots R$, we find its nearest adjacent region, denoted as Ω_{k0} . The threshold T_0 is set to the geometric mean of all the adjacent distance; that is,

$$T_0 = \sqrt[N]{\prod_{k=1}^N \text{dist}(\Omega_k, \Omega_{k0})}. \quad (24)$$

Note that the parameter T_0 depends on the initial segmentation and is a fixed value in the coming iteration. After our implementation of fast merging, the Ncut algorithm is employed to obtain the final segmentation results.

Algorithm 5 (implementation of the fast merging and Ncut).

Preset. N, c .

Input. Initial segmentation.

Output. Segmentation results after the fast merging.

Process.

- (1) For each region, compute the distance with its adjacent regions and find the nearest region.
- (2) Calculate threshold T_0 .
- (3) For $i = 1 : R$.
- (4) For region Ω_k , if the number of all the regions is more than and $\text{dist}(k, k_0) < T_0$, then merge the two region.
- (5) End for.
- (6) Update the regions and their adjacent regions and the number of regions R .
- (7) Go back to step (3) until no region is merged.

3.3. Algorithmic Description. Based on the situation above, GDF partitions an image into oversegmented regions and then the improved normalized Ncut merges these regions into homogenous. Algorithm 5 describes in detail the whole implementation process of GDF-Ncut as follows.

Algorithm 6 (segmentation by the proposed algorithm).

Preset. N, c, cut .

Input. All data objects of an image.

Output. Final segmentation results.

Process.

- (1) Transform RGB color space to $L^*u^*v^*$ color space.
- (2) Use Algorithm 1 for cell mesh and the calculation of the cell gradient.
- (3) Use Algorithm 2 to find the candidate centers and then use Algorithm 3 to group the cells in the first-level grid division into different clusters. Map the clusters into the plane domain, which produces the initial segmentation.
- (4) Use Algorithm 5 to quickly merge the initial regions.
- (5) Use Algorithm 4 to further merge the segmentation from step (4).
- (6) Return final segmentation.

The proposed algorithm contains five parameters $\{N, c, N_0, T_0, cut\}$ for image segmentation. In implementation, the two parameters N_0 and T_0 are fixed to a certain value. For parameter c , it is tuned to 1 in most cases; otherwise, it is equal to 0.5 when $c = 1$ is not satisfied to obtain reasonable initial segmentation. Therefore, the algorithm is mainly driven by two parameters: N and cut threshold. We can adjust N to correctly initialize the segmentation. The threshold cut is a parameter of controlling recursion in normalized cuts, which can be artificially adapted within a certain range to achieve final segmentation.

4. Experiments and Comparisons

GDF-Ncut is experimented for color image segmentation using the Berkeley Segmentation Dataset and Benchmark (BSDS), which are primarily evaluated in efficiency and quality of image segmentation. The BSDS is composed of a train set and a test set, containing 200 train images and 100 test images. The proposed algorithm is implemented with java to complete color image segmentation and all experiments are executed under the same computation environment, a PC equipped with 2.8 GHz Intel Core i7 CPU and 2 GB DDR3 Memory.

We, for illustration, first exhibit a specific segmentation example by gradually executing GDF-Ncut algorithm in Section 4.1. Section 4.2 tests the performance of the proposed algorithm in comparison with two rivals, MS [15], and DRM [21]. We further employ GDF-Ncut to segment representative examples from the train and test images of BSDS in Section 4.3, which further demonstrate the effectiveness of the proposed algorithm.

4.1. Illustration of GDF-Ncut Implementation. We randomly selected an image from the Internet to illustrate the segmentation implementation of GDF-Ncut. The size of the original image is 350×350 , as shown in Figure 9(a). As discussed in Section 3, GDF-Ncut is determined by two parameters N and cut . We here initialize the two parameters as $(N, cut) = (8, 0.05)$ to finish the segmentation of this image.

Figure 9(b) shows the pixels in $L^*u^*v^*$ space after the transformation from RGB to $L^*u^*v^*$. The procedure of two-level hierarchical grid division is conducted in three-dimensional L^*u^*v space. In Figure 9(c), each point

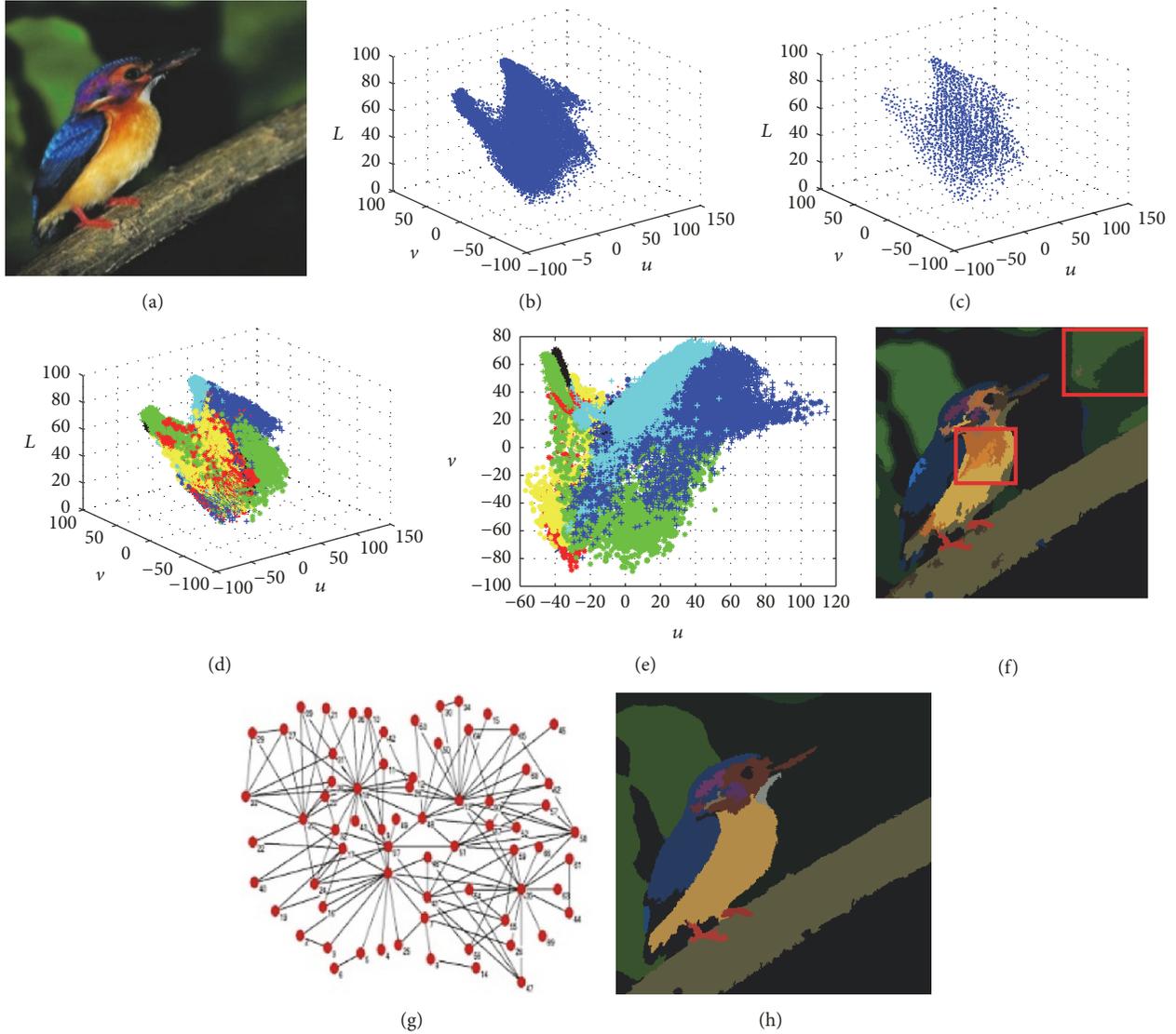


FIGURE 9: Kingfisher image segmentation. (a) Original image. (b) The corresponding $L^*u^*v^*$ space with feature set of 122500 data objects. (c) Feature set of cells in dataset S. (d) and (e) are fifteen clusters by the use of GDF. Clusters are represented by five colors combined with three symbols, that is, $\{\text{blue, red, yellow, black, green}\} \otimes \{., *, +\}$. (f) Initial oversegmentation mapped by fifteen clusters. Red rectangular boxes mark some of oversegmented regions. (g) A weighted graph. (h) Final segmented result using improved region-level to merge oversegmented regions.

represents a cell in the first-level grid division. It can be easily seen that the cell dataset in Figure 9(c) is still distributed in accordance with the original dataset in Figure 9(b) while largely reducing the number of data objects by cell mesh. The GDF procedure is subsequently employed to classify the cells in the first-level grid, resulting in the clusters of pixels in Figures 9(d) and 9(e). Figure 9(e) reveals the latter two feature attributes in Figure 9(d). These clusters in Figure 9(d) are mapped to spatial regions and thus yield an oversegmented image in Figure 9(f). For example, the trunk of kingfisher is partitioned into several parts and the branch segmentation consists of some pieces. To solve the oversegmentation problem, we are encouraged to merge the oversegmented regions by the usage of a fast merging and region-level Ncut. Briefly, a fast merging is first used to decrease the number

of initial regions and then region-level Ncut unceasingly groups the remaining regions for better final segmentation based on the construction of a weighted graph in Figure 9(g). The final segmentation result is shown in Figure 9(h). It can be effortlessly seen that the image is partitioned into homogenous regions such as branch, feathers, and trunk of kingfisher.

4.2. Comparative Experiments. To test the validity of GDF-Ncut, we compare our method with two alternatives, the mean shift method (MS) [15] and the dynamic region merging method (DRM) [21]. Mean shift algorithm realizes image segmentation by nonparametric clustering, which is determined by three parameters, the feature bandwidth, the spatial bandwidth, and the number of pixels in the minimum region.

TABLE 1: The values of parameters and running time for the images in Figure 10.

Figure	GDF-Ncut				MS			DRM (initialized by MS)	
	N	c	cut	Time (ms)	Sigma1	Sigma2	Time (ms)	Lambda	Alpha
(a)	12	1	0.25	23346	9	3	45971	1	0.2
(b)	12	1	0.25	22422	8	3	68661	1	0.2
(c)	28	1	0.04	37179	8	7	40538	1.2	0.2

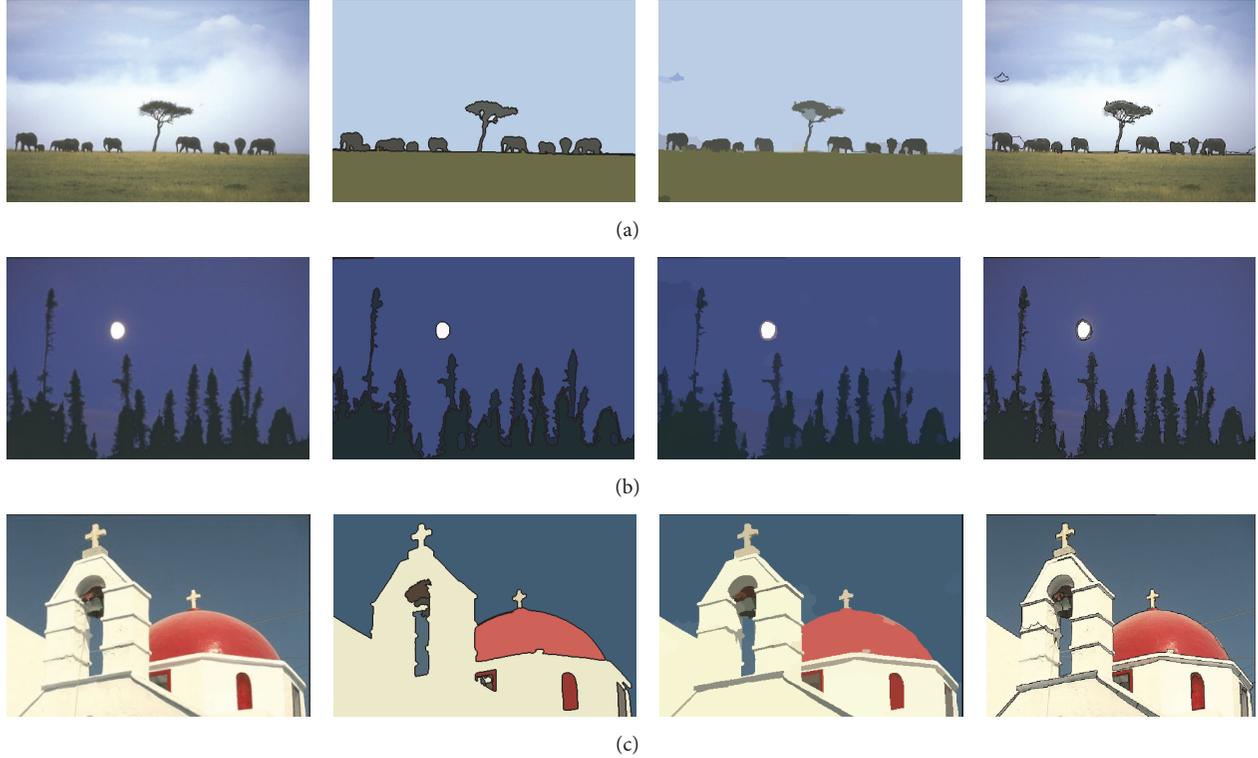


FIGURE 10: Segmentation results by different algorithms. From left to right, original images are displayed in the first column. The following three columns show segmentation results of GDF-Ncut, MS, and DRM, respectively.

Dynamic region merging (DRM) is an automatic image segmentation method that first initializes segmentation and then iteratively merges the initial regions based on a statistical test. We here generate the initial segmentation for DRM algorithm by applying MS rather than Watershed since it will improve the performance of DRM according to [21].

The segmentation results of some images are displayed in Figure 10. Table 1 is the corresponding value of parameters while segmenting images in Figure 10. Seen from Figure 10, GDF-Ncut provides better segmentation results. For instance, the sky, elephant, and tree in image (a) are appropriately partitioned into different regions by our proposed algorithm. We then employ *F-measure* [32] to evaluate the performance of the three algorithms since it is frequently applied for measuring the quality of image segmentation. Conjoining recall (R) and precision (P) with an equal weight, the *F-measure* is defined as

$$F = \frac{2PR}{P + R}. \quad (25)$$

TABLE 2: The scores of *F-measure* for the human and the three methods.

Method	Human	MS	DRM (initialized by MS)	GDF-Ncut
<i>F-measure</i>	0.79	0.65	0.66	0.67

Table 2 shows the *F-measure* scores of MS [15], DRM (initialized by MS) [21], and GDF-Ncut, demonstrating that GDF-Ncut is more advantageous to partition images into more homogenous regions. In Table 1, we also record the computational time to evaluate time efficiency of the three algorithms. It is obviously seen that MS has higher time complexity compared with GDF-Ncut. DRM also consumes more time for segmenting each image than GDF-Ncut since DRM adopt MS to obtain its initial segmentation. From the experimental analysis above, it is clearly concluded that the proposed algorithm owns more competitive time complexity in comparison with other two rivals while preserving meaningful image segmentation.

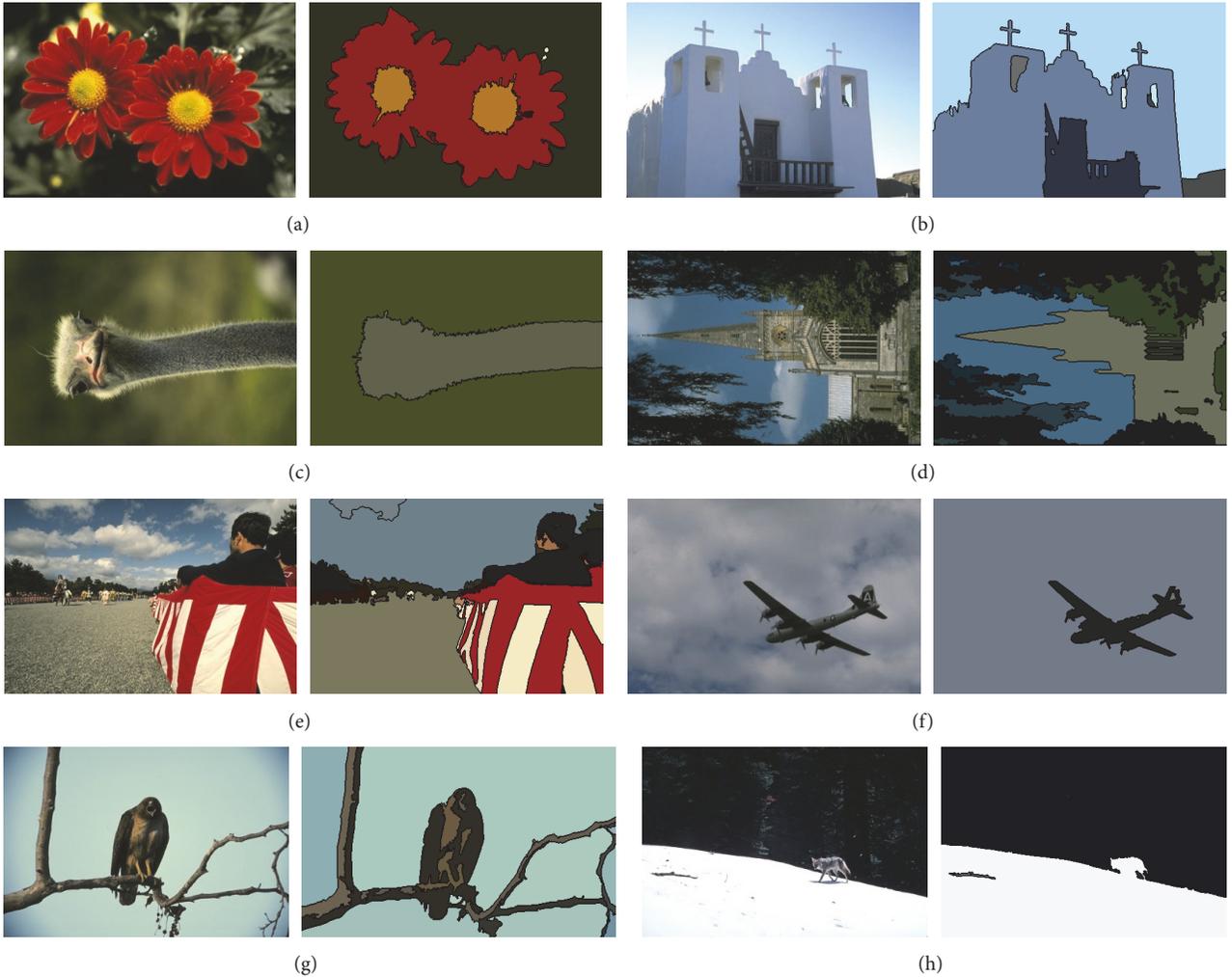


FIGURE 11: Segmentation results of test images. From left to right, (a)–(h) left panels show original images; (a)–(h) right panels present segmentation results of the images by the proposed algorithm.

4.3. Performance of the Proposed Algorithm by Examples. Examples from the train set of BSDS are used to illustrate the segmentation of the proposed algorithm, as shown in Figure 11. These images have been evidently partitioned into meaningful regions. Seen from Section 3, the proposed GDF-Ncut algorithm is determined by three parameters, N , c , and cut . In particular, the parameter c is tuned to 1 in most cases and it is tune to 0.5 only when the colors of the objects are not easily distinguishable. The value of N controls the granularity of cell mesh, which decides the results of the initial segmentation. The cut is the threshold of the region-level $Ncut$, determining when to terminate the segmentation process. In general, we thus adjust the value of parameter N to get appropriate initial segmentation and fix the cut value to cease the merging of improved $Ncut$. Table 3 shows the values of three parameters for segmenting image (a)–(h).

Base on the analysis in Sections 4.2 and 4.3, the advantages of GDF-Ncut can be summarized in the following aspects:

- (a) Initialize regions with GDF. Using hierarchy-grid, the initialization procedure is quickly implemented to

TABLE 3: The values of parameters for segmenting images in Figure 11.

Figure	N	c	cut
(a)	12	1	0.25
(c)	12	1	0.25
(e)	22	1	0.2
(g)	22	1	0.05
(b)	20	1	0.2
(d)	12	0.5	0.25
(f)	16	1	0.15
(h)	22	1	0.02

obtain reasonable but oversegmented segmentations. Besides, the final results are not sensitive to the choice of parameters N and c .

- (b) Optimize the initial segmentation results with improved $Ncut$ algorithm. Combining region-level $Ncut$ with a fast merging procedure, the improved $Ncut$ can

improve the quality of image segmentation with lower time complexity.

5. Conclusion

In this paper, a novel algorithm, GDF-Ncut, has been proposed for color image segmentation by integrating two algorithms: generalized data field (GDF) and normalized cuts (Ncut). In GDF, a nonparametric clustering method based on hierarchal grids is presented to partition an image into disjoint regions. However, the GDF algorithm is prone to attain small fragments of being logically homogeneous. To overcome this defect, the improved region-level Ncut is employed to modify segmentation results of GDF. We conduct an experiment to test the effectiveness of the proposed algorithm, which demonstrates that the GDF-Ncut segments images into meaningful regions compared with other alternatives. Furthermore, the proposed algorithm significantly reduces time complexity because clustering in GDF is implemented on the basis of hierarchy-grid division and the merging process is region-based in Ncut.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors thank Qiliang Chen and Jinfei Yin for helpful programming and discussions. This study is supported by the Foundation of Shandong Province (Grant no. ZR2017PF013) and Science and Technology Development Planning Fund of Shandong Province (Grant no. 2014GGX101048).

References

- [1] J. Yu, D. Huang, and Z. Wei, "Unsupervised image segmentation via Stacked Denoising Auto-encoder and hierarchical patch indexing," *Signal Processing*, vol. 143, pp. 346–353, 2018.
- [2] K. Li, W. Tao, X. Liu, and L. Liu, "Iterative image segmentation with feature driven heuristic four-color labeling," *Pattern Recognition*, vol. 76, pp. 69–79, 2018.
- [3] S. Niu, Q. Chen, L. de Sisternes, Z. Ji, Z. Zhou, and D. L. Rubin, "Robust noise region-based active contour model via local similarity factor for image segmentation," *Pattern Recognition*, vol. 61, pp. 104–119, 2017.
- [4] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object proposal generation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 1, pp. 128–140, 2017.
- [5] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille, "Semantic image segmentation with task-specific edge detection using CNNs and a discriminatively trained domain transform," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '16)*, pp. 4545–4554, July 2016.
- [6] L. Lin, P. Luo, X. W. Chen, and K. Zeng, "Representing and recognizing objects with massive local image patches," *Pattern Recognition*, vol. 45, no. 1, pp. 231–240, 2012.
- [7] A. Angelova and S. Zhu, "Efficient object detection and segmentation for fine-grained recognition," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '13)*, pp. 811–818, June 2013.
- [8] K. Wang, L. Lin, J. Lu, C. Li, and K. Shi, "PISA: pixelwise image saliency by aggregating complementary appearance contrast measures with edge-preserving coherence," *IEEE Transactions on Image Processing*, vol. 24, no. 10, pp. 3019–3033, 2015.
- [9] L. Dong, N. Feng, and Q. Zhang, "LSI: latent semantic inference for natural image segmentation," *Pattern Recognition*, vol. 59, pp. 282–291, 2016.
- [10] L. Zhu, Y. Chen, Y. Lin, C. Lin, and A. Yuille, "Recursive segmentation and recognition templates for image parsing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 359–371, 2012.
- [11] K. Zhang, L. Zhang, K.-M. Lam, and D. Zhang, "A level set approach to image segmentation with intensity inhomogeneity," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 546–557, 2016.
- [12] S. Zhou, J. Wang, S. Zhang, Y. Liang, and Y. Gong, "Active contour model based on local and global intensity information for medical image segmentation," *Neurocomputing*, vol. 186, pp. 107–118, 2016.
- [13] M. W. Ayeche and D. Ziou, "Terahertz image segmentation using k-means clustering based on weighted feature learning and random pixel sampling," *Neurocomputing*, vol. 175, pp. 243–264, 2016.
- [14] R. Hettiarachchi and J. F. Peters, "Voronoi region-based adaptive unsupervised color image segmentation," *Pattern Recognition*, vol. 65, pp. 119–135, 2017.
- [15] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [16] Z. Zhu, Y. Wang, and G. Jiang, "Unsupervised segmentation of natural images based on statistical modeling," *Neurocomputing*, vol. 252, pp. 95–101, 2017.
- [17] S. K. Choy, S. Y. Lam, K. W. Yu, W. Y. Lee, and K. T. Leung, "Fuzzy model-based clustering and its application in image segmentation," *Pattern Recognition*, vol. 68, pp. 141–157, 2017.
- [18] J. Hou, W. Liu, X. Ex, and H. Cui, "Towards parameter-independent data clustering and image segmentation," *Pattern Recognition*, vol. 60, pp. 25–36, 2016.
- [19] W. B. Tao, H. Jin, and Y. M. Zhang, "Color image segmentation based on mean shift and normalized cuts," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 5, pp. 1382–1389, 2007.
- [20] P. Ghosh, K. Mali, and S. K. Das, "Superpixel segmentation with contour adherence using spectral clustering, combined with normalized cuts (N-Cuts) in an iterative k-means clustering framework (NKSC)," *International Journal of Engineering and Future Technology*, vol. 14, no. 3, pp. 23–37, 2017.
- [21] P. Bo, L. Zhang, and D. Zhang, "Automatic image segmentation by dynamic region merging," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3592–3605, 2011.
- [22] D. G. Yan, *Physics*, Higher Education Press, 6th edition, 2016.
- [23] S. Wang, Y. Li, W. Tu, and P. Wang, "Automatic quantitative analysis and localisation of protein expression with GDF," *International Journal of Data Mining and Bioinformatics*, vol. 10, no. 3, pp. 300–314, 2014.

- [24] S. Wang, Y. Li, and D. Wang, "Data field for mining big data," *Geo-Spatial Information Science*, vol. 19, no. 2, pp. 106–118, 2016.
- [25] J. F. Yang, *Atomic Physics*, Higher Education Press, Beijing, China, 2013.
- [26] S. Sarkar and K. L. Boyer, "Quantitative measures of change based on feature organization: eigenvalues and eigenvectors," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 478–483, June 1996.
- [27] S. Zahid, K. N. Whyte, E. L. Schwarz et al., "Feasibility of using patient-specific models and the "minimum cut" algorithm to predict optimal ablation targets for left atrial flutter," *Heart Rhythm*, vol. 13, no. 8, pp. 1687–1698, 2016.
- [28] C. Fernandez-Maloigne, F. Robert-Inacio, and L. Macaire, *Digital Color Imaging*, John Wiley and Sons, Hoboken, NJ, USA, 2012.
- [29] S. Westland, C. Rijpmonti, and V. Chuenq, "CIELAB and colour difference," in *Computational Color Imaging Using Matlab*, Imaging Science and Technology, The Wiley-IS&T, 2nd edition, 2012.
- [30] M. I. Borrajo, W. González-Manteiga, and M. D. Martínez-Miranda, "Bandwidth selection for kernel density estimation with length-biased data," *Journal of Nonparametric Statistics*, vol. 29, no. 3, pp. 636–668, 2017.
- [31] D. M. Bashtannyk and R. J. Hyndman, "Bandwidth selection for kernel conditional density estimation," *Computational Statistics & Data Analysis*, vol. 36, no. 3, pp. 279–298, 2001.
- [32] C. J. Van Rijsbergen, *Information Retrieval*, Butterworths, 2nd edition, 1979.

