Research Article

Salient Object Detection Based on Weighted Hypergraph and Random Walk

WeiYi Wei and Hui Chen

College of Computer Science and Engineering, Northwest Normal University, Lanzhou, Gansu 730070, China

Correspondence should be addressed to Hui Chen; ch19930628@163.com

Received 3 November 2019; Revised 4 May 2020; Accepted 5 June 2020; Published 23 July 2020

Academic Editor: Francesco Aggogeri

Copyright © 2020 WeiYi Wei and Hui Chen. 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.

Recently, salient object detection based on the graph model has attracted extensive research interest in computer vision because the graph model can represent the relationship between two regions better. However, it is difficult to capture the high-level relationship between multiple regions. In this algorithm, the input image is segmented into superpixels first. Then, a weighted hypergraph model is established using fuzzy C-means clustering algorithm and a new weighting strategy. Finally, the random walk algorithm is used to sort all superpixels on the weighted hypergraph model to obtain the salient object. The experimental results on three benchmark datasets demonstrate that the proposed method performs better than some other state-of-the-art methods.

1. Introduction

In computer vision, salient object detection is one of the most fundamental problems, which automatically identifies important and informative regions of an image or video based on human visual mechanisms. In recent years, with the development of artificial intelligence, the salient object detection has attracted the attention of more and more researchers, and many salient object detection algorithms have been proposed [1–3]; these methods are mainly applied to image segmentation [4], image and video compression [5], target extraction [6], image classification [7], and other important fields.

As the research deepens, the graph model has been gradually applied to salient object detection. Zhu et al. [8] propose a weighted manifold ranking algorithm based on unsupervised learning of the simple graph model, which shows high efficiency in salient object detection tasks. Zhang et al. [9] improve the traditional simple graph model and proposes an algorithm based on the new simple graph model and apply it to salient object detection. However, since the simple graph model can only connect any two regions in the image, it is difficult to capture the high-level relationship between multiple regions, which makes the detection results

inaccurate. In order to overcome the shortcomings of the simple graph model, the hypergraph model was introduced into the field of salient object detection. Li et al. [10] perform salient object detection through support vector machine and contextual hypergraph models, in which the hypergraph model transforms the problem of detecting salient objects into the problem of locating salient vertices and edges. However, due to the fact that the hypergraph model established by this algorithm has no adaptivity to R , G , and B values, this method always has good performance for the images with a wide range of pixel values (for example, covering almost the entire range of $[0, 255]$). In order to improve the adaptivity of hypergraph models, Han et al. [11] propose a salient object detection algorithm based on adaptive multiscale hypergraph models, which can build corresponding hypergraph models adaptively according to the range of R , G , and B channels of the image pixel values. Although the adaptivity of the hypergraph model is considered in [11], the influence of color on salient objects is only considered in the hypergraph models in them. In addition, the fact that each vertex has a different importance to salient objects is ignored in the hypergraph models, which causes the salient objects to be incomplete. According to the existing problems of the salient object detection models, in

this paper, a novel salient object detection algorithm is proposed based on a weighted hypergraph model and random walk. The algorithm consists of three steps:

- (1) In order to ensure the integrity of image structural information (a block containing complete information about a part of an object in an image, such as human eyes, a nose, etc.), image is divided into superpixels with the simple linear iterative clustering (SLIC) algorithm.
- (2) To overcome the relationship constraint of the simple graph model, a weighted hypergraph model is constructed, using the fuzzy C-means (FCM) clustering algorithm to consider the global spatial relationship and color similarity.
- (3) The vertices are ranked to obtain the saliency maps by using random walk algorithm on the weighted hypergraph model.

The remainder of this paper is organized as follows. Section 2 reviews existing salient object detection models. Section 3 describes the construction process of the weighted hypergraph model. Section 4 presents the process of detecting the saliency maps based on the weighted hypergraph model. Section 5 shows the experimental results, including a thorough comparison with two salient object detection algorithms and a detailed analysis of the components in the algorithm. Finally, Section 6 concludes the proposed work.

2. Related Work

The fundamental technology of computer vision includes many kinds of salient object detection models. The existing salient object detection models are mainly divided into two categories: traditional models and deep models. Traditional salient object detection algorithms are usually constructed on the spatial domain and frequency domain. In the algorithms on spatial domain, Itti et al. [12] first propose the algorithm based on the biological model to detect the salient objects which integrate the bottom features, the center-surround mechanisms, and three multiscale feature maps through binary Gauss pyramid. Liu et al. [13] make use of the center-surround mechanisms to regard the salient object detection as an image segmentation problem and detect salient objects with the conditional random field. Cheng et al. [14] propose a binarized normed gradients (BING) feature to search for salient objects by using objectness scores. It shows how the BING feature be used for efficient objectness estimation of image windows, which is motivated by the fact that objects are stand-alone things with well-defined closed boundaries and centers. In their work, in order to efficiently quantify the objectness of an image window, they resize it to 8×8 and use the norm of the gradients as a simple 64D feature for learning a generic objectness measure in a cascaded support vector machine (SVM) framework. Nawaz et al. [15] segment image to form fast FCM membership maps by improved FCM algorithm, and they blend these maps by using the Porter-Duff

compositing method to extract salient objects. Zhou et al. [16] propose a novel framework to improve the saliency detection results generated by existing video saliency models. They detect salient objects in the video through local estimation, spatiotemporal refinement, and saliency updates. Song et al. [17] propose a new depth-aware salient object detection framework via multiscale discriminative saliency fusion (MDSF) and bootstrap learning for RGBD. They use random forest regressor and SVM to detect salient objects based on low-level featured contrasts, mid-level feature weighted factors, and high-level location priors. The contrast was later found to be the biggest factor affecting human visual attention [18, 19], so many contrast-based salient object detection algorithms have emerged. Achanta et al. [20] propose a salient object detection algorithm based on multiscale local contrast. Cheng et al. [19] estimate salient objects by using global contrast and spatial weighted correlation. Zhang et al. [21] propose a salient object detection method via background prototypes contrast, in which the regions far from the image center are selected as the background prototype regions, and the salient objects in the image are extracted by calculating the color contrast between any region in the image and the background prototype regions. In the current research, people always combine global contrast and local contrast to study the salient object detection. Therefore, a salient object detection algorithm based on the contrast optimized manifold ranking is proposed by Xie et al. [22]. Liu et al. [23] propose a superpixel-based spatiotemporal saliency model for saliency detection in videos, in which they extract motion histograms and color histograms at the superpixel level and frame level as the local features and global features, respectively, to detect the salient objects in videos. With the wide application of graph theory, the graph model has been introduced into the field of salient object detection. Based on graph model, Ji et al. [24] propose a bottom-up salient object detection method that uses the geodesic distance between image features to construct the affinity matrix and a Laplacian matrix and uses the manifold ranking and multilayer cellular automata to form the saliency maps. Lu et al. [25] propose a multigraph structure for salient object detection. Liu et al. [26] propose a novel framework termed “saliency tree” to detect salient objects. Ye et al. [27] propose an effective salient object segmentation method via the graph-based integration of saliency and objectness. They use the superpixels of the input image to construct the graph model and assign weights to edges by the difference between superpixels. Then by calculating the shortest path, this method can estimate the possibility that each superpixel becomes a saliency region. Finally, they use this possibility and the graph model to form the final saliency maps. Liang et al. [28] propose a new approach to detect salient objects from an image by using content-sensitive hypergraph representation and partitioning. Through analyzing the edge distribution in an image to extract polygonal potential Region-of-Interest, they propose a new content-sensitive method for feature selection and hypergraph construction to detect salient objects. Zhang et al. [29], based on local spatial correlation, global spatial correlation, and color correlation, construct a probabilistic hypergraph to

represent the relations among vertices from different views, and exploit the foreground and the background queries to uniformly highlight the salient objects and suppress the background. Later, based on the original work [29], they propose a new optimized method to detect salient objects [30]. This method is based on the hypergraph model and foreground and background queries. Different from the spatial domain, the algorithm is based on the frequency domain. The most representative ones are the residual spectrum algorithm proposed by Hou and Zhang [31] and the frequency tuning algorithm proposed by Achanta et al. [32]. The frequency domain provides another platform for salient object detection. The salient object detection method based on the spatial-frequency domain hybrid analysis proposed by Yue et al. [33] enables salient object detection to be studied simultaneously in spatial and frequency domains.

The rise of artificial intelligence has pushed deep learning and machine learning to another climax, and the development of these two disciplines has promoted the development of salient object detection, so a series of salient object detection algorithms based on machine learning is proposed. Jiang and Crookes [34] introduce a continuous Markov random field to simulate visual saliency, which is similar to the work of simulating deep propagation along the visual cortex (synaptic communication chain). In deep learning, the human visual and cognitive systems involved in the visual attention process consist of interconnected layers of neurons. For example, the human visual systems have simple and complex cell layers, and their activation is determined by the size of the input signal falling into their receptive field. Since the deep artificial neural network was initially inspired by the biological neural network, it is a natural choice to use the deep artificial neural network to construct a computational model for predicting visual saliency. In deep learning, because the convolution layers in convolutional neural networks (CNN) are similar to the simple and complex cells in the human visual systems [35], the fully connected layers in CNN are similar to the higher-level reasoning and decision-making in the human cognitive systems [36] and are more suitable for salient object detection. In recent years, salient object detection algorithms based on deep learning and CNN have emerged in an endless stream, such as the method of image salient object detection based on deep learning proposed by Zhang [37].

Although existing salient object detection algorithms have good performance, there are still some problems such as inaccurate detection results or incomplete salient objects when the image background is complex or the contrast between foreground and background is not obvious. Aimed at the above problems, this paper proposes a salient object detection algorithm based on the weighted hypergraph model and random walk. Firstly, a weighted hypergraph model is built by using the FCM algorithm at the superpixel level to obtain more complete structural information in the image for the integrity of salient object detection. Secondly, the random walk algorithm is applied to the weighted hypergraph model to increase the contrast between the foreground and background of the image by sorting the superpixels, which is very effective in improving the

accuracy of detection results when the contrast between foreground and background is not obvious.

Thirdly, the pixel-level saliency maps are obtained by a mapping rule. The flowchart of the algorithm is demonstrated in Figure 1.

3. Construction of the Weighted Hypergraph Model

For this algorithm, the weighted hypergraph model is the key. Constructing an efficient and accurate weighted hypergraph model is of great significance for accomplishing salient object detection tasks. This paper uses FCM algorithm and an innovative weighting strategy to construct a weighted hypergraph model.

3.1. Theoretical Knowledge of the Weighted Hypergraph Model.

In a simple graph model, a sample (usually a pixel or superpixel) is represented by a vertex, and the edge connecting two vertices indicates their relationship. Since the simple graph model can only represent the relationship between two regions in the image, it is easy to cause the loss of high-level information between multiple regions. Therefore, this paper introduces the hypergraph model to avoid this problem. The hypergraph models are a generalization of the graph models, while a simple graph model is a special form of the hypergraph model.

Hypergraph models are divided into general hypergraph models and weighted hypergraph models. The difference between hypergraph models and simple graph models is the number of vertices on each edge. Simple graph models have only two vertices on each edge, while the hypergraph models have $n(n \geq 1)$ vertices on each edge, which is named "hyperedge." The hyperedge set E and the vertex set V constitute a general hypergraph model $HG(V, E)$. The association matrix $H \in \mathbb{R}^{|V| \times |E|}$ is used to represent the relationship between hyperedge $e \in E$ and vertex $v \in V$. H is defined as follows:

$$h(v, e) = \begin{cases} 1, & \text{if } v \in e, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Figure 2 is an example of simple graph models and hypergraph models. Among them, the association matrix H in (c) is the tabular form of the hypergraph model (b). In (c), e_i ($i = 1, 2, 3, 4$) represents the hyperedge, and v_i ($i = 1, 2, \dots, 6$) represents the vertex; if the vertex v_i belongs to e_j , it is 1; otherwise, it is 0.

It can be seen from formula (1) that, in the general hypergraph models, 0 or 1 is used to determine whether a vertex belongs to a hyperedge, and all the vertices in each hyperedge have the same importance. This binary model ignores the fact that the samples in the image are of different importance to the salient regions, so a weighted hypergraph model is introduced. Let $HWG = (V, E, w(v_e), w(e))$ denote a weighted hypergraph model, where $w(v_e)$ represents the weight value of vertex $v: v \rightarrow \mathbb{R}^+$ on the hyperedge $e: e \rightarrow \mathbb{R}^+$ to which it belongs, $w(e)$ represents the weight value of hyperedge e , and the weighted hypergraph model is

FIGURE 1: Framework of our model for salient object detection.

FIGURE 2: Simple graph model and hypergraph model example: (a) simple graph model, (b) hypergraph model, and (c) association matrix H .

represented by the associative matrix $H_W \in \mathbb{R}^{|V| \times |E|}$ as follows:

$$h_w(v, e) = \begin{cases} w(v_e), & \text{if } v \in e, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

In addition, the degree of the vertex $d(v_w)$ and the degree of the hyperedge $d(e_w)$ in weighted hypergraph models are defined as

$$d(v_w) = \sum_{e \in E} w(e)h(v, e), \quad (3)$$

$$d(e_w) = \sum_{v \in V} h_w(v, e). \quad (4)$$

The difference between the general hypergraph models and the weighted hypergraph models is whether the weight values are assigned to vertices and hyperedges, respectively.

3.2. Construction Process of the Weighted Hypergraph Model.

The appropriate weighted hypergraph model can make the results more accurate. The main ideas for constructing a weighted hypergraph model in the algorithm are as follows: (i) A general hypergraph model is constructed by the FCM algorithm. (ii) A weighted hypergraph model is constructed using a weighted strategy to assign weight values to the vertices and hyperedges in the general hypergraph model. In

the clustering results, each class is connected by an edge to form a hyperedge. The number of hyperedges is the number of clustering classes, and the number of vertices in each hyperedge is the number of superpixels in each clustering class. In order to extract salient objects from images, they should be assigned to larger weight values. Therefore, in constructing the weighted hypergraph model, the most important thing is how to make weighting rules to assign weight values to the vertices and hyperedges so that the salient objects can be extracted according to the weight values. The weighted hypergraph model construction steps are shown in Figure 3.

3.2.1. Superpixel Segmentation. In human visual systems, image is usually processed with semantic information. In computer vision, superpixel segmentation imitates the preprocessing stage in the human visual systems. A series of adjacent pixels with similar features such as color, brightness, and texture are composed of N small regions, which is called superpixel. Most of the superpixels retain the effective information for further image segmentation and do not destroy the boundary information of the object in the image. Among many superpixel segmentation algorithms, the SLIC algorithm [38] has the advantages of fast segmentation speed, less memory consumption, uniform size of pixel blocks, and better boundary information preservation

FIGURE 3: Process chart for building a weighted hypergraph model.

compared with others. Therefore, the input image is segmented by the SLIC algorithm first.

3.2.2. Construction of the General Hypergraph Model. In this paper, FCM algorithm is used to construct a general hypergraph model. As an improvement of the traditional C-means algorithm, its idea is to maximize the similarity between objects in the same clustering class and minimize the similarity between different classes. In the FCM algorithm, a sample belongs to all classes rather than a certain class, and the membership is used to mark the probability that a sample belongs to a certain class. The FCM algorithm obtains clustering results by iterating its objective function. Its objective function is defined as

$$\min J_m(U, Z) = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m d^2(x_j, z_i), \quad (5)$$

where C is the number of clustering classes, $u_{ij} \in [0, 1]$ denotes the membership of the sample j in class i , and $\sum_{i=1}^C u_{ij} = 1$, $(j = 1, 2, \dots, N)$. $X = \{x_1, x_2, \dots, x_j, \dots, x_N\}$ represents a particular feature dimension matrix of the sample, and $Z = \{Z_1, Z_2, \dots, Z_i, \dots, Z_C\}$ is the clustering center. For each membership, the weight values of the fuzzy degree are controlled by $m \in (1, \infty)$. $d^2 = (x_j, c_i) = |x_j - Z_i|$ is a similarity measure. The u_{ij} and z_i equations are updated as follows:

$$\begin{cases} u_{ij} = \frac{1}{\sum_{k=1}^C (|x_j - z_i|/|x_j - z_k|)^{2/m-1}}, \\ z_i = \frac{\sum_{j=1}^N u_{ij}^m x_j}{\sum_{j=1}^N u_{ij}^m}. \end{cases} \quad (6)$$

The steps of the FCM algorithm are shown as follows:

- (1) Set the precision ε of the objective function, the fuzzy index m (m usually takes 2), and the maximum number of iterations T_m
- (2) Initialize the fuzzy clustering center z_i

- (3) Update fuzzy partition matrix $U = \{u_{ij}\}$ and clustering center $Z = \{z_i\}$ by (6)
- (4) If $|J(t) - J(t-1)| < \varepsilon$ or $C > T_m$, end clustering; otherwise, $t \leftarrow t + 1$ goes to step (3)
- (5) $U = \{u_{ij}\}$ denotes the classification results of each sample

In the traditional FCM clustering algorithm, each sample belongs to all classes. However, a hyperedge cannot contain all vertices generally, so the traditional FCM algorithm is not suitable for the process. Therefore, the following constraints are added to the traditional FCM algorithm:

$$u'_{ij} = \begin{cases} u_{ij}, & u_{ij} \geq T_c, \\ 0, & u_{ij} < T_c, \end{cases} \quad (7)$$

where T_c represents the membership threshold of the sample j belonging to a class i , $U' = \{u'_{ij}\}$ is the membership matrix with the constraint condition (7). In addition, when $j \geq T_c$, the membership of the sample j in the class i is not changed; otherwise, the membership is 0. The important thing is that $\sum_{i=1}^C u'_{ij} < 1$ ($j = 1, 2, \dots, N$). According to the corresponding relationship between the general hypergraph model and clustering results, U' is the general hypergraph model.

From the abovementioned process of using the FCM algorithm to construct the general hypergraph model, the modified FCM algorithm used in this paper is different from that in [15], which satisfies the characteristics of the hypergraph models and sets the values of the membership to less than 0.2 in the clustering results; that is to say, the sample does not belong to this class.

3.2.3. Weight Setting. The weighted hypergraph models are the result of assigning weight values to vertices and hyperedges in a general hypergraph model according to a weighted strategy. In this paper, a weighted hypergraph model based on superpixel is constructed by analyzing two important features that affect the saliency of objects in the image: global spatial relationship and color similarity. The steps of constructing the weighted hypergraph model are as follows: (i) Calculate vertex weight values according to

global spatial relationship and color similarity, respectively. (ii) The relationship between global spatial relationships and color similarity is comprehensively considered to obtain the final weight values of the vertices. (iii) The average weight values of vertices in any hyperedge are used to obtain the hyperedge weight values. The weighting detailed process of vertices and hyperedges is shown in Figure 4.

(1) *Global Spatial Relationship.* In the salient object detection, the distance between the samples is usually used to represent the similarity between the two regions. Based on the research of existing salient object detection algorithms [39], the salient regions often appear at the center of the image. Specifically, when the objects are far away from the center of the image, it is less likely to become a salient object, and vice versa. According to this prior knowledge, the vertex weight values d_w based on the global spatial relationship are defined as

$$d_w = \begin{cases} \frac{1}{\|d_i - d_c\|}, & d_i \neq d_c, \\ 1, & \text{otherwise,} \end{cases} \quad (8)$$

where d_i and d_c represent the position information of the superpixels in the image and the center superpixel, respectively.

(2) *Color Similarity.* Human visual systems are always more sensitive to color features than others, so color features are important in computer vision. If the color of one region is different from others, it is more likely that this region will become a salient region. Since the central region of an image is likely to become a salient region, the possibility that other regions become salient regions can be obtained by calculating the color similarity between these regions and the central region. Accordingly, the vertices weight values c_w based on the color similarity are as follows:

$$c_w = \frac{1}{e^{\|c_i - c_c\|}}, \quad (9)$$

where c_i and c_c are the color information of the image superpixels and the center superpixel on the CIELab color space, respectively. c_w is the result of normalization.

(3) *Weight Values of Vertices and Hyperedges.* In salient object detection, if only spatial relationship or color information is considered, some salient regions will be lost in the saliency maps, so two factors are usually considered together. Although color similarity is considered more important in distinguishing salient objects from the background, global spatial relationship is considered as important as color similarity in distinguishing salient objects from the background because of the particularity of salient objects position (generally located in the image center). Consequently, $\eta = \gamma = 1$ in formula (10). The weight values of vertices in the hypergraph mode are defined as

$$V_w = \sqrt{\eta d_w^2 + \gamma c_w^2}, \quad (10)$$

where, by adding squares of color similarity weight values and global spatial relationship weight values, the distance between the background and salient objects can be increased. The two weights can be fused by the sum calculation with the coefficients, and the values can be constrained within a reasonable range by square root calculation.

In order to prevent the weight value of each hyperedge from being affected by the number of vertices contained, the weight value E_w of the hyperedge is defined as follows:

$$E_w = \frac{U' \cdot V_w}{n}, \quad (11)$$

where U' is the general hypergraph model with constraints condition. V_w denotes the weight values matrix of vertices. n is the number of vertices contained in each hyperedge.

4. Forming Saliency Map with the Walk Algorithm

Here, the random walk algorithm on the weighted hypergraph model is used to rank the importance of the superpixels in the image, and the superpixel-level saliency maps are formed into the final pixel-level saliency maps through a mapping rule.

4.1. Random Walk. The random walk is a special case of the Markov chain and a ranking algorithm suitable for graph models. For a simple graph model, the random walk is as follows: Given a start vertex on a simple graph model, randomly select a neighbor vertex, move it to the neighbor vertex, then take the current vertex as the new start vertex, and repeat the above process. Those randomly selected vertex sequences constitute a random walk sequence. Suppose that these vertices are treated as a state set $\{s_1, s_2, \dots, s_n\}$, and the transition process between vertices is regarded as a Markov chain M that constrains these states; the transition probability $p(u, v) = \text{Prob}(S_{t+1} = v | S_t = u)$ is essentially on the Markov chain M ; when the t -th step is the state u , step $t + 1$ is the probability of state v . In the random walk, the transition probability matrix $p(u, v)$ is used to mark the transition probability between the two vertices. In addition, for any vertex u , there is a relationship $\sum_v p(u, v) = 1$; it means that the probability value sum of the vertex u to all other vertices is 1. In the simple graph models, the process of random walk is very clear. However, the hypergraph model structure is essentially different from that of the sample graph models, so a more general random walk method is needed for hypergraph models. In order to rank the vertices in the hypergraph models, the random walk on the hypergraph models has been generalized by Bellaachia and Al-Dhelaan [40]. In this paper, superpixels are treated as the vertices, and the set of superpixels are hyperedges to construct hypergraph models, which also consider the two key factors of color similarity and spatial relations to weigh vertices and hyperedges. According to the definition of the transition probability matrix, it can be found that the most

FIGURE 4: Specific flowchart for weighting vertices and hyperedges in the hypergraph model.

important supersixels are more significant, so the salient objects can be selected by the random walk algorithm. They define the transition probability matrix as follows:

$$P(u, v) = \sum_{e \in E} w(e) \frac{h(u, e)}{\sum_{\hat{e} \in \mathcal{E}(u)} w(\hat{e})} \frac{h_w(v, e)}{\sum_{v \in e} h_w(\hat{v}, e)}, \quad (12)$$

or in matrix notation as

$$P = D_v^{-1} H W_e D_e^{-1} H_w^T, \quad (13)$$

where $h(u, e)$ is the correlation matrix of the general hypergraph model, and 0 or 1 is used to indicate whether the start vertex u belongs to the hyperedge e . $h_w(v, e)$ is the weight values of the destination vertex v in the hyperedge e . $\sum_{\hat{e} \in \mathcal{E}(u)} w(\hat{e})$ represents the weight values sum of all hyperedges containing the vertex u , and $\sum_{v \in e} h_w(\hat{v}, e)$ refers to the sum of all vertex weight values contained in the hyperedge e . D_v , W_e , and D_e are the diagonal matrix of the vertex degree, the diagonal matrix of the hyperedge weight values, and the diagonal matrix of the hyperedge degree, and the calculation method of the elements in these matrices is as shown in (3), (11), and (4), respectively. H and H_w are the correlation matrices of the general hypergraph model and the weighted hypergraph model, respectively. In order to simplify the calculation, the transition probability matrix $P(u, v)$ needs to be normalized. The random walk steps in the weighted hypergraph model are as follows:

- (1) Input the weighted hypergraph model and the number of iterations D
- (2) Initialize all vertex probabilities to $1/|V|$, forming a vertex initial probability matrix $v_0 \in \mathbb{R}^{|V| \times 1}$
- (3) The iterative update probability matrix is $v_{t+1} = P^T v_t$, forming a new vertex probability matrix v
- (4) If the number of iterations is greater than D or the vertex probability matrix v no longer changes, the random walk process ends
- (5) Output the vertex probability matrix v (the vertices can be sorted according to the probability of each vertex)

In order to ensure the convergence of the random walk algorithm, this paper introduces the PageRank algorithm [41]. If an isolated point (a point in the hypergraph model constitutes a hyperedge called an isolated point) is encountered in the random walk process, the process will end and the algorithm will not converge. However, the PageRank algorithm can make the random walk process jump randomly to any vertices when encountering the isolated point and continue the random walk algorithm; that is, the PageRank algorithm uses the idea of teleporting to restart the random walk process, making it useful for the previous conditions. The teleporting is depicted with a small probability called the damping factor α . It also ensures that the graph is irreducible since the random walker always has the probability of teleporting to any other vertex. The PageRank algorithm represents the probability matrix v as

$$v_{t+1} = \alpha P^T v_t + \frac{(1 - \alpha)e}{N}, \quad (14)$$

where v_{t+1} is a matrix v that is iterated $t + 1$ times, α is the damping coefficient, N is the number of vertices in the weighted hypergraph model (the number of supersixels), and $e \in \mathbb{R}^{N \times 1}$ is a vector of all elements being 1.

4.2. Formation of Saliency Map. The saliency values of each region in the image represent the probability of becoming a salient object. The greater the saliency values, the greater the possibility of becoming a salient region. In this paper, the saliency values are formed by the probability matrix v generated. After the transition probability matrix $P(u, v)$ is normalized, the value in v is mapped to (0, 255) to form the saliency maps. It should be noted that the basic unit of the random walk algorithm in this paper is supersixel, so it is necessary to use the mapping rule to form final gray saliency maps at pixel level. The mapping rule is as follows: The gray values of a supersixel are equivalently assigned to each pixel within this supersixel region. The saliency maps at pixel level are expressed as I_{op} .

Besides, the background regions and salient regions may be reversed in the saliency maps. To avoid this problem, this paper sets the gray threshold T_f and row-column threshold

According to the fact that salient regions always appear in the center of the image. The front-rear k rows and front-rear k columns of the image are selected, and the saliency map I_{op} is corrected as follows:

$$I_{op} = \begin{cases} 255 - I_{op}, & M_w > M_b, \\ I_{op}, & \text{otherwise.} \end{cases} \quad (15)$$

where M_w and M_b represent the number of pixels whose gray value is greater than or equal to the gray threshold T_f in the front-rear k rows and the front-rear k columns, respectively. If $M_w > M_b$ indicates that the background regions in the image are displayed in the form of the salient regions by mistake, this means that the background regions and salient regions are reversed in saliency maps, and then the saliency maps are reversed to form the final saliency maps.

5. Experimental Results and Analysis

The algorithm proposed in this paper and the comparison algorithms are compared through precision-recall (P - R) curve and F -measure. The P - R curve and F -measure are two of the most commonly used indicators in the field of salient object detection. The advantages and disadvantages of the algorithm proposed in this paper are analyzed by the results and indicators comparison in the field of salient object detection.

5.1. Datasets and Comparison Algorithms. The proposed salient object detection algorithm is compared with 2 state-of-the-art algorithms in three datasets. The two comparison algorithms are HM [10] and HAM [11]. On the one hand, the algorithm in [10] is a classic one using the hypergraph model for salient object detection, and comparison with it can make the algorithm proposed in this paper more convincing. On the other hand, the experimental results of [11] are relatively new, and HAM has proved its superiority compared with many classical algorithms, and we further prove the superiority of the algorithm proposed in this paper by comparing it with HAM. The three datasets are MSRA-1000 [32], SED [42], and SOD [43]. The MSRA-1000 dataset contains 1000 natural images. The SED dataset contains two subsets, SED1 and SED2. The image in the SED1 dataset contains one salient object, while the image in the SED2 dataset contains two salient objects. The SOD dataset contains 300 natural images with complex background.

5.2. Parameter Settings. The values of the parameters in our algorithm are obtained based on the actual experimental results. The number of superpixels N is set to 300 in SLIC. The clustering classes C in formula (5) and the membership threshold T_c in formula (7) are set to 3 and 0.2, respectively. The number of iterations D (the number of random walk algorithms performed on the weighted hypergraph model) is set to 1. The damping coefficient α in formula (14) is 0.85 according to [41]. The gray threshold T_f and the row-column threshold k are set to 128 and 10, respectively. The

experimental results of different superpixels and clustering classes are shown in Figure 5.

5.3. Evaluation Metrics. For comprehensive evaluation, this paper uses two metrics, namely, the P - R curve and F -measure. The precision and recall scores are obtained by binarizing the saliency maps with a threshold (0, 255). However, the P - R curves are not intuitive because of two factors that need to be considered in the evaluation, so F -measure is used as the overall performance measure indicator, which is defined as

$$F_\beta = \frac{(1 + \beta^2) \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}}. \quad (16)$$

$\beta^2 = 0.3$ as suggested in [32] to emphasize precision.

5.4. Visual Saliency Contrast Map. To visually evaluate the accuracy of our algorithm, six representative images for evaluating the accuracy of salient object detection are given in Figure 6.

In Figure 6, the first row of images contains one salient object, and the last three rows contain two salient objects. Although the background of the first row image is simple, the salient object is not in the center of the image and the boundary is complex, which makes the salient object detection more difficult. However, it can be seen from the saliency maps that our algorithm can locate the salient object more accurately than the comparison algorithms, but little background regions are detected, mainly because the prior that salient objects are usually located in the image center is taken into account. In the second row of the images, the salient object is located at the center of the image, but due to the complex background, it can be seen from the saliency map that our algorithm lacks robustness when the boundary of the salient objects is complex. The image background of the third to sixth rows is a simple and salient object that is located in the image center, but the contrast between the foreground and background is not sharp. From the saliency maps, it can be seen that the salient object detected by proposed algorithm is more precise and has clearer boundaries. From the results of the last three rows, we can see that our algorithm is better than the comparison algorithms. However, in the seventh row, the background is also detected by our algorithm, which results in the inaccuracy of the detection results. The main reason is that our algorithm uses the idea of central prior and assigns the same saliency values to those pixels in a superpixel, which causes the background in the boundaries to be also regarded as the salient object.

In brief, our algorithm performs satisfactorily when the salient objects are located in the center of the image. For the situation where salient objects are located at image boundaries and where there are multiple salient objects, our algorithm can detect the complete salient objects better than the comparison algorithms.

FIGURE 5: Experimental results of different superpixel numbers and clustering classes: (a) the first row and (b) the second row are the P - R curves and F -measure of different superpixel numbers, respectively; (c) the third and (d) fourth rows are the P - R curves and F -measure of different clustering classes, respectively.

FIGURE 6: Visual saliency maps with different methods: (a) input, (b) GT, (c) ours, (d) HAM, and (e) HM.

5.5. Objective Evaluation. In Figure 7, the six maps, respectively, represent the P - R curve and the F -measure histograms generated by our algorithm and the comparison algorithms on the three datasets.

From the P - R curves, we can see that our algorithm always has a part of the curve higher than the comparison algorithms. According to the meaning of the P - R curve, the

salient object detected by our algorithm is better than the comparison algorithms. In particular, the accuracy of the hypergraph-based HAM algorithm on the MSRA-1000 dataset in Figure 7(a) is close to 97%, but our algorithm is higher than HAM. It also can be seen from the SOD dataset of Figure 7(c) that part of our algorithm curve is slower than the other two comparison algorithms. That is because the

FIGURE 7: Continued.

FIGURE 7: P - R curve and F -measure on the three datasets: (a), (b) the P - R curve and F -measure on the MSRA-1000 dataset; (c), (d) the P - R curve and F -measure on the SOD dataset; (e), (f), (g), (h) the P - R curve and F -measure on the SED dataset.

image background is complex and salient objects of images are often located at image boundaries in SOD dataset; however, the proposed algorithm is based on the center prior that salient objects are always located in the center of the image when weighting vertices and hyperedges in the

hypergraph model. From Figures 7(e) and 7(g), it can be seen that our algorithm outperforms the two comparison algorithms on the two subsets of the SED dataset, mainly because the background of the two subsets is simple and the salient objects are large. However, for the P - R curves in this paper,

TABLE 1: F -measure of different algorithms.

	MSRA-1000	SOD	SED1	SED2
HAM	0.80	0.59	0.72	0.67
HM	0.81	0.58	0.69	0.65
OUR	0.83	0.68	0.75	0.71

the descending trend is always faster than the comparison algorithms, mainly because when the saliency maps at the pixel level are formed, the superpixel saliency values are distributed equivalently. As can be seen from the F -measure histograms, the F -measure of our algorithm is higher than that of the comparison algorithms. The specific F -measure is shown in Table 1. On the SED1 and SED2 datasets especially, our algorithm is better than the HAM algorithm based on the hypergraph models. In general, the results show that our algorithm not only improves the advantages of the sample graph models in salient object detection but also contributes to the field of salient object detection based on hypergraph models.

In summary, combining the P - R curves with the F -measure histograms indicates that the performance of our algorithm is optimal compared with the other algorithms. The saliency maps are closer to the GT maps, the detected salient objects are more complete, and the boundaries are clearer.

6. Conclusions

In this paper, a novel salient object detection algorithm based on a weighted hypergraph model and random walk is proposed. The key of this algorithm is to use the features of salient objects to formulate a new weighting strategy and to build a weighted hypergraph model. Compared with the two state-of-the-art methods on three popular datasets, the proposed method achieves a competitive objective and visual performance. However, there exists an unclear boundary of the salient objects, mainly because the saliency values of the superpixels are equally distributed to all the pixels within one superpixel region, which causes the loss of the pixel information. For future work, we are considering the construction of weighted hypergraph with adaptive method instead of prior knowledge.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant no. 61861040), National Natural Science Foundation of China (Grant nos. 61762080 and 61762078), Science and Technology Plan of Gansu Province (Grant no. 17YF1FA119).

References

- [1] H. Peng, B. Li, H. Ling, W. Hu, W. Xiong, and S. J. Maybank, "Salient object detection via structured matrix decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 818–832, 2017.
- [2] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Salient object detection with recurrent fully convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1734–1746, 2019.
- [3] J. Zhang, F. Malmberg, and S. Sclaroff, "Efficient distance transform for salient region detection," in *Visual Saliency: From Pixel-Level to Object-Level Analysis*, pp. 45–61, Springer, New York, NY, USA, 2019.
- [4] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof, "Saliency driven total variation segmentation," in *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision*, pp. 817–824, Kyoto, Japan, September 2009.
- [5] Y. Fang, Z. Chen, W. Lin, and C.-W. Lin, "Saliency detection in the compressed domain for adaptive image retargeting," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 3888–3901, 2012.
- [6] L. I. Ping-Na and Q. Wu, "Detection and application of improved frequency-tuned salient region," *Modern Computer*, vol. 8, p. 17, 2017.
- [7] B. Lei, E.-L. Tan, S. Chen, D. Ni, and T. Wang, "Saliency-driven image classification method based on histogram mining and image score," *Pattern Recognition*, vol. 48, no. 8, pp. 2567–2580, 2015.
- [8] X. Zhu, C. Tang, P. Wang et al., "Saliency detection via affinity graph learning and weighted manifold ranking," *Neurocomputing*, vol. 312, pp. 239–250, 2018.
- [9] J. Zhang, K. A. Ehinger, H. Wei, K. Zhang, and J. Yang, "A novel graph-based optimization framework for salient object detection," *Pattern Recognition*, vol. 64, pp. 39–50, 2017.
- [10] X. Li, Y. Li, C. Shen, A. Dick, and A. V. D. Hengel, "Contextual hypergraph modeling for salient object detection," in *Proceedings of the 2013 IEEE International Conference on Computer Vision*, pp. 3328–3335, Sydney, Australia, December 2013.
- [11] F. Han, A. Han, and J. Hao, "Saliency detection method using hypergraphs on adaptive multiscales," *IEEE Access*, vol. 6, pp. 29444–29451, 2018.
- [12] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.
- [13] T. Liu, Z. Yuan, J. Sun et al., "Learning to detect a salient object," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 353–367, 2010.
- [14] M. M. Cheng, Z. Zhang, W. Y. Lin, and P. Torr, "BING: binarized normed gradients for objectness estimation at 300 fps," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3286–3293, Columbus, OH, USA, June 2014.
- [15] M. Nawaz, S. Khan, J. Cao, R. Qureshi, and H. Yan, "Saliency detection by using blended membership maps of fast fuzzy-C-mean clustering," in *Eleventh International Conference on Machine Vision (ICMV 2018)*, Article ID 1104123, Munich, Germany, March 2019.
- [16] X. Zhou, Z. Liu, C. Gong, and W. Liu, "Improving video saliency detection via localized estimation and spatiotemporal refinement," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 2993–3007, 2018.

- [17] H. Song, Z. Liu, H. Du, G. Sun, O. Le Meur, and T. Ren, "Depth-aware salient object detection and segmentation via multiscale discriminative saliency fusion and bootstrap learning," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4204–4216, 2017.
- [18] Y.-F. Ma and H. J. Zhang, "Contrast-based image attention analysis by using fuzzy growing," in *Proceedings of the Eleventh ACM International Conference on Multimedia'03*, pp. 374–381, Berkeley, CA, USA, November 2003.
- [19] M. M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, 2014.
- [20] R. Achanta, F. Estrada, P. Wils, and S. Süsstrunk, "Salient region detection and segmentation," in *Proceedings of the International Conference on Computer Vision Systems*, pp. 66–75, Santorini, Greece, May 2008.
- [21] C. H. Luo, W. Zhang, Q. X. Shen, and B. Ye, "Saliency detection via background prototypes contrast," *Computer Measurement and Control*, vol. 25, no. 10, pp. 259–262, 2017.
- [22] C. Xie, H. L. Zhu, X. Lin, and L. Z. Ma, "Salient object detection based on contrast optimized manifold sorting," *Computer Application*, vol. 37, no. 3, pp. 684–690, 2017.
- [23] Z. Liu, X. Zhang, S. Luo, and O. Le Meur, "Superpixel-based spatiotemporal saliency detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 9, pp. 1522–1540, 2014.
- [24] Y. Ji, H. Zhang, K.-K. Tseng, T. W. S. Chow, and Q. M. J. Wu, "Graph model-based salient object detection using objectness and multiple saliency cues," *Neurocomputing*, vol. 323, pp. 188–202, 2019.
- [25] Y. Lu, K. Zhou, X. Wu, and P. Gong, "A novel multi-graph framework for salient object detection," *The Visual Computer*, vol. 35, no. 11, pp. 1683–1699, 2019.
- [26] Z. Liu, W. Zou, and O. Le Meur, "Saliency tree: a novel saliency detection framework," *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 1937–1952, 2014.
- [27] L. Ye, Z. Liu, L. Li, L. Shen, C. Bai, and Y. Wang, "Salient object segmentation via effective integration of saliency and objectness," *IEEE Transactions on Multimedia*, vol. 19, no. 8, pp. 1742–1756, 2017.
- [28] Z. Liang, Z. Chi, H. Fu, and D. Feng, "Salient object detection using content-sensitive hypergraph representation and partitioning," *Pattern Recognition*, vol. 45, no. 11, pp. 3886–3901, 2012.
- [29] J. Zhang, S. Fang, K. A. Ehinger, W. Guo, W. Yang, and H. Wei, "Probabilistic hypergraph optimization for salient object detection," in *Proceedings of the International Conference on Intelligent Science and Big Data Engineering*, Dalian, China, September 2017.
- [30] J. Zhang, S. Fang, K. A. Ehinger et al., "Hypergraph optimization for salient region detection based on foreground and background queries," *IEEE Access*, vol. 6, no. 99, pp. 26729–26741, 2018.
- [31] X. Hou and L. Zhang, "Saliency detection: a spectral residual approach," in *Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Minneapolis, MN, USA, June 2007.
- [32] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk, "Frequency-tuned salient region detection," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1597–1604, Miami, FL, USA, June 2009.
- [33] Y. Juan, Y. Xia, L. Jie, and X. Peng, "RGB-D visual saliency detection method based on spatial-spectral mixture analysis," *Robot*, vol. 39, no. 5, pp. 652–660, 2017.
- [34] R. Jiang and D. Crookes, "Deep salience: visual salience modeling via deep belief propagation," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, Québec City, Canada, July 2014.
- [35] K. Fukushima and "Neocognitron," "Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.
- [36] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5455–5463, Boston, MA, USA, June 2015.
- [37] H. Zhang, "Research on saliency detection of image based on deep learning," M.S. thesis, BJTU, Beijing, China, 2018.
- [38] W. F. Noh and P. Woodward, "SLIC (simple line interface calculation)," in *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics*, pp. 330–340, Holland, Netherlands, July 1976.
- [39] Q. Zhang, "Saliency detection algorithm based on background prior," *Chinese Journal of Image and Graphics*, vol. 21, no. 2, pp. 165–173, 2016.
- [40] A. Bellaachia and M. Al-Dhelaan, "HG-RANK: a hypergraph-based keyphrase extraction for short documents in dynamic genre," in *Proceedings of the 4th Workshop on Making Sense of Microposts*, pp. 42–49, Seoul, Korea, April 2014.
- [41] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, 1998.
- [42] S. Alpert, M. Galun, A. Brandt, and R. Basri, "Image segmentation by probabilistic bottom-up aggregation and cue integration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 315–327, 2011.
- [43] J. Li, M. D. Levine, X. An, X. Xu, and H. He, "Visual saliency based on scale-space analysis in the frequency domain," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 996–1010, 2012.