Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

OSS: Optimized Superpixel Segmentation

Rintu Chacko[1], Divya M Moni[2]
[1,2]*Department of Computer Science, IIET,Nellikuzhi*

Abstract- Optimized superpixel segmentation is an approach using lazy random walk and an energy optimization algorithm. The first step is to initializing the seed positions and runs the lazy random walk algorithm on the input to obtain the probabilities of each pixel. The initial superpixels are iteratively optimized by the new optimization function, which is defined based on the texture measurement LBP. Lazy random walk algorithm has the advantage of segmenting the weak boundaries and complicated texture regions by the global probability maps. The superpixel performance is improved by relocating the seed positions of superpixels and splitting the large superpixels into small ones with the optimization algorithm and for foreground object separation use combination of these two methods.

Keywords- Foreground object separation, Lazy Random Walk, Optimization, Segmentation, Superpixel

I. INTRODUCTION

Superpixels are commonly defined as the grouping of uniform pixels in the image, which have been widely, used in many computer vision applications such as image segmentation and object recognition [10].The main advantage of superpixel is that it provide more natural and meaningful representation of images. Therefore, compared to the traditional pixel representation of the image, the superpixel representation reduces the number of image primitives and improves the representative efficiency of images. The original superpixel concept proposed by Ren and Malik[7] Furthermore, it is more convenient and effective to compute the region based visual features by superpixels, which will provide the important benefits for the vision tasks such as object recognition The desired properties of an ideal superpixel algorithm should not only adhere well to object boundaries of image, but also maintain the compact constrains in the complicated texture regions. Superpixel representation of images and it provide compact representation of image, great improvement in computational efficiency, lower run time and it will save the amount of memory because of the transition from thousands of pixel to few hundreds of superpixels. Thus use of superpixel is essential for image pre-processing in image processing and computer vision.

Superpixels aim to segment the image in to homogenous and uniform regions which is smaller than objects. It is used as an image labeling problem .It can be used as the basis for image segmentation. In order to satisfy desired requirements, develop a new image superpixel segmentation method by the lazy random walk (LRW) and energy optimization algorithm [1] to achieve better performance than the previous approaches. Image superpixel segmentation algorithm is based on the generalized random walk (RW) algorithms [10].

By considering the global relationships between all the pixels and the seed points, develop a superpixel algorithm using the LRW with the compactness constraints. LRW algorithm with self-loops effectively solves the segmentation problem in weak boundary and complex texture regions. On the other hand, the LRW based superpixel algorithm may suffer from the sensitiveness of the initial seed positions. In order to overcome these limitations and improve the performance, further develop a new superpixel optimization approach by using energy optimization framework. Begin with placing the initial seeds of the assigned superpixels.

Then, use the LRW algorithm to obtain the initial superpixels and their boundaries. In order to make the superpixels more compact and their boundaries more consistent with the object boundaries

in image, use energy optimization algorithm to optimize the seed positions and split the large superpixels. Superpixel approach consists of two main steps. The first step is to obtain the superpixels using the LRW algorithm with initial seed points In order to improve the superpixel performance, optimize the initial superpixels by the new energy function in the second step and these two are combined for foreground object separation.

II . RELATED WORKS

Superpixels aim to segment the image in to homogenous and uniform regions which provide more meaningful and natural representation of images. The existing superpixel approaches are classified into two categories one is algorithm that consider compactness constraints during superpixel generation like Ncuts[7], Turbopixel[5], lattice cut[6] and graph cut approaches. The other category algorithms will not consider the compactness constraints for superpixel generations like mean shift[2],and graph based [3] algorithms. The second category algorithm produces the superpixels of highly irregular shapes. The Ncuts superpixel approach obtains the regular size superpixels but its computational cost is high. The Turbopixel algorithm exhibits relatively poor adherence in complicated regions.

For foreground object separation use some methods like Lazy Snapping [9] and GrabCut[8]. Lazy Snapping is an interactive image cutout system, consisting of two steps: a quick object marking step and a simple boundary editing step. Lazy Snapping provides a better user experience. Grab cut interactive extraction of a foreground object in a complex environment whose background cannot be trivially subtracted. The resulting foreground object which reflects the proportion of foreground and background

III. PROPOSED WORK

The purpose of superpixel is to segment the input image into small regions with homogeneous appearance. In this method a unique label is assigned to each superpixel. Here superpixel segmentation process starts with placing initial seeds of the assigned superpixel. Then use a LRW algorithm to obtain initial superpixel boundaries. An energy optimization algorithm is used to make the superpixel more homogeneous and better behavior near object boundaries. This optimization algorithm optimizes the seed position and split the larger superpixels.

Figure 1 shows the workflow of proposed system. It includes two major steps. Initially obtain superpixel boundaries using LRW algorithm by giving initial seed positions. Then improve superpixel performance by optimizing initial superpixel using energy optimization algorithm. Then LRW algorithm is performed to get better boundaries of superpixel with new seed positions. Finally superpixel optimization and LRW executed iteratively to achieve the final superpixel results. For foreground object separation combine these to algorithms. For that first we have to draw a line on input image and take each (x,y) coordinates as seed points. For optimal performance choose 50x50 regions on four directions from seed point and apply segmentation and optimization on each region and lastly combine the resulting segmentation masks to achieve the desired results.

In following subsections we will discuss in detail the LRW algorithm, LRW based superpixel initialization; optimization algorithm and algorithm for foreground object separation.

A. Lazy Random Walk algorithm

The random walk algorithm is used for interactive segmentation in image processing and computer vision applications. It will compute the first arrival probability of one pixel that reaches one of the seeds with each label and that pixel is denoted as same label with maximum probability of corresponding seed. It considers the local relationship between pixel and seeds and ignores the whole relationship between current pixel and other pixels. These random walks suffer from weak boundaries and complex texture segmentations. So in orders to overcome this limitation add self-loops on graph vertex makes random walk process lazy and it will use the global relationship

between pixel and seeds. Adding self loops are based on the concept of LRW.The segmentation results by LRW achieves better foreground object separation than other methods.

Initially a graph G= (V,E) is defined on input image $I(x_i)$ represents a weighted graph containing a set of nodes V and edges $E \subseteq V \times V$ Every pixel x_i is undirected graph and degree of each vertex is computed as $d_i = \sum_j w_{ij}$ for all edges incident on v_i.Edge-weight computation method is used in graph based image segmentation approach and this edge-weights represents the image intensity changes. The edge-weight measure the similarity between neighbouring nodes v_i andv_j.edge-weight w_{ij} defined as a Gaussian weightening function.

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma}\right) \qquad (1)$$

x_i and x_j are the image intensity values of two nodes v_i and v_j.σ is the user defined parameter and fixed to $1/30$ .then using this edge weights define an adjacency matrix A_{ij}

$$A_{ij} = \begin{cases} 1-\alpha & \text{if } i = j \\ \alpha . w_{ij} & \text{if } i \sim j \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

where i~j means node v_i and v_j are adjacent nodes and α is a control parameter and range is (0, 1).in adjacency matrix A majority of elements are zeros, so large size images it requires 16-32 GB RAM for storing the image matrix. To avoid zeros A can be represented as sparse matrix and whose non-zero elements are positive. Then construct another matrix D, diagonal matrix and D_{ij} represents degree of i^{th} node v_i.This means current position v_i will have the probability (1-α) to stay at the current position and probability α to walk out to the arbitrary edges. After constructing diagonal matrix define another matrix $X=D^{-1/2}AD^{-1/2}$ and using this X likelihood l_h is written as

$$l_h = (I - \alpha X)^{-1} y_h \qquad (3)$$

Where I is the identity matrix,$l_h: V \to R$ ia a Nx1 vector and theprobabilities of the node assigned the label h.y_h is a $N \times 1$ column vector as 0,0,….0,1,0….,0 where all elements are zero except the seed pixel and seed pixel represented by 1.then $y_h(x_i) = 1$ if x_i is labeled with h and $y_h(x_i) = 0$ otherwise. Here α set to 0.99 throughout this work to produce sufficient results in boundary adherence.

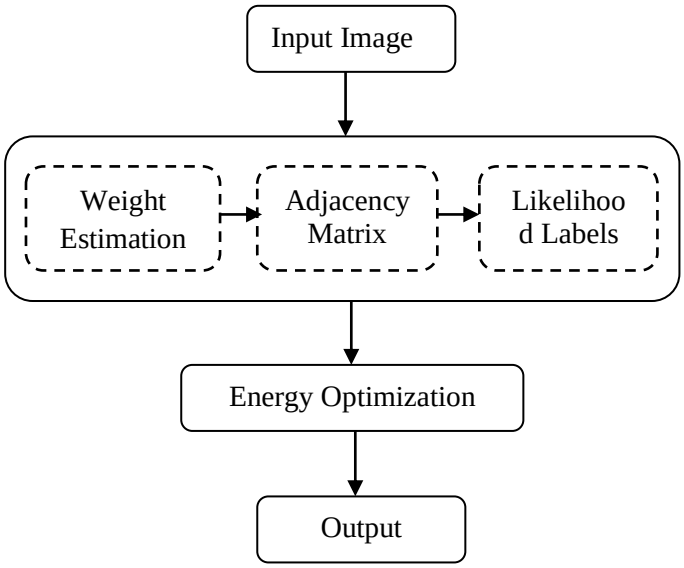


Figure 1: The workflow of optimized superpixel segmentation

B. LRW Based Superpixel Initialization

Begin with placing initial superpixel Seeds on input image and it will use the seed initialization strategy in[4].for that first we have to specify the size of seeds, usually set to 10.the total number of seeds depend on the size of input image.

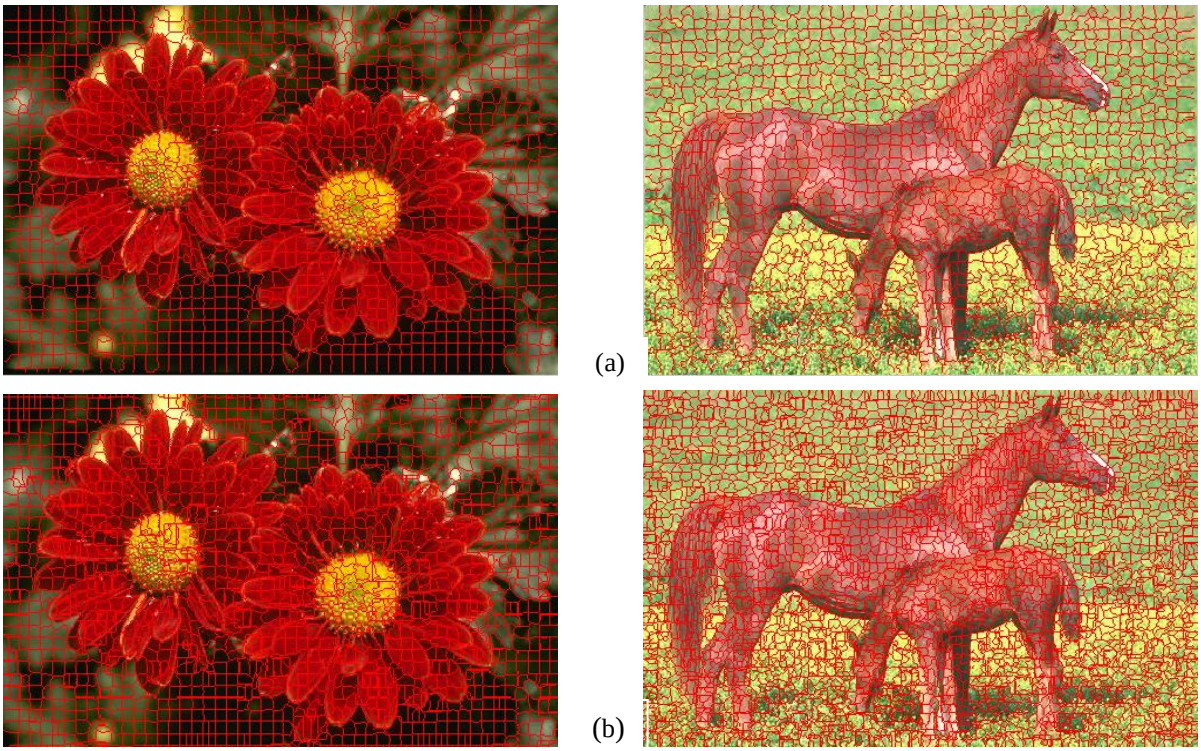


Figure 2: Superpixel results obtained by the proposed optimizes superpixel segmentation on natural images from BSD benchmarks (a) Initial Suprepixel boundaries (Before optimization)(b)After optimization

Then use LRW algorithm to obtain the superpixel boundaries. In each step algorithm transmit to its neighbouring pixel with probability which is proportional to the edge weight w_{ij}.it will group pixel x_i with boundary likelihood probability $l_{h_k}(x_i)$ of superpixel in equation (3).Finally group all those pixels with same labels and label h_k is assigned to each pixel x_i to obtain boundaries of superpixel.

Algorithm 1: LRW Based Superpixel Initialization
Input: Input image $I(x_i)$ and seed size K
Step 1: Find edge weights of each node and using this edge weights define adjacency matrix $A=[w_{ij}]_{m\times n}$
Step 2: Construct the matrix $X=D^{-1/2}AD^{-1/2}$
Step 3: Compute $l_{h_k} = (I - \alpha X)^{-1} y_{h_k}$
Step 4: Obtain superpixel S_{h_k} by grouping pixel with same label where k={1,2,3 ... ... K}.
Output: Initial superpixel result S_{h_k}

C. Superpixel Optimization

An ideal superpixel algorithm makes the superpixel to be regular with uniform size in complicated texture regions. It is used to improve the performance of these superpixels.It will makes the texture information of images to be uniformly distributed in superpixels and makes boundaries of superpixels to more consistent with object boundaries in given input image. For that it will initially find Area(S_h),area of superpixels and Area(S),average area of superpixels. Here the energy has

minimum value when area of superpixel equals to average area of superpixels. Usually the value of Area (S_h), represents the texture information in the superpixel S_h.i.e energy has minimum value when texture information is uniformly distributed in every superpixels.Energy prefer to split the large superpixel with more texture information's in to small superpixel in optimization stage which makes the final superpixel to be more homogenous in complicated texture regions.To measure texture information in each pixel use texture feature of local binary pattern (LBP)[12].The LBP values of each pixel can be defined as

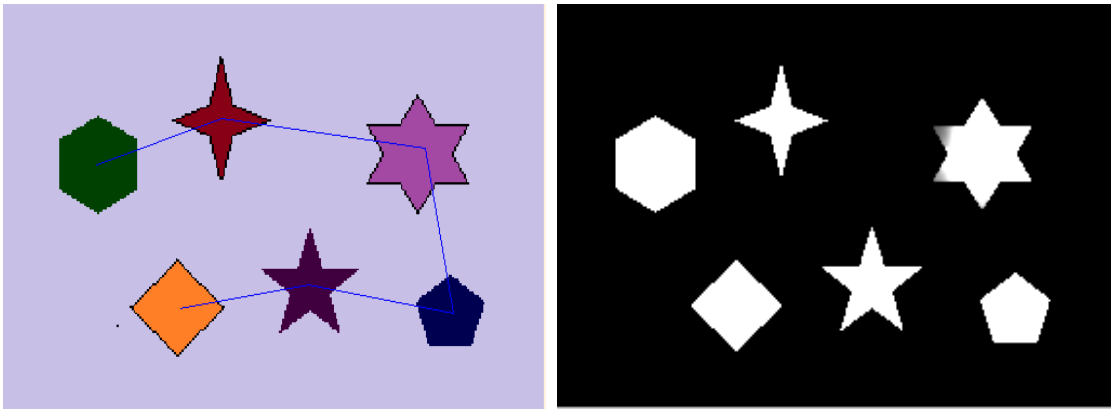
$$LBP_i^{x,r} = \sum_{z=0}^{x-1} s(q_z - q_i)2^z \qquad (4)$$

where x is the gray level of LBP, r radius of the circle around the pixel i , q is the gray value of image. This implementation choose x=8 and r=1.based on these LBP texture feature the area of superpixel is defined as

$$Area(S_h)=\prod_{i\in S_h} LBP_i \qquad (5)$$

$$Area(S)=\sum_{i\in S_h} \frac{LBP_i}{N_{sp}} \qquad (6)$$

where Area(S) is the average area of superpixel and $N_{sp}$is the number of superpixels in the input .The area of superpixel is high when it contains large texture information's. A threshold parameter P_t controls the number of iterations in the optimization process. Large P_t means more iteration is required and gives better superpixel results. Good superpixel result achieves when the value of P_t is in the range of (1.0, 1.5). When $Area(S_h)/Area(S) \geq P_t$,it contain more texture information and the algorithms divide the superpixel whose area is higher than the threshold areas.



(a) (b)

Figure 3: Foreground object separation (a) Seed points (b)Output image

Algorithm 2: Superpixel Optimization
Input: Initial superpixels S_h
Step 1: Find $Area(S_h)/Area(S)/$
Step2: If $Area(S_h)/Area(S) \geq P_t$,split the superpixel into two smaller superpixels.
Step 3: Repeat step 2 for all labeled superpixels
Output: Optimized superpixel results

D. Foreground Object Separation

Both algorithm 1and 2 is combined for foreground object separation. First we have to draw a line on input image and take each (x,y) coordinates as seed points .Apply segmentation and optimization on each regions and merge the resulting segmentation masks to get the required results. For optimal performance take 50×50 regions in four direction form (x,y) coordinates.

Algorithm 3: Foreground Object Separation
Input: Image with multiple objects
Step 1: Draw a line on input image
Step 2: Take (x,y)coordinates as seed points
Step3: Choose 50x50 regions in four direction
Step4: Apply Segmentation and optimization on each region and combine the resulting segmentation mask.
Step 5: Repeat step 3-4 on selected coordinates point.
Output: Objects in input image

IV. RESULTS

The proposed optimized superpixel segmentation process use images from the Berkely segmentation database (BSD). Fig 2 shows the superpixels which are generated by the proposed optimized superpixel segmentation for natural images from the BSD benchmarks. Superpixels boundaries by this approach adhere to the object boundaries very well. Optimization strategy guarantee s the boundaries of superpixels have uniform size and shape as much as possible.

LRW is used to obtain the initial superpixel boundaries. In optimization stage the threshold parameter P_t control the number of iteration required to achieve the final results. The number of superpixel increases after optimization because of the splitting process. The major aim of splitting is to reduce the area of large superpixels to the average area of superpixel. When the value of P_t increases it will spend more time for optimization. Fig 3 shows the foreground object separation results. First draw a line on input images represents by the blue lines and each (x, y) coordinates takes as seed points. Then algorithm 3 is used to get the final results

V. CONCLUSION

We introduce an effective method of superpixel segmentation. It makes an image easier to analyze in the image processing tasks. In this method it will first run LRW in order to get the initial superpixel results. Initial superpixel positions can be obtained by placing the seed positions on the input image. Then optimize the label of superpixel using an energy function. This optimization improves the regularity and adherence performance by relocating the central position of superpixel and divide large superpixel into small uniform regions in an energy optimization algorithm. The performance of superpixel is improved by relocating the center positions of superpixels and dividing the large superpixels into small ones with the optimization algorithm. Finally combination of these two algorithms can be effectively used for foreground object separation

REFERENCES

[1] Jianbing Shen, Yunfan Du, Wenguan Wang, and Xuelong Li,"Lazy Random Walks for Superpixel Segmentation" IEEE Transactions On Image Processing, Vol. 23, No. 4, April 2014
[2] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 5, pp. 603–619, May 2002.
[3] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," Int. J.Comput. Vis., vol. 59, no. 2, pp. 167–181, 2004.

[4] A. K. Sinop and L. Grady, "A seeded image segmentation framework unifying graph cuts and random walks which yields a new algorithm," in Proc. IEEE ICCV, Oct. 2007, pp. 1–8.
[5] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 12, pp. 2290–2297,Dec. 2009.
[6] A. Moore, S. Prince, and J. Warrel, "Lattice cut—Constructing superpixels using layer constraints," in Proc. IEEE CVPR, Jun. 2010,pp. 2117–2124.
[7] X. Ren and J. Malik, "Learning a classification model for segmentation,"in Proc. 9th IEEE ICCV, Oct. 2003, pp. 10–17.
[8] Yin Li," Jian Sun ,Chi-Keung Tang ,Heung-Yeung Shum, Lazy Snapping",ACM Transaction On Graphics volume 23 issue 3 pp 303-308 ,2004
[9] Carsten Rother_ Vladimir KolmogorovAndrew Blake"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts", ACM Transaction On Graphics volume 23 issue 3 pp 309-314 ,2004
[10] L. Grady, "Random walks for image segmentation," IEEE Trans. Pattern Anal. Mach Intell., vol. 28, no. 11, pp. 1768–1783, Nov. 2006.

