Novel Image Superpixel Segmentation Approach using LRW Algorithm

Arpita G.Chakkarwar
Department of Computer Sci. & Engg.
J.C.O.E.T.YAVATMAL

M.V.Sarode, Ph.D
Department of Computer Sci. & Engg.
J.C.O.E.T.YAVATMAL

ABSTRACT
We present a novel image superpixel segmentation approach using the proposed lazy random walk (LRW) algorithm in this paper. Our method begins with initializing the seed positions and runs the LRW algorithm on the input image to obtain the probabilities of each pixel. Then, the boundaries of initial superpixels are obtained according to the probabilities and the commute time. The initial superpixels are iteratively optimized by the new energy function, which is defined on the commute time and the texture measurement.

General Terms
Image processing.

Keywords
Lazy random walk, commute time, optimization, superpixel, texture.

1. INTRODUCTION
Superpixel are commonly defined as contracting and grouping uniform pixels in the image, which have been widely used in many computer vision applications such as image segmentation and object recognition. Compared to the traditional pixel representation of the image, the superpixel representation greatly reduces the number of image primitives and improves the representative efficiency. 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.

It is still challenging to develop a high quality superpixel algorithm, which avoids the under-segmentation and locally groups the pixels respecting the intensity boundaries. 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. In order to satisfy these desired requirements, we develop a new image superpixel segmentation method by the lazy random walk (LRW) and energy optimization algorithm to achieve better performance than the previous approaches. Our image superpixel segmentation algorithm is based on the generalized random walk (RW) algorithms [12]. However, the original RW algorithm depends on the local relationship between the pixel and its corresponding seeds with the first arrival probability. This may lead to the irregular shape of the final non-uniform superpixel results. By considering the global relationships between all the pixels and the seed points, we then develop a novel superpixel algorithm using the LRW with the compactness constraints.

Our 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, we further develop a new superpixel optimization approach by introducing an energy optimization framework. Our superpixel optimization strategy is essentially a compactness constraint, which ensures the resulting superpixels to distribute uniformly with the homogeneous size by relocation and splitting mechanism.

Our energy function is composed of two items, the first data item adaptively optimizes the positions of seed points to make the superpixel boundaries adhere to the object boundaries well, and the second smooth item adaptively divides the large superpixels into small ones to make the superpixels more homogeneous. According to these relocated seed positions and newly created seeds by the splitting scheme, our LRW algorithm is executed again to optimize the initial superpixels, which makes the boundaries of final superpixels adhere to object boundaries very well.

The existing superpixel approaches can be roughly classified into two categories. The first category is the algorithms that do not consider the compactness constrains during the superpixels generation procedure, such as meanshift [5], and graph based [11] algorithms. In order to avoid the superpixels crossing the object boundaries, these segmentation algorithms generally produce the superpixels by over-segmenting the image. Since these algorithms do not consider the compactness constrains, which may produce the superpixels of highly irregular shapes and sizes.

The second category of superpixel algorithms considers the compactness constrains, such as normalized cuts [7], lattice cut, TurboPixels, and graph cut approaches. Ren and Malik [7] proposed an image superpixel approach to segment the image into a large number of small compact and homogeneous regions by the normalized cuts. The NCuts method is very powerful in feature extraction for obtaining the regular superpixels in size and shape. However, the computational cost of NCuts superpixel approach is very high and expensive when the number of superpixels or the size of image increases greatly.

Levinshtein et al presented an efficient TurboPixel superpixel algorithm using the level set based geometric flow evolution from the uniformly placed seeds in the image. However, it exhibited relatively poor boundary adherence because of its numerical stability issues especially with complicated textures. Veksler et al developed an image superpixel approach by the graph cut optimization, and the superpixels were obtained by stitching each pixel that belonged to only one of the overlapping image patches.

been proposed to fulfill the need of increasing applications, such as the algorithms in Moore et al. used a single graph cut method to construct an optimal solution in either the horizontal or vertical direction, which took into account both the edges and the coherence of resulting superpixel lattices. Xiang et al proposed a lattice-like structure of superpixel regions with uniform size by learning the eigen-images from the input image, which improved the evolution speed in the TurboPixel framework. Achanta et al presented a simple linear iterative clustering (SLIC) superpixel algorithm, and adopted the k-means clustering approach to generate the superpixels with relatively lower computational cost. Liu et al formulated the superpixel segmentation problem as an objective function on the entropy rate in the graph. The entropy rate can help to cluster the compact and homogeneous regions, which also favors the superpixels to overlap with a single object on the perceptual boundaries. Zengs et al. presented a structure sensitive image superpixel technique by exploiting the geodesic distance. Recently, Wang et al. proposed edge-weighted centroidal voronoi tessellations (EWCVTs)-based superpixel algorithm, which generated the uniform superpixels and preserved the image boundaries well.

The goal of superpixel is to over-segment the input image into small compact regions with homogenous appearance. The superpixel segmentation can be considered as a pixel labeling problem where each superpixel is assigned to a unique label. Our approach begins with placing the initialized seeds of the assigned superpixels. Then, we use the LRW algorithm to obtain the initial superpixels and their boundaries.

In order to further make the superpixels more compact and their boundaries more consistent with the object boundaries in image, we develop a novel energy optimization algorithm to optimize the seed positions and split the large superpixels. Fig. 1 illustrates the workflow and gives the procedure of the proposed LRW superpixel optimization algorithm. Our superpixel approach consists of two main steps. The first step is to obtain the superpixels using the LRW algorithm with initial seed points [Fig. 1(b)]. In order to improve thesuperpixel performance, we optimize the initial superpixels by the new energy function in the second step. Our energy includes two items: the data item makes the superpixels more homogenous with regular size by relocating the seed positions [Fig. 1(c)], and the smooth item makes the superpixels more adhering to the texture edges by dividing the large irregular superpixels into small regular ones [Fig. 1(f)].

Then, our LRW algorithm is performed again to obtain the better boundaries of superpixels with these new seed positions [Fig. 1(d)]. Finally, our superpixel optimization and LRWsteps are executed iteratively so as to achieve the final satisfying superpixel results [Fig. 1(g)]. In the followingsubsections, we will discuss in detail the LRW algorithm, LRW-based superpixel initialization and optimization algorithm.

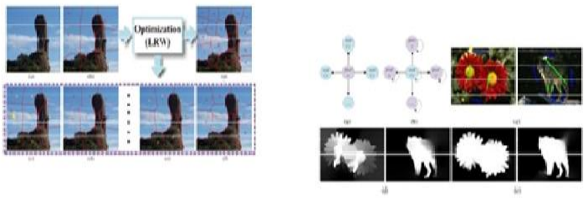

Fig. 1. The workflow of our LRW based superpixel method.

(a) Input image; (b) initial superpixels by LRW and seed points (red "o"); (c) seeds relocation by superpixels optimization (yellow "+" is the relocated seeds from the original positions in (b), and yellow arrow "→" denotes the motion of some seed); (d) superpixel refinement by our LRW method with updated center positions (red "+"); (e) seeds relocation and newly created superpixels with their center positions (green "+") by superpixels optimization; (f) superpixel refinement by LRW; (g) final superpixels. Note that steps (c) to (f) (rectangle with dash lines) are performed iteratively until the final superpixels are obtained

2. Lazy Random Walk Algorithm

The RW algorithm has been used extensively for interactive image segmentation in the image processing and computer vision literatures [12], [14], [22]. The RW algorithm

computes the first arrival probability that a random walk starts at one pixel first reaches one of the seeds with each label, and then that pixel is denoted as the same label with maximum probability of the corresponding seed.

A random walk starts from a pixel must first arrive at the position of the pre labeled seed, and thus it only considers the local relationship between the current pixel and its corresponding seed. This first arrival probability ignores the whole relationship between the current pixel and other seeds. As denoted by Grady [12], these limitations of the original RW method give the reason that it suffers from the weak boundary and complex texture segmentations.

In order to make full use of the global relationship between the pixel and all the seeds, we add the self-loop over the graph vertex to make the RW process lazy, which is inspired by the original LRW concept [3], [12]. However, the original LRW was initially proposed for the website data classifying and mining applications in [3], [8], and [9]. In our application, we need to further develop the original LRW algorithm to be suitable for our image superpixel segmentation application. As shown in Fig. 2, the main contribution of our LRW based superpixel algorithm is two-fold. On one hand, the self-loop [Fig. 2(b)] is added on each vertex to ensure the boundary constrain for superpixels. Since a vertex with a heavy self-loop is more likely to absorb its neighboring pixels than the one with light self-loop, which makes the vertex to absorb and capture both the weak boundary and texture information with self-loops. On the other hand, instead of starting from the pixels to the seed points as the original RW algorithm does [Fig. 2(a)], our LRW algorithm computes the commute time from the seed points to other pixels [Fig. 2(b)]. The probability maps by our LRW approach [Fig. 2(e)] give more confident separation than the ones by RW method [Fig. 2(d)]. Therefore, our LRW algorithm significantly outperforms the original RW algorithm on the test images [Fig. 2(c)] with the same background and foreground seed scribbles. We use the RW implementation2 to produce the RW segmentation results [Fig. 2(f)]. The segmentation result by our LRW algorithm

[Fig. 2(g)] achieves the better foreground objects separation than the result by RW method [Fig. 2(f)].

Fig. 2. Illustration the structure of RW and LRW algorithms with their comparison results. (a) Traditional RW methodwithout self-loops; (b) our LRW algorithm with self-loops; (c) input images with user seeds (scribbles); (d) and (e) are the probability maps by RW and LRW algorithms; (f) and (g) are the segmentation results by RW and our LRW method. Image segmentation result by our LRW algorithm has the better performance than the one by classic RW method [12] with the same user scribbles (green for foreground and blue for background), especially in the leg regions of wolf and the flower parts

3. LRW Based Superpixel Initialization

Our method begins by placing the initial superpixel seeds on the input image. Our goal is to make the superpixels to be evenly distributed over the whole image as much as possible. We first place K circular seeds in a lattice formation, and the distance between lattice neighbors is equal to √N/K where N is the total number of pixels in the image. This strategy ensures that the superpixels will be evenly distributed on the whole image. However, this placement strategy may cause some seeds to occasionally close to a strong edge because these images are not completely uniform distribution. Thus, the initial seed position is perturbed by moving it along its gradient direction according to the seed density.

After we have finished the seed initialization stage, we then use the LRW algorithm to compute the boundaries of superpixels. At each step, the LRW algorithm transmits to its neighborhood with the probability which is proportional to the aforementioned edge-weight wi j. The LRW algorithm will converge at a pixel xi with the boundary likelihood probabilities flk (xi) of superpixels . Finally, we obtain the labeled boundaries of superpixels from the commute time as follows:

$$R(x_i) = \underset{l_k}{argmin}\ CT(c_{l_k}, x_i) = \underset{l_k}{argmax}\ f_{l_k}(x_i)$$

Where clk denotes the center of the l-th superpixel, and the label lk is assigned to each pixel xi to obtain the boundaries of superpixels. Algorithm 1 gives the main steps of our LRW based superpixel initialization algorithm.

Algorithm 1 LRW Based Superpixel Initialization Algorithm

Input: Input image $I(x_i)$ and an integer of initial seeds K

step 1. Define an adjacency matrix $W = [w_{ij}]_{M\times N}$

step 2. Construct the matrix $S = D^{-1/2}WD^{-1/2}$

step 3. Compute $f_{l_k} = (I - \alpha S)^{-1} y_{l_k}$

step 4. Compute $R(x_i) = \underset{l_k}{argmax}\ CT(c_{l_k}, x_i)$ to obtain the labels by assigning label $R(x_i)$ to each pixel x_i.

step 5. Obtain superpixels by $S_{l_k} = \{x_i | R(x_i) = l_k\}$ where $\{i = 1, \cdots, N\}$ and $\{k = 1, \cdots, K\}$

Output: the initial superpixel results S_{l_k}.

4. Superpixel Optimization

As described in the previous paragraphs, the main principle of an ideal superpixel algorithm should contain that the superpixel boundaries adhere well to image intensity boundaries and also make the superpixels to be regular with uniform size in the complicated texture regions. By considering the compactness constraints, we further improve the performance of superpixels with the following energy optimization function.

$$E = \sum_l (Area(S_l) - Area(\bar{S}))^2 + \sum_l \widetilde{W}_x CT(c_l^n, x)^2$$

where the first term is the data item and the second term is the smooth item. The data item makes the texture information of image to be distributed uniformly in the superpixels, which produces more homogeneous superpixels. The smooth item makes the boundaries of superpixels to be more consistent with the object boundaries in the image. Area(Sl) is the area of superpixel and Area(¯S) defines the average area of superpixels. Wx is a penalty function for the superpixel label inconsistency, and we set.

$$\widetilde{W}_x = e^{-CT(c_l, x)/\beta}$$

in our paper where β is a normalization factor. When the commute time CT (cl , x) between seed point cl and pixel x is small, $\widetilde{W}_x$ will be a large value. This makes the optimized superpixel to be more compact and more homogeneous in texture regions

Algorithm 2 Superpixel Optimization Algorithm

Input: Initial superpixels S_l and an integer N_{sp}

step 1. Apply Equation

$$c_l^n = \frac{\sum_l \widetilde{W}_x \frac{CT(c_l^{n-1}, x)}{||x - c_l^{n-1}||} x}{\sum_l \widetilde{W}_x \frac{CT(c_l^{n-1}, x)}{||x - c_l^{n-1}||}}$$

to obtain the new c_l^n

step 2. Apply Equation

$$c_{l_{new},1} = \frac{\sum_{\{(x-c_l)\cdot s>0\}} \widetilde{W}_x \frac{CT(c_l, x)}{||x - c_l||} x}{\sum_{\{(x-c_l)\cdot s>0\}} \widetilde{W}_x \frac{CT(c_l, x)}{||x - c_l||}}$$

$$c_{l_{new},2} = \frac{\sum_{\{x|x\in S_l,(x-c_l)\cdot s<0\}} \widetilde{W}_x \frac{CT(c_l, x)}{||x - c_l||} x}{\sum_{\{x|x\in S_l,(x-c_l)\cdot s<0\}} \widetilde{W}_x \frac{CT(c_l, x)}{||x - c_l||}}$$

to get the new $c_{l_{new},1}$ and $c_{l_{new},2}$

step 3. Refine S_l by Equation

$$R(x_i) = \underset{l_k}{argmin}\ CT(c_{l_k}, x_i) = \underset{l_k}{argmax}\ f_{l_k}(x_i)$$

with c_l^n, $c_{l_{new},1}$ and $c_{l_{new},2}$

step 4. Run steps 1 to 3 iteratively until convergence

Output: the final optimized superpixel results

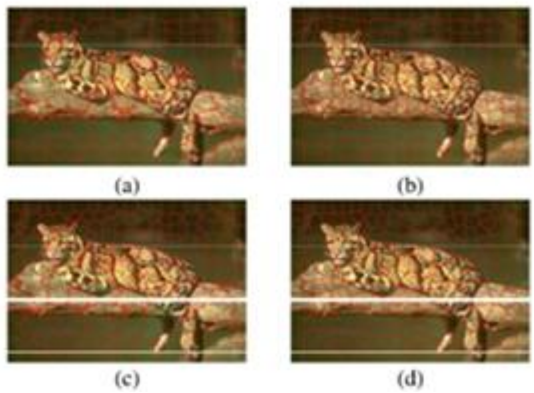

Fig. 3. Comparison results of image superpixels by RW and LRW algorithms. (a) superpixels by RW [12]; (b) superpixels by our LRW; (c) superpixels by RW and optimization; (d) superpixels by our LRW and optimization. Note that the superpixels by our LRW algorithms have the better performance such as the uniform size and good boundary adherence.

5. CONCLUSION

We have presented a novel image superpixel approach using the LRW and energy optimization algorithm in this paper. Our method first runs the LRW algorithm to obtain the initial superpixel results by placing the seed positions on input image. Then we further optimize the labels of superpixels to improve the regularity and boundary adherence performance by relocating the center positions of superpixels and dividing the large superpixels into small uniform ones in an energy optimization framework. The experimental results have demonstrated that our superpixel algorithm achieves better performance than the previous well-known superpixel approaches. Our algorithm is capable of obtaining the good boundary adherence in the complicated texture and weak boundary regions, and the proposed optimization strategy significantly improves the quality of superpixels.

6. REFERENCES

[1] F. R. K. Chung, Spectral Graph Theory. Providence, RI, USA: Amer. Math. Soc., 1997.

[2] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in Proc. 8th IEEE ICCV, Vancouver, BC, Canada, Jul. 2001, pp. 416-423.

[3] D. Aldous and J. Fill. (2002). Reversible Markov Chains and Random Walks on Graphs [Online]. Available:http://statwww.berkeley.edu/users/aldous/RWG/book.html

[4] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in Proc. NIPS, 2002, pp. 849-856.

[5] D. Comaniciu and P. Meer, "Mean shift: A robustapproach toward feature space analysis," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 5, pp. 603-619, May 2002.

[6] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray scale and rotation invariant texture analysis with local binary patterns," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 971-987, Jul. 2002.

[7] X. Ren and J. Malik, "Learning a classification model for segmentation," in Proc. 9th IEEE ICCV, Oct. 2003, pp. 10-17

[8] J. Ham, D. D. Lee, S. Mika, and B. Sch¨olkopf, "A kernel view of the dimensionality reduction of manifolds," in Proc. 21st ICML, 2004, pp. 1-9.

[9] D. Zhou and B. Sch¨olkopf, "Learning from labeled and unlabeled data using random walks," in Proc. DAGM, 2004, pp. 237-244.

[10] G. Mori, X. Ren, A. A. Efros, and J. Malik, "Recovering human body configurations: Combining segmentation and recognition," in Proc. IEEE CVPR, Jul. 2004, pp. 326-333.

[11] P. Felzenszwalb and D. Huttenlocher, "Efficient graphbased image segmentation," Int. J. Comput. Vis., vol. 59, no. 2, pp. 167-181, 2004.

[12] L. Grady, "Random walks for image segmentation," IEEE Trans. Pattern Anal. Mach Intell., vol. 28, no. 11, pp. 1768-1783, Nov. 2006.

[13] L. Grady and E. Schwartz, "Isoperimetric graph partitioning for image segmentation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 3, pp. 469-475, Mar. 2006. [14] 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.

[15] X. Bai and G. Sapiro, "A geodesic framework for fast interactive image and video segmentation and matting," in Proc. IEEE 11th ICCV, Oct.2007, pp. 1-8.

[16] A. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in Proc. IEEE CVPR, Jun. 2008, pp. 1-8.

[17] 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.

[18] D. S. Watkins, Fundamentals of Matrix Computations, 3rd ed. New York, NY, USA: Wiley, 2010.

[19] O. Veksler, Y. Boykov, and P. Mehrani, "Superpixels and supervoxels in an energy optimization framework," in Proc. ECCV, 2010, pp. 211-224.

[20] A. Moore, S. Prince, and J. Warrel, "Lattice cut—Constructing superpixels using layer constraints," in Proc. IEEE CVPR, Jun. 2010, pp. 2117-2124.

[21] s R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fsua, and S. Sässtrunks, "SLIC superpixels," EPFL, Lausanne,Switzerland, Tech. Rep. 149300, 2010.