International Journal For Technological Research In Engineering
Volume 2, Issue 7, March-2015 *ISSN (Online): 2347 - 4718*

IMAGE SEGMENTATION PROCESS BY USING A MODULARITY BASED APPROACH

P. Srinu[1], P. Ravi Kumar[2], R.V.S Vinod Kumar[3], Asst. Prof. V. Nancharaiah[4]
Dept of Electronics & Communication Engineering, 4th year (2011 -2015)
Lendi institute of engg & technology – 535005 AP India
(Affiliated to JNTU Kakinada, A.P., India)

Abstract:* A new approach to address the problem of segmenting an image into sizeable homogeneous regions, this paper proposes an efficient agglomerative algorithm based on modularity optimization. When an over segmented image is given as input which is consisting of several small regions, our proposed algorithm automatically merges those neighboring regions that produce the largest increase in modularity index. When the modularity of the segmented image is maximized, the algorithm stops merging and produces the final segmented output image. To store the repeated patterns in a homogeneous region, we propose an feature based histogram of states of image gradients, and we use it together with the color feature to characterize the similarity matrix in a regenerative manner. By the process of using the proposed algorithm we can avoid the over segmentation problem. This algorithm which is 'proposed is tested on the publicly available Berkeley Segmentation Data Set and Semantic Segmentation Data Set also this results are compared with the previously proposed algorithms. Experimental results have shown that our algorithm is producing sizeable segmentation, preserves repetitive patterns with appealing time complexity, and achieves object-level segmentation to some extent.

***Keywords:* modularity, clustering, image segmentation ,community detection, histogram**

I. INTRODUCTION

Image segmentation refers to a process of partitioning a digital image into N number of parts. The images are segmented on the basis of set of pixels or pixels in a region that are similar on the basis of some homogeneity criteria such as color, intensity or texture which helps to identify and locate objects or boundaries in a image. In terms of mathematical formulae, image segmentation divides a digital image f(x,y) into continuous , disconnect and nonempty subsets f1,f2,f3,……………………..fn, from these subsets high level information can be easily extracted. Practical applications of image segmentation include object identification and object recognition, medical image processing, facial processing, airport security systems ,satellite images and factories outlet images and many more wide applications. Due to importance of image segmentation, large number of algorithms have been proposed but the selection of the image type and the nature of the problem

A. Modularity and community detection:

Modularity was first defined by M.E.J Newman in [25] for the analysis of weighted networks. For a weighted network G with the weighted adjacent matrix A, the modularity Q is defined by:

$$Q=\frac{1}{2m}\sum\left[Ai,j-\frac{kikj}{2m}\right]\delta(Ci,Cj) \quad (1)$$

where Aij represents the weight between node i and node j; m $=\frac{1}{2}\sum ij, Aij$represents the total weights of the network; ki $=\sum j, Aij$ is the weighted degree of the node i; ci is the community label to which node i belongs; δ(ci; cj) is 1 if node i and node j are in the same community, otherwise it's 0. Intuitively, modularity means to evaluate the difference between the actual probability of the connectivity of two nodes in the same community and the estimated probability under the assumption that the two nodes are connected randomly. Community Detection becomes a hot topic in network science during the past few years, for example, social networks. A community is a group of nodes from the network, where nodes in the same community are densely connected with each other, and nodes in different communities are sparsely connected. Communities are of vital importance in a network, since they may represent some functional modules in the network. For example, a community in the social network may represent a group of friends sharing the same hobbies; a community in the citation network may reveal the related work in a certain research area. To uncover the interconnection of the nodes in a network, Community Detection algorithms aim to find a partition of the network such that every partition can well represent certain community property. Since the first proposal of modularity, it has been widely used to evaluate the performance of community detection algorithms and also works as an optimization index for community detection. For example, Louvain method [29] is based on modularity increase to detect the communities. The modularity increase caused by merging community j into community I can be computed by Equation

$$\Delta Q_{ij}=\left[\frac{\sum in+kjin}{2m}-\left(\frac{\sum tot+kj}{2m}\right)^2\right]-\left[\frac{\sum in}{2m}-\left(\frac{\sum tot}{2m}\right)^2-\left(\frac{kj}{2m}\right)^2\right] \quad (2)$$
$$=\frac{1}{2m}\left(kj,in-\frac{\sum tot\, kj}{m}\right),$$

where$\sum in$ in the total weights of the edges inside community i; $\sum tot$ is the total weights of the edges incident to nodes in community i; kj;in is the sum of the weights from community j to community i; other notations are the same as defined in Equation(1). The basic idea of Louvain method is to iteratively repeat the process of Modularity Optimization in Phase 1 and Community Aggregation in Phase 2 below:

Phase 1: Modularity Optimization at the beginning of this phase, the network is composed of several communities (each community is a single node initially, after several iterations, each community is a group of nodes), for each community i and its connected nodes Ni = fjjAij > 0g, compute the potential modularity increase ΔQij if we merge community j (8j 2 Ni) into community i, according to Equation (2). Find the maximum modularity increase caused by merging community j* and community i and merge these two communities. Repeat this process until no modularity increase for all the communities in the network;

Phase 2: Community Aggregation

To reconstruct the network, merge the communities sharing the same label and re label them; treat the communities with the same label as a single node in the network and re compute the weighted adjacency matrix by summing over all the weights connecting two communities. The above processes are repeated until there is no modularity increase caused by merging any two communities

B. Related Approaches for Image Segmentation

Recently, modularity optimization has been applied in image segmentation. [26] explores the possibility of directly applying modularity maximization to image segmentation, where a top down spectral method with an iterative rounding scheme is proposed for fast computation. Such a scheme can reduce the computational cost to some extent, compared with the practically used exchange heuristic [30]. However, it can only deal with images of relatively small size on normal PCs, due to the involved manipulation of a dense modularity matrix. Besides, direct application of modularity maximization to image segmentation is known to result in serious over segmentation. To address the over-segmentation problems, [27] proposes to use a weighted modularity, where the modularity computation only occurs locally within a pre-defined distance. Moreover, an approximation of the Louvain method, the so called basic iteration, is used for faster computation. However, the newly introduced distance parameter depends heavily on the images and the objects. For different images with different object sizes, the distance parameter is ad-hoc, and it is very difficult to choose a universal distance parameter. Both of these two methods focus on how to apply modularity optimization to segmentation, and ignore the differences between community detection and image segmentation. Specifically, both methods start from single pixel, thus, the computational cost, though reduced to some extent by using different computational algorithms, is still too expensive, especially for the first one or two iterations.

II. OUR APPROACH

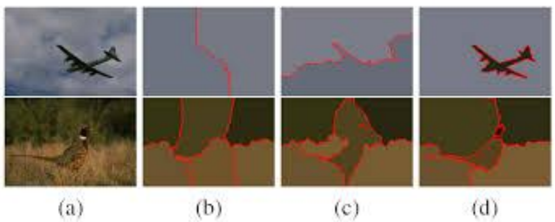
Motivated by the limitations exposed in the existing work, our approach takes the following three aspects into consideration: 1) time complexity; 2) regularity preservation; 3) the prevention of over-segmentation. Inspired by the application of community detection algorithms in large scale networks, we attempt to view an image from the perspective of a network. For a network, modularity [25] is a crucial quantity, which is used to evaluate the performance of various community detection algorithms. In more detail, the larger the modularity of a network is, the more accurate the detected communities are. Considering the efficient calculation of modularity in the community detection algorithm, similarly, we regard image segmentation problem as a community detection problem, and the optimal segmentation is achieved when the modularity of the image is maximized. Although modularity has been applied to image segmentation by some researchers recently, e.g., [26], [27], it still faces similar problems as other segmentation algorithms mentioned above, due to the ignorance of the inherent properties of images (see Section II-B for more details). Different from the existing algorithms based on modularity, we identify the differences between community detection and image segmentation, start from ‘superpixels’, and propose a new texture feature from low level cues to capture the regularities for the visually coherent object and encode it into the similarity matrix; moreover, the similarity among regions of pixels is constructed in an adaptive manner so as to avoid over-segmentation. Compared with other existing segmentation algorithms, our proposed algorithm can automatically detect the number of regions/segments in an image, produces sizable regions with regularities preserved, and achieves better semantic level segmentation to some extent. The contributions of this paper are the following: An efficient agglomerative segmentation algorithm incorporating the advantage of community detection and the inherent properties of images is developed. The algorithm enjoys low time complexity as well as comparable performance; A new texture feature, namely, Histogram of States (HoS) is proposed to capture the regularities in the image. The HoS feature, together with the color feature, encodes better similarity measure from the semantic level, and is more likely to preserve regularities in the object; An adaptive similarity matrix construction is proposed to avoid over-segmentation. In each iteration, the similarity between two regions of pixels is recalculated to reevaluate the color and texture similarity. In this way, it can effectively avoid breaking visually coherent regions, which share some regularities or have smooth changes in color or texture caused by shadow or perspectives.

III. ALGORITHM DESCRIPTION

Image segmentation is related to community detection to some extent. Similar to nodes in the same community, the pixels inside the same segment also share some properties in common, like pixel color value. In this sense, we can treat each homogeneous image segment as a community, and think of image segmentation as a community detection problem. However, due to the inherent properties of images, segmentation is not exactly a community detection problem and directly apply community detection algorithms to image segmentation will lead to awful performance. The differences between image segmentation and community detection can be revealed from the following aspects: 1) different from single node in a community, single pixel cannot capture these regularities in each visually homogeneous segment; 2) the pixels inside the same segment possibly have completely different properties, like color; while for communities, a community is a group of nodes share exactly similar properties. Take the face image as an example, the whole face should be treated as one segment for the purpose of image segmentation. In contrast, for community detection, the eye pixels would be treated as a separate community, while other parts of the face would be treated as another community due to the fact that the pixel color value property of the eyes is totally different from that of other parts of the face; 3) compared with communities, images share some a prior information, say, adjacent regions are more likely to belong to the same segment; 4) as the aggregation process goes on, more pixels are included in one region and the texture inside the region keeps updating, while the properties of the aggregated communities do not change much. To address the above mentioned problems, we propose an efficient agglomerative image segmentation algorithm, taking advantage of the efficient calculation of the modularity optimization in community detection and the inherent properties of images. The algorithm starts from a set of over-segmented regions, thus, runs very fast, and produces sizable segmentation with the regularities inside the same object preserved. The overview of the proposed segmentation algorithm is summarized . And the detailed presentation of some technical points for our algorithm is as follows.

A. Super pixels
The agglomerative algorithm can start the aggregation process by treating each single pixel as a community, however, it turns out that this will be too much time consuming, especially for the first Louvain iteration. Fortunately, this is indeed not necessary, because no texture information is included for single pixel. Therefore, instead, we start with 'super pixels', which can reduce the computational cost as well as capture the regularities. Super pixels are a set of very small and homogeneous regions of pixels. Initializing with super pixels can greatly reduce the time complexity without affecting the segmentation performance. Hence, we first employ a pre-processing step to over segment the image into a set of super pixels. This preprocessing step can be achieved by simple K-Means clustering algorithm (K is set to be a relative large value, e.g., 200 or more) or other super pixels generating algorithms. In our implementation, we use a publicly available code [31] to super pixel initialization. As is shown in the middle of Figure 2, the super pixel generation step usually gives more than 200 over segmented regions on average. This step can greatly reduce the complexity to only consider about 200 nodes in the first iteration for our algorithm. The right column of Figure 2 shows the segmentation result given by our proposed algorithm, where only around 10 homogeneous regions with similar regular patterns inside are left. This fact demonstrates that the segmentation results are indeed the effects of our proposed algorithm rather than the
super pixel generation algorithm.

Algorithm;
Modularity based image segmentation
Input: Given a color image I and its over segmented initialization with a set of super pixels R = fR1; :::;Rng
1: while Pixel labels still change do
2: Reconstruct the neighborhood system for each region in R.
3: Re compute the histogram of states texture feature and estimate the distribution of the color feature for each region.
4: Adaptively update the similarity matrix W according to Equation (3), Wij 6= 0 only if Ri and Rj are adjacent regions in I.
5: while modularity increase still exists by merging any two adjacent regions do
6: for each region Ri 2 R do
7: Compute the modularity increase caused by merging region Ri with any of its neighboring regions according to Equation (2) and find the neighboring region RJ , which gives the largest modularity increase among all of the neighboring regions of Ri.
8: Merge region Ri and region Rj by setting the labels of pixels in these two regions to be of the same label
9: end for
10: end while
11: Update the region labels to get a new set of regions R = fR1; :::;Rmg, where m is current number of regions;
12: end while
Output: The set of image segments R.



B. Choice of Color Space
To capture different aspects of the color, various color spaces are proposed in the literature [32], such as RGB, L*a*b, YUV, HSV and XYZ. To achieve good segmentation performance, the choice of color space is very important. Among all the color spaces, the L*a*b color space is known to be in accordance with human visual system and perceptually uniform, hence the image representation in this color space has been widely used in the field of image processing and computer vision. Due to this facet, all of our discussions of the algorithm are in the L*a*b color space. Later, in the experimental evaluation section, we have also validated that the segmentation performance in L*a*b color space is much better than that in the RGB color space.

C. Neighborhood System Construction
Different from normal networks, such as social networks or citation networks, images have self-contained spatial a priori information, i.e., spatial coherent regions are more likely to

be regarded as a single segment, while regions far away from each other are more likely to belong to different segments. Hence, different from Louvain method where two regions are considered to be neighbors as long as the similarity weight between them are nonzero, we have instead constructed a different neighborhood system by incorporating this spatial a prior information of images. To be specific, we only consider the possibility of merging neighboring regions in the image during each aggregation process. To achieve this, for each region in the image, we only consider the adjacent regions of this region to be its neighbors and store its neighboring regions using an adjacent list. The adjacent regions are defined to be the regions that share at least one pixel with the current region. In the following processes for the similarity matrix construction and aggregation, we only consider the current region and the regions in its neighborhood system.

D. Features for Similarity

Color is the most straightforward and important feature for segmentation, so we use the pixel value in the L*a*b color space as one of the features for computing the similarity.

However, the color feature alone cannot achieve good segmentation performance, since it does not consider the repetitive patterns of different colors in some homogeneous object. For example, in the case of a zebra in Figure 2, the black and white stripe regularities on zebra would be treated as a whole part according to human’s perception. Simply using color feature will break down these regularities into different segments. To address this problem, we not only employ the color feature, but we have also proposed a novel texture feature to capture the regularities in the image. Our proposed texture depicter is motivated by the Histogram of Oriented Gradients (HoG) [33] for pedestrian detection, however, instead of constructing a histogram of gradients, we construct a Histogram of States (HoS) for each region following the three steps below.

1) Compute the gradient magnitude and 360 degree orientation information for each pixel in the image and then eliminate those magnitude without any texture information by thresholding the magnitude information;

2) At each pixel position, use a 5_5 sliding window1, and form a histogram of the oriented gradient by dividing 360 degree orientation into 8 bins, then each pixel is represented by a 8 dimensional binary orientation vector with each dimension indicating whether this orientation exists in the current sliding window. In this sense, the texture information for each pixel belongs to one of the 28 = 256 states;

3) For each region, we construct a 256 dimensional histogram of states vector, each dimension counts the number of such state in the segment, then normalize it by the total number of pixels in the segment. An experimental comparison between segmentation with HoG and segmentation with HoS is shown in Figure 3. It is clear that with HoG used in encoding the similarity, the visually coherent pyramid together with the desert are broken. In contrast, HoS leads to better visual performance, owing to the robustness of HoS to some small noise (because HoS thresholds the gradient magnitude with small values and uses a 8-dimensional binary vector to indicate the existence of the corresponding orientation), and HoS’s capability of better capturing the texture information (a sliding overlapping window, rather than a dense grid (for HoG), is used in HoS).

Remarks: HoS can be treated as a special case of the general Bag-of-Words model. For the general Bag-of-Words model, one important step is to construct the ‘Words’. In our case, each of the 256 states is used as a ‘Word’. For each region, we calculate the occurring frequency of each ‘Word’, and form a normalized histogram to represent the texture for this region.

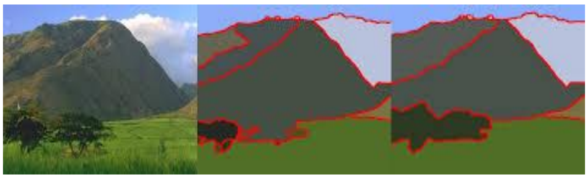


Fig . 2. Segmentation results based on different similarity matrices. Top: Original images. Middle: Segmentation results with consistent similarity matrix. Bottom: Segmentation results with adaptive similarity matrix.

IV. EXPERIMENTAL EVALUATION

In this section, extensive experiments have been done to evaluate the segmentation performance of our proposed algorithm, qualitatively as well as quantitatively. Moreover, we also discuss the time complexity of several different algorithms. The algorithms are tested on two datasets: 1) one dataset is the publicly available Berkeley Segmentation Data Set 500 (BSDS500) [7]. BSDS500 is comprised of 500 images, including 200 images for training, 200 images for testing and 100 images for validation. BSDS500 also provides ground-truth segmentations manually generated by several human subjects. For each image, 5 to 8 ground-truth segmentation maps are provided; 2) the other dataset is the Semantic Segmentation Data Set (SSDS) [28], which includes 100 images selected from BSDS500, and also contains the semantic level ground-truths that are generated by using the existing ground-truths of BSDS500 as well as an interactive segmentation tool. Figure 6 shows some sample images from BSDS500 and the corresponding ground-truth segmentations provided by BSDS500 and SSDS. It can be seen that some ground-truth segmentations provided by BSDS500 is of fine granularity, while SSDS gives better semantic level ground truth segmentations instead.

A. Time Complexity

Considering the large amount of pixels to deal with for images, lower time complexity without impacting the performance much is always preferred, especially in the situation where real time application is needed. We compare the run time of our proposed algorithm with CTM and TBES, since these three algorithms start with superpixels. Given the same superpixel initializations. we run the algorithms over the 100 validation images from BSDS500 once, and then compute the mean time and the lower/upper bound of the 95% confidence interval for the run time

V. CONCLUSION

In this paper, we have proposed an efficient image segmentation algorithm taking advantages of the scalability of modularity optimization and the inherent properties of images. Adopting the bottom-up framework, the proposed algorithm automatically detects the number of segments in the image, and by employing the color feature as well as the proposed Histogram of States (HoS) texture feature, it adaptively constructs the similarity matrix among different regions, optimizes the modularity and aggregates the neighboring regions iteratively. The optimal segmentation is achieved when no modularity increase occurs by aggregating any neighboring regions. Results of extensive experiments have validated that the proposed algorithm gives impressive qualitative segmentation results; besides, it is reported that the new algorithm achieves the best performance among all the experimented popular methods in terms of VOI and Precision on BSDS500. Since the algorithm aims to avoid over-segmentation, it produces low Recall value. In addition, it is demonstrated that the new algorithm can preserve regularities in the object and achieve the best performance from the semantic level on SSDS. What's more, our proposed algorithm provides appealing time complexity and runs consistently faster than CTM and TBES under the same experiment settings.

REFERENCES

[1] B. Bhanu and J. Peng, "Adaptive integrated image segmentation and object recognition," IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 30, no. 4, pp. 427–441, 2000.

[2] J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888–905, 2000.

[3] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 7, pp. 1271–1283, 2010.

[4] P. Arbel´aez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: An empirical evaluation," in IEEE Conference on Computer Vision and Pattern Recognition, CVPR. IEEE, 2009, pp. 2294–2301.

[5] L. Grady and E. L. Schwartz, "Isoperimetric graph partitioning for image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 3, pp. 469–475, 2006.

[6] J. Wang, Y. Jia, X.-S. Hua, C. Zhang, and L. Quan, "Normalized tree partitioning for image segmentation," in IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE, 2008, pp. 1–8.

[7] C. Couprie, L. Grady, L. Najman, and H. Talbot, "Power watershed: A unifying graph-based optimization framework," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 7, pp. 1384– 1399, 2011.

[8] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 5, pp. 898–916, 2011.

[9] V. Grau, A. Mewes, M. Alcaniz, R. Kikinis, and S. K. Warfield, "Improved watershed transform for medical image segmentation using prior information," IEEE Transactions on Medical Imaging, vol. 23, no. 4, pp. 447–458, 2004.

[10] B. Li and S. T. Acton, "Active contour external force using vector field convolution for image segmentation," IEEE Transactions on Image Processing, vol. 16, no. 8, pp. 2096–2106, 2007.

[11] J. Puzicha, J. Buhmann, Y. Rubner, and C. Tomasi, "Empirical evaluation of dissimilarity measures for color and texture," in The Proceedings of the 7th IEEE International Conference on Computer Vision, ICCV., vol. 2. IEEE, 1999, pp. 1165–1172.

[12] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 5, pp. 530–549, 2004.

