OPEN ACCESS

International Journal Of Modern Engineering Research (IJMER)

An Efficient Algorithm based on Modularity Optimization

S. Santhi Kumari[1], K. Jogi Naidu[2], K. Avinash Kumar[3]

[1]*Dept. of ECE, DADI Institute of Engg. & Tech., Anakapalle, Visakhapatnam, AP, India.*
[2]*Assistant Professor, Dept. of ECE, DADI Institute of Engg. & Tech., Anakapalle, Visakhapatnam, AP, India.*
[3]*Associate Professor, Dept. of ECE, AVANTHI Institute of Engg. & Tech., Cherukupally, Vizianagaram, AP, India.*

ABSTRACT:- *Digital images are much more pervasive than they were a number of years ago, causing in the critical need for the exploration of images, such as object recognition, image searching, indexing, and categorization. For some applications, such as image recognition or compression, we cannot process the whole image directly for the reason that it is inefficient and unpractical. As a very vital step for these high level image exploration tasks, image segmentation is a pre-processing process to group image pixels into some sizable homogeneous regions so that the complexity of further analysis can be substantially reduced. Therefore, several image segmentation algorithms were proposed to segment an image before recognition or compression. This paper outlines an awareness to hinge on the tedious arrangements in a homogeneous area based on the histogram of positions of image gradients and link of two regions can be done with the color feature. Over-segmentation problem can be meritoriously eluded by creating the comparison matrix.*

Keywords:- *Segmentation; exploration; recognition; pixels.*

I. INTRODUCTION

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subcategory or field of digital signal processing, digital image processing has many advantages over analog image processing. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems.

In imaging science, image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it.

Image segmentation has received considerable attention since the problem was proposed, *e.g.*, [3]–[10], yet it still remains to be a challenging problem due to the following reasons: 1) image segmentation is an ill-defined problem and the optimal segmentation is user or application dependent; 2) image segmentation is time consuming in that each image includes a large number of pixels, especially for high resolution images, and this prevents image segmentation from being applied to real-time applications. In fact, Gestalt principles [1] and some cognition and psychological studies [2] have pointed out that several key factors affect perceptual grouping a lot, for example, proximity, similarity, regularity, *i.e.*, the repetitive patterns, relative size and etc. In this paper, we will take all these factors into consideration, and develop a computational efficient algorithm. Moreover, comprehensive evaluations of the segmentation performance under various metrics are also presented.

II. PREVIOUS TECHNIQUES

A. Mean Shift Algorithm

It treats image segmentation as a problem of clustering by detecting the modes of the probability density function in the feature space. Each pixel in the image is transformed to the joint spatial-range feature space by concatenating the pixel color value and its spatial coordinates into a single vector. Then the mean shift procedure is applied in this feature space to yield a convergence point for each pixel. All the pixels whose convergence points are closer than the spatial bandwidth h_s and the range bandwidth h_r are claimed to be in the same segment. In addition, minimum segment size is enforced to guarantee sizable segmentation. This method is usually fast. However, it is very sensitive to the bandwidth parameter h_r and h_s, and often results in over-segmentation. Since it is based on the density estimation of the color feature for all the pixels, some smooth changes in brightness and texture or the regularities of different colors will converge to different modes, though they belong to the same segment visually.

B. Multiscale Normalized Cut Approach

It allows to deal with larger images by compressing large images into multiple scales. However, it has the same problem with *Normalized Cut*, specifically, it 1) often breaks uniform or smooth regions where the eigenvectors have smooth gradients; 2) has a high time complexity; 3) needs a predefined number of segments, which itself is a challenging problem to deal with. More graph based segmentation can be found in the literature, *e.g.*, [12]–[14]. However, they all need human intervention, *e.g.*, they need a user to specify the number of regions resulting from image segmentation.

C. Watershed Segmntation

This method regards the gradient magnitude of an image as a topographic surface. The pixels where a water drop starts from would drain to the same local intensity minimum are in one segment, which is called *catchment basins*. A watershed is formed by 'flooding' an image from its local minima, and forming 'dams' where waterfronts meet. When the image is fully flooded, all dams together form the watershed of an image. The watershed of an edgeness image (or, in fact, the watershed of the original image) can be used for segmentation. The idea is that when visualizing the edgeness image as a three-dimensional landscape, the catchment basins of the watershed correspond to objects, i.e., the watershed of the edgeness image will show the object boundaries. As the landscape in the figure shows, the object boundaries mark the catchment basins, but there are small defects because of image artifacts. Because of the way the watershed is constructed –it forms 'dams' where waterfronts meet– these small defects do not disturb the watershed segmentation much.

D. Compression-based Texture Merging

This method [11] fits the image textures using the Gaussian Mixture Model, and employs the principle of Minimum Description Length to find the optimal segmentation, which gives the minimum coding length under a certain distortion ratio. Later, the Texture and Boundary Encoding-based Segmentation algorithm [8] improves the CTM by considering a hierarchy of multiple window sizes and offering more precise coding length computation. To be specific, it only encodes the texture information inside the non-overlapping windows in each region and also encodes the boundary with the adaptive chain code. However, this greatly increases the computational time cost; besides, the texture feature used in these two algorithms is essentially the pure color information from the cut-off windows and neglects the regularities inside the image, thus it may split the object with some regularities of different color.

Another broad class of methods address the image segmentation problem via solving the *Partial Differential Equations* (PDEs). Most of PDE based methods are carried out by the active contour model or snakes [15], where the basic idea is to evolve a curve (object boundary) such that an energy function is minimized. The energy function usually contains the internal energy as well as the external energy. The internal energy controls the smoothness of the curve, whereas the external energy guides the curve toward the true object boundary. Moreover, lots of researches have made improvements over this model, however, generally, these methods are very sensitive to the noise, the model parameters and suffer from high computational cost.

III. IMAGE SEGMENTATION

Image segmentation is to classify or cluster an image into several parts (regions) according to the feature of image, for example, the pixel value or the frequency response. Up to now, lots of image segmentation algorithms exist and be extensively applied in science and daily life. A great variety of segmentation methods has been proposed in the past decades, and some categorization is necessary to present the methods properly here. A disjunct categorization does not seem to be possible though, because even two very different segmentation approaches may share properties that defy singular categorization[1].

The following categories are used:
• Threshold based segmentation: Histogram thresholding and slicing techniques are used to segment the image. They may be applied directly to an image, but can also be combined with pre- and post-processing techniques.
• Edge based segmentation: With this technique, detected edges in an image are assumed to represent object boundaries, and used to identify these objects.
• Region based segmentation: Where an edge based technique may attempt to find the object boundaries and then locate the object itself by filling them in, a region based technique takes the opposite approach, by (e.g.) starting in the middle of an object and then "growing" outward until it meets the object boundaries.

Perfect image segmentation –i.e., each pixel is assigned to the correct object segment– is a goal that cannot usually be achieved. Indeed, because of the way a digital image is acquired, this may be impossible, since a pixel may straddle the “real” boundary of objects such that it partially belongs to two (or even more) objects. Most methods presented here –indeed most current segmentation methods– only attempt to assign a pixel to a single segment, which is an approach that is more than adequate for most applications. Methods that assign a segment probability distribution to each pixel are called probabilistic. This class of methods is theoretically more accurate, and applications where a probabilistic approach is the only approach accurate enough for specific object measurements can easily be named. However, probabilistic techniques add considerable complexity to segmentation –both in the sense of concept and implementation–and as such are still little used.

Perfect image segmentation is also often not reached because of the occurrence of over-segmentation or under-segmentation. In the first case, pixels belonging to the same object are classified as belonging to different segments. A single object may be represented by two or more segments. In the latter case, the opposite happens: pixels belonging to different objects are classified as belonging to the same object.

IV. OUR APPROACH

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 priori 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 detailed presentation of some technical points for our algorithm is as follows:

A. Superpixels

Superpixels are a set of very small and homogeneous regions of pixels. Initializing with superpixels 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 superpixels. 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 superpixels generating algorithms. In our implementation, we use a publicly available code [16] to get the superpixel initialization.

B. Choice of Color Space

To capture different aspects of the color, various color spaces are proposed in the literature [17], 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.

E. Similarity Measure

We use different similarity measures for the two features. For the color feature, each pixel is represented by a three dimensional vector in the *L*a*b* color space. To measure the similarity between two regions of pixels, we assume that the pixel value in the same region follows a three dimensional Gaussian distribution, for example, pixels in region R_i and region R_j follow two Gaussian distributions, respectively, *i.e.*, $R_i \sim N(\mu_1,\Sigma_1)$ and $R_j \sim N(\mu_2,\Sigma_2)$.

F. Adaptive Similarity Matrix Construction

In the traditional *Louvain method* for *community detection*, a consistent similarity matrix is used to update the new similarity matrix by simply summing over the weights of nodes in two different communities during each iteration. However, in our algorithm, we propose to construct the similarity matrix adaptively. We maintain an adaptive similarity matrix during each iteration by re-computing the similarity between regions. The reason for this is because during the aggregation process, the region keeps expanding, and the similarity measure computed from the previous iteration might not suitable for current iteration. By maintaining an adaptive similarity matrix, we reevaluate the similarity between current regions and hence can effectively overcome the problem of splitting the non-uniformly distributed color or texture, which should be grouped into the same segment from the perspective of human vision system. In this way, over-segmentation is avoided.

V. RESULTS

The proposed algorithm produces sizeable segments for the original image shown in Fig.1. Even if some pixels have very different values inside the same segment, the similarity matrix encoding of the HoS texture feature successfully preserves the regularities and classifies those pixels into the same segment. Superpixels of the original image can be observed in Figure 2 and the overall segmented image can be seen in Figure 3.

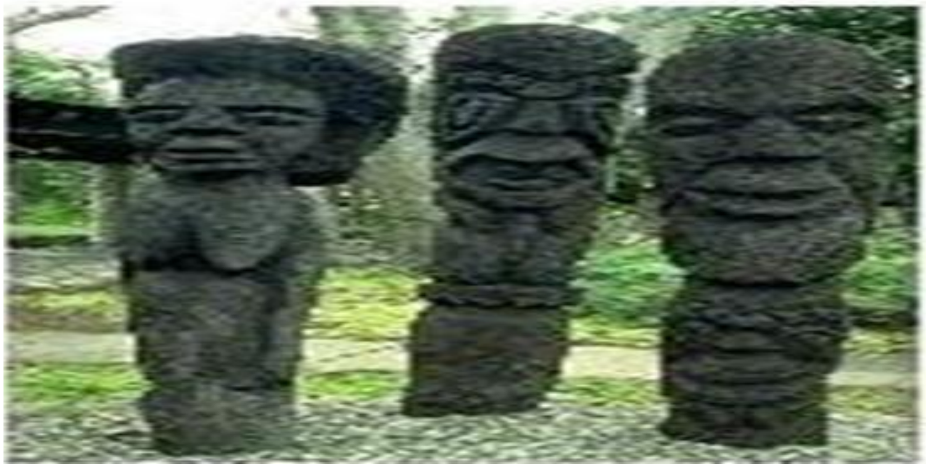

Fig.1: Original Image

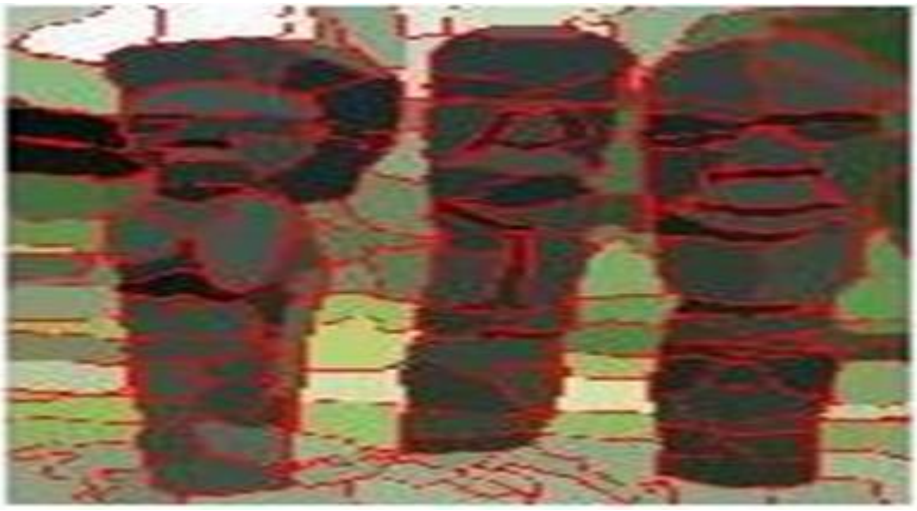

Fig. 2: Super pixels of an image

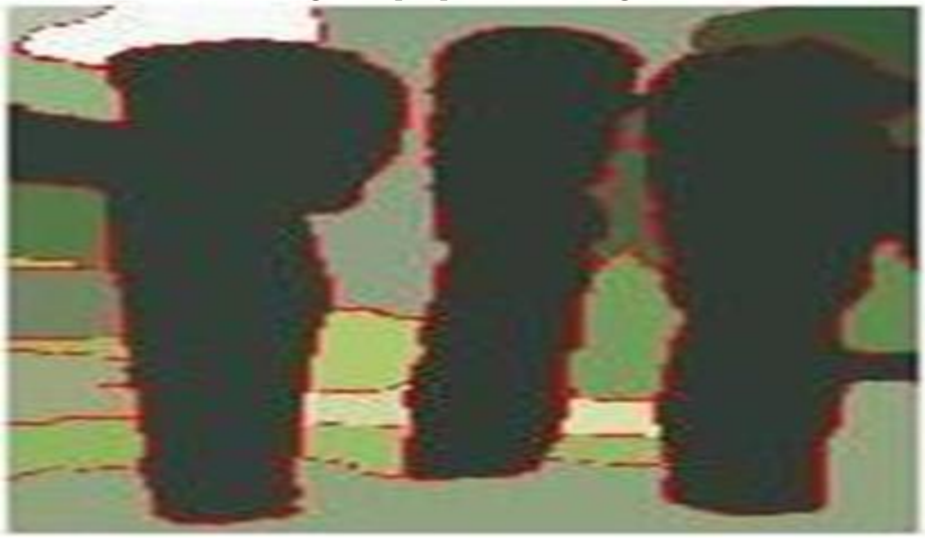

Fig. 3: Segmented image

The *Probabilistic Rand Index (PRI)* is a classical evaluation criteria for clusterings. *PRI* measures the probability that the pair of samples have consistent labels in the two segmentations. The range of *PRI* is [0, 1], with larger value indicating greater similarity between two segmentations.

The *Variation of Information (VOI)* measures how much we can know of one segmentation given another segmentation. VOI is defined by:

$$VOI(C, C^0) = H(C) + H(C^0) - 2I(C,C^0), \qquad (1)$$

Where, $H(C)$ and $H(C^0)$ are the entropy of segmentation C and C^0, respectively and $I(C,C^0)$ is the mutual information of two segmentations C and C^0. The range of this metric is $[0, +\infty)$, and the smaller the value is, the more similar the two segmentations are. For each algorithm, we use the same parameter settings to run across all the images. The mean values of *PRI* and *VOI* are reported in Table I.

TABLE I: Comparison of Different Algorithms

Algorithms	**PRI (larger better)**	**VOI (smaller better)**
Human	0.87	1.16
Our's (*L*a*b*)	0.767	1.789
Our's (*RGB*)	0.746	2.149
MS	0.772	2.004
MNC	0.742	2.651
MCW	0.753	2.203
CTM	0.735	1.978

In Table I, the third and fourth rows show the performance of the proposed algorithm in different color spaces. Again, it has been validated that the algorithm works better in the *L*a*b* color space than in the *RGB* color space in terms of both *PRI* and *VOI*. Therefore, we run our algorithm in the *L*a*b* color space for all the following experiments. Also, in terms of *VOI*, the proposed algorithm achieves the best performance among all the popular segmentation algorithms; in terms of *PRI*, our algorithm has outperforms the existing algorithms.

VI. CONCLUSION

When the modularity of the segmented image is maximized, the algorithm stops merging and produces the final segmented image. In this paper, we have proposed an effective image segmentation algorithm taking doles of the scalability of modularity optimization and the inherent properties of images. To preserve the repetitive patterns in a homogeneous region, we propose a feature based on the histogram of states of image gradients, and use it together with the color feature to characterize the similarity of two regions. The proposed algorithm spontaneously 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 sections, optimizes the modularity and groups the neighboring sections iteratively. The optimal segmentation is achieved when no modularity increase ensues by grouping any neighboring regions. This algorithm attains the best performance among all the trialed common techniques in terms of *VOI* and *PRI*.

ACKNOWLEDGMENT

We would like to acknowledge Hon'ble Correspondent of DIET, Sri Dadi Ratnakar garu for his immense support and Principal Dr. CH. Srinivasu for his suggestions. Also, we glad to acknowledge HOD ECE Sri A. Vamsidhar who gave the extreme support and all the faculty members of ECE and friends who supported for this work directly or indirectly.

REFERENCES

[1]. M. Wertheimer, "Laws of organization in perceptual forms," A Source Book of Gestalt Psychology, pp. 71–88, 1938.

[2]. D. D. Hoffman and M. Singh, "Salience of visual parts," Cognition, vol. 63, no. 1, pp. 29–78, 1997.

[3]. 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.

[4]. 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.

[5]. 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.

[6]. P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," International Journal of Computer Vision, vol. 59, no. 2, pp. 167–181, 2004.

[7]. P. Arbelaez, 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.

[8]. S. Rao, H. Mobahi, A. Yang, S. Sastry, and Y. Ma, "Natural image segmentation with adaptive texture and boundary encoding," Asian Conference on Computer Vision, ACCV., pp. 135–146, 2010.

[9]. 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.

[10]. H. Zhu, J. Zheng, J. Cai, and N. M. Thalmann, "Object-level image segmentation using low level cues," IEEE Transactions on Image Processing, 2013.

[11]. A. Yang, J. Wright, Y. Ma, and S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," Computer Vision and Image Understanding, vol. 110, no. 2, pp. 212–225, 2008.

[12]. 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.

[13]. 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.

[14]. 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.

[15]. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," International Journal of Computer Vision, vol. 1, no. 4, pp. 321–331, 1988.

[16]. G. Mori, "Guiding model search using segmentation," in The Proceedings of the 10th IEEE International Conference on Computer Vision, ICCV., vol. 2. IEEE, 2005, pp. 1417–1423.

[17]. K. Plataniotis and A. Venetsanopoulos, Color image processing and applications. Springer, 2000.