AUTOMATIC COLOR IMAGE SEGMENTATION BY DYNAMIC REGION GROWTH AND MULTIMODAL MERGING OF COLOR AND TEXTURE INFORMATION

Luis García-Ugarriza[(1)], *Eli Saber* [(1)], *Vincent Amuso* [(1)], *Mark Shaw*[(2)] *and Ranjit Bhaskar*[(2)]

[1] Department of Electrical Engineering, Rochester Institute of Technology, Rochester, NY
[2] Color and Imaging Division, Hewlett Packard Company, Boise, ID
luisgarciau@gmail.com, {esseee, vjaeee}@rit.edu, mark.q.shaw@hp.com, ranjit.bhaskar@hp.com

ABSTRACT

Image segmentation is a fundamental task in many computer vision applications. In this paper, we present a novel unsupervised color image segmentation algorithm that utilizes color gradients, dynamic thresholding and texture modeling algorithms in a split and merge framework. To this effect, pixels without edges are clustered and labeled individually to identify the preliminary image content. Pixels that contain higher gradients are further classified by utilizing an iterative dynamic threshold generation technique and an appropriate entropy based texture model. The proposed algorithm was demonstrated successfully on an extensive database of images and benchmarked favorably against prior art.

***Index Terms*—** Image Segmentation, Texture Segmentation, Color Segmentation, Region Merging

1. INTRODUCTION

An essential step in object based content analysis is to perform semantically meaningful spatial and temporal object segmentation. This is required to localize and track desired objects in an image or video source, perform adaptive rendering and/or scene classification. Although identification of object boundaries comes naturally to a human observer, automatically driven useful and accurate computer based image segmentation has proven to be a difficult task. Applications that would benefit from semantically meaningful object segmentations are widespread including visual database indexing and retrieval, medical imaging/analysis and compression. Extraction/segmentation of relevant content sets the stage for the automatic classification of objects on an assembly line, detection of faces in complex images, object-based multimedia editing, and so on.

There has been a significant amount of work published on gray & color image segmentation (see [1] for a survey). Pappas [2] proposed a spatial varying Gibbs random field (GRF) based model for gray scale image segmentation. Chang et al. [3] extended the above to accommodate multi-channel images by assuming conditional independence among the color channels. Saber et al. [4] introduced a GRF framework yielding improved segmentations and linked edge maps by fusing edge field and region based content. D'Elia et al. [5] proposed a tree structured binary Markov random field (MRF) model to recursively segment the image into progressively smaller regions. Alternatively, an initial segmentation may be obtained by over-segmenting the image and then merging the resulting regions as described in [6] or by utilizing semantic labels in a watershed framework [7]. Deng et al. [8] introduced the JSEG algorithm, which provided effective image segmentations in the presence of texture without utilizing prior models. Chen et al. [9] combined knowledge of human perception with an understanding of signal characteristics in order to segment natural scenes into perceptually uniform regions. Schwartz et al. [10] presented a new image segmentation method that utilizes texture features extracted by wavelet transforms combined with spatial dependence modeled by a MRF. However, even with the extensive research in this area, the current algorithms contain significant drawbacks that limit their effectiveness in real life/real time scenarios.

In this paper, we propose a novel unsupervised automatic color segmentation algorithm (see Fig. 1 for a block diagram) that utilizes color gradients, dynamic thresholding, and texture modeling in a split and merge framework. An initial segmentation is first obtained by dynamically thresholding the color gradient. This is followed by a color and entropy based texture detection algorithm that provides additional information about the initial segmented regions. The final segmentation is obtained using a multimodal-merging approach that recognizes objects displaying occlusion and complex patterns, yielding an improved semantically meaningful segmentation map. The remainder of this paper is organized as follows. The Region Growth and Dynamic Seed Generation is detailed in Section 2. Texture modeling and region merging are outlined in Section 3 and 4 respectively. Results are provided in Section 5 and conclusions drawn in Section 6.

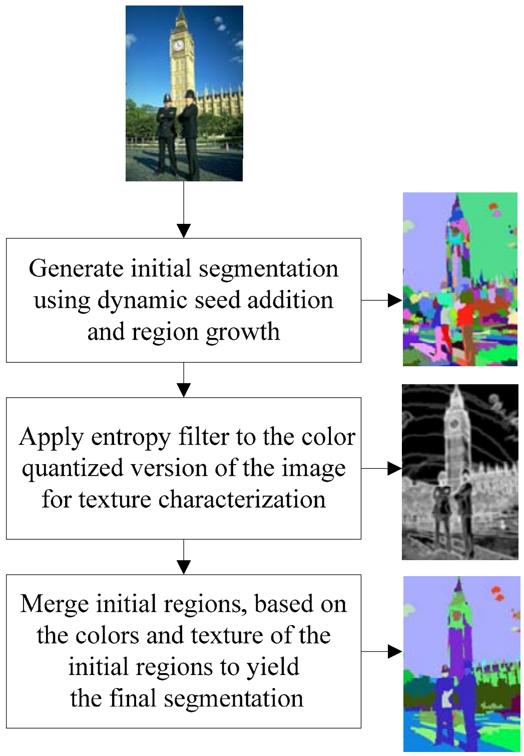


Figure 1: Flowchart of Segmentation Algorithm

2. REGION GROWTH AND DYNAMIC SEED GENERATION

The quality of current region growing techniques is, in general, highly dependent on the locations chosen to initialize the growing procedure. We propose an alternative process for region growth that does not depend exclusively on the initial assignment of clusters in order to yield a stable and robust final segmentation. The procedure begins by searching for regions in the image, where the gradient is below a certain threshold. The color gradient is computed as described in detail in [11]. These regions form the initial set of clusters used to seed the segmentation process. Subsequent pixels/clusters are incorporated dynamically by utilizing their corresponding gradient as well as an entropy based texture model in a fusion framework. A block diagram of the proposed algorithm is illustrated in Figure 1 followed by a detailed description in the corresponding sub-sections.

2.1 Initial Seed Generation

The gradient map, computed as described in [11], is utilized to generate the parent seeds (PS) and initialize the segmentation process. This is done by first analyzing the color gradient dynamic range in order to select a suitable threshold for identifying "flat" regions. This threshold is then utilized to provide an initial classification of all pixels found in the image. Pixels that fall below the threshold are classified as non-edge pixels and utilized to generate initial clusters.

A problem arises at this stage due to the fact that real life images are not always easily classified into edge and non edge type pixels. A threshold that may correctly delineate the boundary of a given region may allow other regions to be merged. We combat this inherent problem by initially selecting regions that do not contain any edges within. If such regions are not found, the threshold value is increased until at least one region is detected to form the initial parent seeds (PS) map. To prevent multiple seed generation within homogeneous and connected regions, the region selection at this stage is restricted to clusters of pixels which are larger than 0.5% of the image. Each individual cluster is assigned a particular label for differentiating purposes.

2.2 Region Growth

Once the initial parent seeds have been clearly identified and labeled, we proceed to expand the clusters formed in Section 2.1 by incorporating new unlabeled pixels. To this effect, the region growth procedure is initiated by increasing the threshold found in the initial seed generation in order to detect new regions or child seeds that fall below the new threshold. These child seeds are classified into one of two categories: the first consists of seeds that are adjacent to the currently classified ones, while the second encompasses newly available seeds that are not adjacent to current seeds and may lead to new potential regions. In order to make the region growth process as efficient as possible, it is imperative to keep track of the parent seeds that are adjacent to the child ones. The objective is to be able to process all adjacent child seeds in a computationally efficient manner. We implement this by first detecting the outside edges of the PS map, using a nonlinear spatial filter. The output of the filter F(i, j) is computed as follows:

$$\boldsymbol{F(i,j)} = \begin{cases} \mathbf{0} & \boldsymbol{if\ PS(i,j)} > 0 \\ \mathbf{0} & \boldsymbol{if\ \sum_{(m,n)\in\beta} PS(m,n) = 0} \\ \mathbf{1} & \boldsymbol{otherwise} \end{cases} \qquad \mathbf{(1)}$$

where β represents the 3 x 3 neighborhood that is being operated on. The result of applying this filter is a mask indicating the borders of the PS map.

The child seeds are individually labeled. Those that are adjacent to the parent seeds are identified by performing an element-by-element multiplication of the parent seed edge mask and the labeled child map. Since the edges of parent seeds are composed of ones, the multiplication will determine the labels of the child seeds that are located adjacent to them. In order for the adjacent child seeds to be incorporated into the region, it is necessary to compare their individual color differences to their parents. Reduction of the number of seeds to be evaluated is accomplished by combining parent and child seeds that have a size smaller than the minimum seed size (MSS). In our algorithm the MSS is set to 0.01% of the image.

To combine regions efficiently, an association between parent and child seeds is required. A non-linear spatial filter is applied to the PS map to obtain the parent labels. The filter's response at each center point is equal to the maximum pixel value in its neighborhood. The association between child and parent is obtained by creating a two column matrix, where the first column contains the adjacent child pixels and the second column contains the output of the non-linear filter mentioned above.

The association matrix serves two purposes. It provides the number of child pixels that are attached to each parent seed, and identifies the child seeds that share edges with more than one parent. Child seeds smaller than MSS can now be directly attached to the corresponding parent. On the other hand, those that share less than 5 pixels with their parents and are larger than the MSS are not labeled at this time are re-considered for classification at subsequent iterations and so on until all child seeds have been labeled.

2.3 Dynamic Seed Generation

The dynamic addition of seeds to the PS map is designed to provide a segmentation label for the remaining pixels/regions that display different levels of edge intensities and have not yet been associated with any parent. This is accomplished by selecting a set of threshold values, so that additional parent seeds may be formed. The threshold values are adjusted to account for the exponential decay in the number of edge values. Low edge values correlate to relatively large areas in the image. For the dynamic seed generation process to incorporate new areas, the threshold values need to increase exponentially in order to include elements of considerable size into the segmentation map. The threshold values selected for the addition of new seeds is advanced over a series of discrete gradient levels. For the work presented in this paper, we have chosen the following set of threshold values {15, 20, 30, 50, 85, and 120}, which takes into account an increment of ±10% of the area of the image added at each interval.

At these intervals, the addition of new parent seeds follows a similar procedure to the region growth method. The regions that fall below the selected edge threshold are detected. All the regions that are not attached to any parent seeds and are larger than the MSS are added to the PS map. For the addition of new seeds that share borders with existent seeds, two qualifications need to be met: 1) the group must be large enough to merit its own label, and 2) the color difference between the region and its neighbors is greater than the maximum allowed threshold.

3. TEXTURE CHANNEL GENERATION

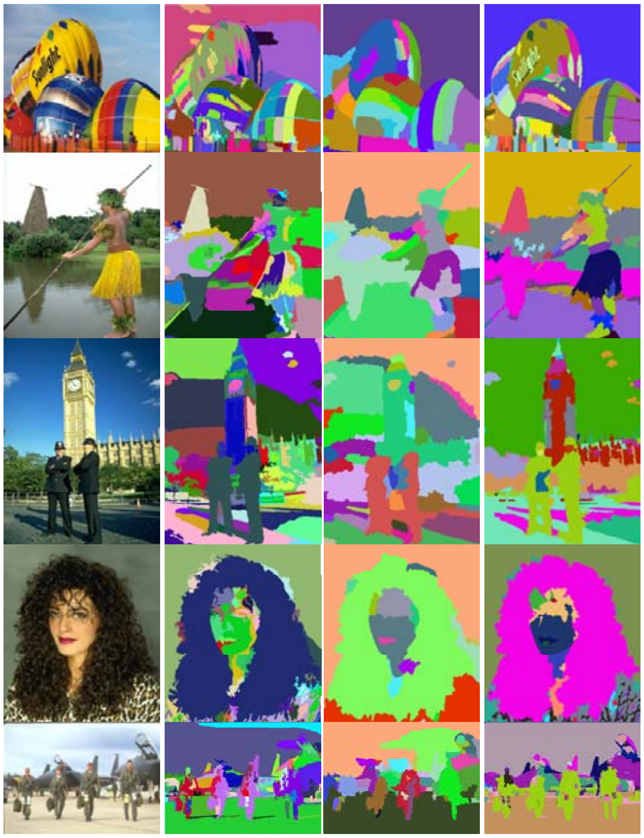
Texture information can be used to aid in the segmentation process. Regions of pixels within an image that exhibit a large variance in colors and/or shading indicate potential differences in texture. Entropy provides a measure of uncertainty in these regions. By calculating the entropy of image segments, regions with similar entropy values may be grouped together in the merging process.

To take advantage of the color information while minimizing the level of computational complexity and maximizing the classification accuracy, the colors in the image are uniformly quantized using six levels per channel. Each sub-cube is then given a specific color label yielding a mapping from a multi-dimensional color image (i.e. RGB image) to a single color label channel. Once the colors have been quantized and labeled as described above, each pixel in the image is now indexed to one of the representative color labels. The texture channel is hereby created by computing the local entropy on a 9-by-9 neighborhood around each pixel in the color labeled image, and assigning the resulting value to the center pixel of the neighborhood. This is utilized to aid in the merging process described in Section 4 below.

4. MULTIMODAL REGION MERGING

The image color information, the current labeled segmentation map derived from Section 2, and the texture channel computed as described above are utilized to merge neighboring pixels/regions in order to create the final segmentation map. Using a multivariate analysis of the independent regions [12], the resultant Mahalanobis distances between groups is used to merge similar regions. Given that we have multiple sources of information (colors and texture), as well as individual regions with a different number of pixels per region, we require a suitable method to arrange the data in order to efficiently investigate inter-relationships between the regions. Hence, the data is modeled using an N x P matrix, where N is the total number of pixels in the image, and P is the total number of variables that contain information about each pixel. Let G be the total number of regions in which the image has been segmented insofar, then the N x P matrix is composed of G separate individual regions.

The merging procedure is then accomplished as follows. The global mean value for each region is first computed. Applying one-way analysis of variance [12] to the N x P matrix, the Mahalanobis squared distances for each pair of regions is then calculated. Our algorithm uses these distances to find and merge similar regions based on the minimum Mahalanobis distance. Once two regions have been merged, the distances and segmentation labels are updated accordingly and the above process is repeated until



(a) (b) (c) (d)

Figure 2: a) Original Image, b) GRF segmentation, c) JSEG, d) Proposed Algorithm

the minimum Mahalanobis distance exceed a user specified threshold. This yields the final segmentation map for the image at hand.

5. RESULTS

Our proposed algorithm was tested on a large database (~4,000 images) acquired from the University of California at Berkeley as well as many others. A sample set is illustrated in Figure 2. The original image is shown in Figure 2a, its corresponding GRF and JSEG segmentation using the algorithms documented in [4] and [8] are displayed in Figures 2b and 2c respectively, and our proposed segmentation is illustrated in Figure 2d. The parameter utilized for our segmentations were selected empirically for optimum qualitative results and held constant for all images in the database. The implementation was done in a MATLAB environment (version 7.4 on an Intel P-4 dual core 3.2 GHz) and is capable of segmenting a real life 512 x 512 image in approximately 45 seconds.

The advantages of our proposed algorithm are illustrated in the resulting segmentations found in Fig. 2d. Note the clarity of the "Sunlight" text in the top image, the definition of the segmentation as well as the reduction in "sky" classes in "Big Ben", and the clear separations of neighboring classes in the "Hawaii", "airplane" and "girl" scenes in comparison to prior art. In general, our algorithm is less susceptible to over/under segmentation and provides clearer distinctions between neighboring regions.

6. CONCLUSIONS

In this paper, we presented a novel automatic image segmentation algorithm based on color edge detection and dynamic region growing/merging. Our proposed algorithm was benchmarked against existing work yielding superior qualitative performance. The underlying segmentations serve to provide a firm foundation for object/content based image and video classification algorithms.

7. ACKNOWLEDGMENTS

The authors wish to thank Hewlett-Packard Corporation for their generous support of this research project.

8. REFERENCES

[1] L. Lucchese and S. Mitra, "Color image segmentation: A state of the art survey," *Proc. Indian National Science Acad.*, Vol. 67, No. 2, pp. 207 – 221, 2001.

[2] T. Pappas, "An adaptive clustering algorithm for image segmentation," *IEEE Trans. on Signal Process*, vol. 40, no. 4, pp. 901–914, 1992.

[3] M. Chang, M. Sezan, and A. Tekalp, "Adaptive Bayesian segmentation of color images," *Jour. of Elec. Imag.*, Vol. 3, no. 4, pp. 404–414, 1994.

[4] E. Saber, A. Tekalp, and G. Bozdagi, "Fusion of color and edge information for improved segmentation and edge linking," *Imag. and Vision Comp.*, Vol. 15, pp. 769–780, 1997.

[5] C. D'Elia, G. Poggi, and G. Scarpa, "A tree-structured Markov random field model for Bayesian image segmentation," *IEEE Trans. on Image Proc.*, Vol. 12, pp. 1259–1273, 2003.

[6] Q. Luo and T. Khoshgoftaar, "Unsupervised Multiscale Color Image Segmentation Based on MDL Principle," *IEEE Trans. on Image Proc.*, Vol. 15, pp. 2755-2761, 2006.

[7] T. Athanasiadis, P. Mylonas, Y. Avrithis and S. Kollias. "Semantic Image Segmentation and Object Labeling," *IEEE Trans. on Circ. and Sys. for Video Tech.*, Vol. 17, No. 3, pp. 298-312, 2007.

[8] Y. Deng and B. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. on Pat. Anal. Mach. Int.*, Vol. 23, pp. 800–810, 2001.

[9] J. Chen, T. Pappas, A. Mojsilovic and B. Rogowitz. "Adaptive Perceptual Color-Texture Image Segmentation," *IEEE Trans. on Image Proc.*, Vol. 14, No. 10, pp. 1524-1536, 2005.

[10] W. Schwartz and H. Pedrini, "Texture Image Segmentation Based on Spatial Dependence using a Markov Random Field Model," *IEEE Int. Conf. on Image Proc.*, pp. 2449-2452, 2006.

[11] H. Lee and D. Cok, "Detecting boundaries in a vector field," *IEEE Trans. on Signal Proc.*, Vol. 39, pp. 1181–1194, 1991.

[12] W. Krzanowski, *Principles of Multivariate Analysis*, chapter 11, Oxford University Press, 1988.