Processing of Images Based on Segmentation Models for Extracting Textured Component

V M Viswanatha, Nagaraj B Patil, Dr. Sanjay Pande M B

Abstract— The method for segmentation of color regions in images with textures in adjacent regions being different can be arranged in two steps namely color quantization and segmentation spatially. First, colors in the image are quantized to few representative classes that can be used to differentiate regions in the image. The image pixels are then replaced by labels assigned to each class of colors. This will form a class-map of the image. A mathematical criteria of aggregation and mean value is calculated. Applying the criterion to selected sized windows in the class-map results in the highlighted boundaries. Here high and low values correspond to possible boundaries and interiors of color texture regions. A region growing method is then used to segment the image.

Key Words- Texture segmentation, clustering, spital segmentation, slicing, texture composition, boundry value image, median-cut.

1 INTRODUCTION

Segmentation is the low-level operation concerned with partitioning images by determining disjoint and homogeneous regions or, equivalently, by finding edges or boundaries. Regions of an image segmentation should be uniform and homogeneous with respect to some characteristics such as gray tone or texture Region interiors should be simple and without many small holes. Adjacent regions of segmentation should have significantly different values with respect to the characteristic on which they are uniform. Boundaries of each segment should be simple, not ragged, and must be spatially accurate".

Thus, in a large number of applications in image processing and computer vision, segmentation plays a fundamental role as the first step before applying the higher-level operations such as recognition, semantic interpretation, and representation.

Earlier segmentation techniques were proposed mainly for gray-level images on which rather comprehensive survey can be found. The reason is that, although color information permits a more complete representation of images and a more reliable segmentation of them, processing color images requires computation times considerably larger than those needed for gray-level images. With an increasing speed and decreasing costs of computation; relatively inexpensive color camera the limitations are ruled out. Accordingly, there has been a remarkable growth of algorithms for segmentation of color images. Most of times, these are kind of "dimensional extensions" of techniques devised for gray-level images; thus exploit the well-established background laid down in that field. In other cases, they are ad hoc techniques tailored on the particular nature of color information and on the physics of the interaction of light with colored materials. More recently, Yining Deng and B. S. Manjunath [1][2] uses the basic idea of separate the segmentation process into color quantization and spatial segmentation. The quantization is performed in the color space without considering the spatial distributions of the colors. S Belongie,

- *V M Viswanatha is currently pursuing Ph.D from Singhania University, Rajastan. At present he is working as a Asst. Professor , E&CE Dept. at SLN College of Engg. Raichur E-mail: vmviswanatha@gmail.com*
- *Nagaraj B Patil is currently pursuing. Ph.Dfrom Singhania University Rajastan. At present he is working as Asst. Prof. and HOD Dept. of CSE, Govt. Engg. College Raichur. E-mail: nagarajbpatil@yahoo.co.in*
- *Dr. Sanjaya Pandey , At present is working as a Prof. and HOD Dept. of CSE at VVIT, Mysore. E-mail: rakroop99@gmail.com*

ISSN 2229-5518

et.al [3], in their paper present a new image representation, which provides a transformation from the raw pixel data to a small set of image regions that are coherent in color and texture space. A new method of color image segmentation is proposed in [4] based on K-means algorithm. Both the hue and the intensity components are fully utilized.

2 Color Quantization

Color Quantization is a form of image compression that reduces the number of colors used in an image while maintaining, as much as possible, the appearance of the original. The optimal goal in the color quantization process is to produce an image that cannot be distinguished from the original. This level of quantization in fact may never be achieved. Thus, a color quantization algorithm attempts to approximate the optimal solution.

The process of color quantization is often broken into four phases .

1) Sample image to determine color distribution.
2) Select colormap based on the distribution
3) Compute quantization mapping from 24-bit colors to representative colors
4) Redraw the image, quantizing each pixel.

Choosing the colormap is the most challenging task. Once this is done, computing the mapping table from colors to pixel values is straightforward.

In general, algorithms for color quantization can be broken into two categories: Uniform and Non-Uniform. In Uniform quantization the color space is broken into equal sized regions where the number of regions N_R is less than or equal to Colors K. Uniform quantization, though computationally much faster, leaves much room for improvement. In Non-Uniform quantization the manner in which the color space is divided is dependent on the distribution of colors in the image. By adapting a colormap to the color gamut of the original image, it is assured of using every color in the colormap, and thereby reproducing the original image more closely.

The most popular algorithm for color quantization, invented by Paul Heckbert in 1980, is the median cut algorithm. Many variations on this scheme are in use. Before this time, most color quantization was done using the popularity algorithm, which essentially constructs a histogram of equal-sized ranges and assigns colors to the ranges containing the most points. A more modern popular method is clustering using Octree,

3. Segmentation

The division of an image into meaningful structures, *image segmentation*, is often an essential step in image analysis. A great variety of segmentation methods have been proposed in the past decades. They can be categorized into

Threshold based segmentation: Histogram thresholding and slicing techniques are used to segment the image.

Edge based segmentation: Here, detected edges in an image are assumed to represent object boundaries, and used to identify these objects.

Region based segmentation: Here the process starts in the middle of an object and then grows outwards until it meets the object boundaries.

Clustering techniques: Clustering methods attempt to group together patterns that are similar in some sense.

Perfect image segmentation cannot usually be achieved because of *oversegmentation* or *undersegmentation*. In oversegmentation pixels belonging to the same object are classified as belonging to different segments.

In the latter case, pixels belonging to different objects are classified as belonging to the same object.

4. Design

Natural pictures are rich in color and texture. Texture segmentation algorithms require the estimation of texture model parameters which is difficult. The goal of this work is to segment images into homogeneous color-texture regions. The approach tests for the homogeneity of a given color-texture pattern, which is computationally more feasible than model parameter estimation. In order to identify homogeneity, the following assumptions are made:

- Each image contains homogeneous color-texture regions.
- The information in each image region can be represented by a set of few quantized colors.
- The colors between two neighboring regions are distinguishable

4.1 The Process of Segmentation

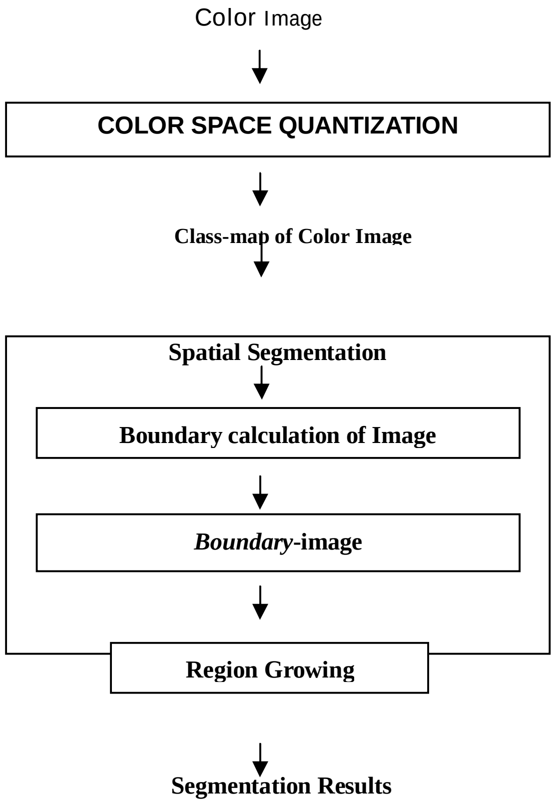
The segmentation is carried out in two stages. In the first stage, colors in the image are quantized to several representative classes to differentiate regions in the image. Then the image pixel values are replaced by their corresponding labels to form a image class-map that can be viewed as a special kind of texture composition. In the second stage, spatial segmentation is performed directly class-map. A few good existing quantization algorithms are used in this work.

The focus of this work is on spatial segmentation and can be summarized as:

• For image segmentation a parameter is calculated. This involves minimizing a cost associated with the partitioning of the image based on pixel labels.

• Segmentation is achieved using an algorithm. The notation of Boundary-images, correspond to measurements of local homogeneities at different scales, which can indicate potential boundary locations.

• A spatial segmentation algorithm that grows regions from seed areas of the Boundary-images to achieve the final segmentation. Figure. 4.1 shows a schematic of the algorithm for color image segmentation.



4.2 The Parameter Calculation as a criterion for Segmentation

It is difficult to handle a 24-bit color images with thousands of colors. Image are coarsely quantized without significantly degrading the color quality. Then, the quantized colors are assigned labels. A color class is the set of image pixels quantized to the same color. The image pixel colors are replaced by their corresponding color

ISSN 2229-5518

class labels and this image is called a class-map. Usually, each image region contains pixels from a small subset of the color classes and each class is distributed in a few image regions.

The class-map image is viewed as a special kind of texture composition. The value of each point in the class-map is the image pixel position, a 2-D vector (*x*, *y*). Each point belongs to a color class.

Let *Z* be the set of all *N* data points in a class-map. Let , z= (x,y), z Є Z , and *m* be the mean,

$$m = 1/N \sum_{z \in Z} z$$

Suppose *Z* is classified into *C* classes, Z_i, i =1,…C. Let m_i be the mean of the N_i data points of class Z_i,

$$m_i = 1/N_i \sum_{z \in Z_i} z$$

Let

$$S_T = \sum_{z \in Z} \| z-m \|^2$$

$$S_W = \sum_{i=1} S_i = \sum_{i=1} \sum_{z \in Z_i} \| z-m_i \|^2$$

S_W is the total variance of points belonging to the same class.

$$J = (S_T - S_W) / S_W$$

For the case of an image consisting of several homogeneous color regions, the color classes are more separated from each other and the value of *J* is large. An example of such a class-map is class-map 1 in Figure.4.2 and the corresponding *J* value is 1.7. Figure. 4.2 shows another example of a three class distribution for which *J* is 0.8, indicating a more homogeneous distribution than class-map 1. For class-map 2, a good segmentation would be two regions. One region contains class '1' and the other one contains classes '2' and '0'.

Now *J* is recalculated over each segmented region instead of the entire class-map and define the average J_{avg} by

$$J_{avg} = 1/N \sum M_k J_k$$

where *Jk* represents *J* calculated over region *k*, *Mk* is the number of points in region *k*, *N* is the total number of points in the class-map, and the summation is over all the regions in the class-map. J_{avg} is the parameter used for segmentation. For a fixed number of regions, a better segmentation tends to have a lower value of J_{avg}. If the segmentation is good, each segmented region contains a few uniformly distributed color class labels and the resulting *J* value for that region is small. Therefore, the overall J_{avg} is also small. Notice that the minimum value J_{avg} can take is 0. For the case of class-map 1, the partitioning shown is optimal(J_{avg}=0).

Figure.4.2 shows the manual segmentation results of class-maps.

Since the global minimization of J_{avg} for the entire image is not practical , *J*, if applied to a local area of the class-map, indicates whether that area is in the region interiors or near region boundaries

J=1.72

$J1 = 0$, $J2 = 0$, $J0 = 0$

$J_{avg} = 0$

1 1 1 1 1	0 0 0 0
1 1 1 1 1	0 0 0 0
1 1 1 1 1	0 0 0 0
1 1 1 1 1	0 0 0 0
1 1 1 1 1	0 0 0 0
1 1 1 1	2 2 2 2 2
1 1 1 1	2 2 2 2 2
1 1 1 1	2 2 2 2 2
1 1 1 1	2 2 2 2 2

J=0.855

$J1 = 0$, $J\{2,0o\} = 0.011$

$J_{avg} = 0.05$

1 1 1 1 1	2 0 2 0
1 1 1 1 1	0 2 0 2
1 1 1 1 1	2 0 2 0
1 1 1 1 1	0 2 0 2
1 1 1 1 1	2 0 2 0
1 1 1 1	2 0 2 0 2
1 1 1 1	0 2 0 2 0
1 1 1 1	2 0 2 0 2
1 1 1 1	0 2 0 2 0

ISSN 2229-5518

Figure 4.2: Segmented class-maps and their corresponding J values.

Now an image is constructed whose pixel values correspond to these *J* values calculated over small windows centered at the pixels. These are refered as *Boundary-values Images* and the corresponding pixel values as local *J* values. The higher the local *J* value is, the more likely that the corresponding pixel is near a region boundary. The *Boundary*-image contains intensities that actually represent the region interiors and region boundaries, respectively. Windows of small size are useful in localizing the intensity/color edges, while large windows are useful for detecting texture boundaries. Often, multiple scales are needed to segment an image. In this implementation, the basic window at the smallest scale is a 9 x 9 window. The smallest scale is denoted as scale 1. The window size is doubled each time to obtain the next larger scale.

The characteristics of the *Boundary*-images allow us to use a region-growing method to segment the image. Initially the entire image is considered as one region. The segmentation of the image starts at a coarse initial scale. It then repeats the same process on the newly segmented regions at the next finer scale. Region growing consists of determining the seed points and growing from those seed locations. Region growing is followed by a region merging operation to give the final segmented image.

5. Implementation

The implementation of segmentation is carried out using JDK 1.5. and JAI 1.3 (Java Advanced Imaging 1.3) on windows platform.

The module ColorQuantizer implements a GUI that takes the number of colors the image should be quantized into along with the algorithm to choose from between Median-Cut, NeuQuant, Oct-Tree.

The module Segmentor gets arguments regarding the initial window size and maximum number of iterations. Here the segmentation parameter is calculated and boundary image is created. It scans the color vector in boundary image and calculates initial segmented image. Then for the number of iterations mentioned it segments images repeatedly by converging the values. The accuracy of segmentation depends on the number of iterations the image is segmented.

The module for Region Growing implements a simple region growing algorithm. It runs a classic stack-based region rowing algorithm. It finds a pixel that is not labeled, labels it and stores its coordinates on a stack. While there are pixels on the stack, it gets a pixel from the stack (the pixel being considered), Checks its neighboring pixels to see if they are unlabeled and close to the considered pixel, if are, label them and store them on the stack Repeats this process until there are no more pixels on the image.

Figure 5.1: Snapshot of Selected Image and Segmented Image (NeuQuant,12)

ISSN 2229-5518

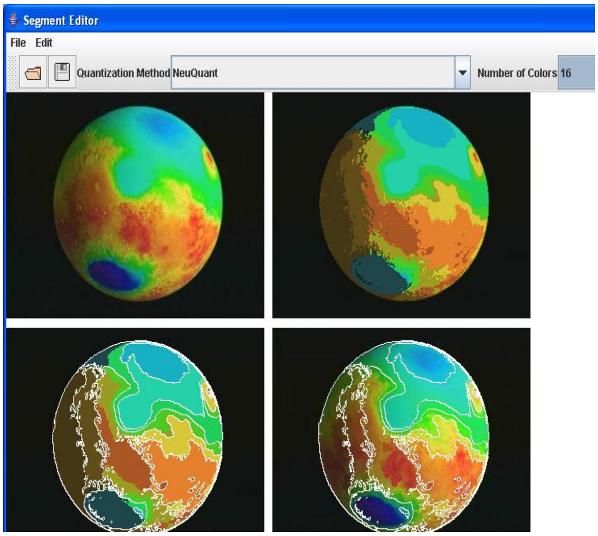
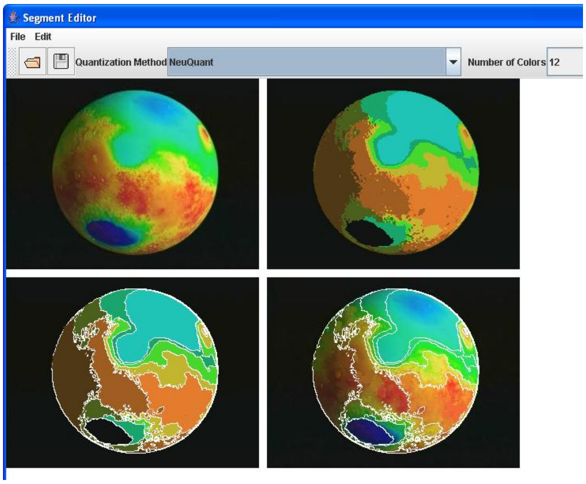


Figure 5.2: Snapshot of Selected Image and Segmented Image (NeuQuant,16)



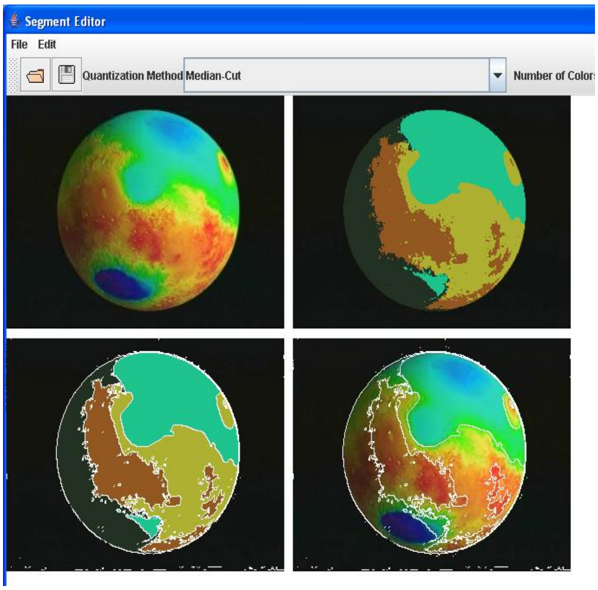
In Figures 5.1 thru 5.3 ,we can see the border displayed on the original image, border displayed on the segmented image, original image and segmented image , shown in top left to bottom right in that order

It is observed that changing of the parameters also changes the result. The number of colors for quantization chosen and the number of iterations changes the number of regions in the segmented image.

6. Conclusion

The color image is quantized and then is segmented with a focus on spatial segmentation. The threshold values are calculated that is used for the segmentation at various scales. At each step the segments are identified through the boundaries and valleys that are identified with the threshold values that are low near the boundary.The accuracy of segmentation depends on the algorithm used for quantization and the number of iterations.

Figure 5.3: Snapshot of Selected Image and Segmented Image (Mediun Cut,8)



Once the segmentation is over the image is displayed with appropriate borders to differentiate between the segments. The images having different colors in the adjacent regions are the best candidates for the segmentation using the proposed method.

However, it has does not handle pictures where smooth transition takes place between adjacent regions and there is no clear visual boundary. For instance, the color of a sunset sky can vary from red to orange to dark in a very smooth transition.

7. Future Work

The algorithm can be modified to find out the number of colors adaptively for quantization instead of taking it as a parameter.

The algorithm can be extended to handle images with smooth transitions between adjacent regions. The method can be extended to segment the color-texture regions in video data.

Acknowledgements

The authors like to thank Singhania Universaity, Rajasthan and VVIT Institute of Technology, Mysore.

Bibliography

[1] Y. Deng, B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," IEEE Transactions on Pattern Analysis and Machine Intelligence , Aug 2001.

[2] Y. Ma and B. S. Manjunath, "EdgeFlow: A technique for boundary detection and segmentation", IEEE Transactions on Vol. 9, pp. 1375-88, August 2000.

[3] S.Belongie, C. Carson, H.Greenspan and J.Malik,"Color and texture based image segmentation usingEM and its application to content based image retrieval", International Conference on Computer Vision, pp. 675-682, 1998.

[4] Pappas, T.N.," An adaptive clustering algorithm for image segmentation", International Conference on Pattern recognition, Vol 3, pp 3619, 2000.

[5] H.D. Cheng, X,H Jiang, Y Sun and Jingli Wang," Color Image Segmentation: Advances and Prospects", IEEE Transactions on Pattern Recognition, Vol 34, pp 2259- 2281, December 2001.

[6] D.K. Panjwani and G. Healey, "Markov random field models for unsupervised segmentation of textured color images," *IEEE Trans. on Pattern* Analysis and Machine Intelligence, vol. 17, no.10, p. 939-54, 1995.

[7] Heng-Da Cheng Ying Sun ," A hierarchical approach to color image segmentation using homogeneity", Vol 9, pp 2071- 2082.

[8] Jianbo Shi, Jitendra Malik,"Normalized cuts and image segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence ,Vol 22, pp 888-905, August 2000.

[9] O'Reilly - Safari Books Online - Java 2D Graphics.

[10] http://java.sun.com/products/java-media/jai/forDevelopers/

[11] Digital Image Processing, Rafael C. Gonzalez, Richard E.Woods, Pearson Education, Addison Wesley Longman

[12] Digital Image Processing, a practical introduction using Java, Nick Efford, Pearson Education