Color Image Classification

Marcel J. Castro
Advisors: Vidya Manian and Ramón Vásquez

Computing Research Conference
December 4, 1999

Laboratory for Applied Remote Sensing and Image Processing
Electrical and Computer Engineering Department
University of Puerto Rico, Mayagüez, PR 00681
mcastro@ece.uprm.edu

ABSTRACT

This paper describes the process of classifying color images based on color texture information. The images are originally in Red-Green-Blue (RGB) and they are changed to xyY to facilitate the image processing. Chromacity information (xy) is combined with luminance (Y) in the image. Luminance and chrominance image processing implementation is included in this paper. The process analyzes them separately to finally use them both together to classify the image. Luminance information is processed in three stages: filtering, smoothing, and boundary detection. Chrominance information on the other hand, is processed in one stage: histogram multi-thresholding. Classification based on luminance is done by Gabor filtering and then calculating the approximately features. Results are presented for 6 color images.

1. *Introduction*

The feature of primary importance in image processing and computer vision is the texture. It provides unique information about the physical characteristics of surfaces, scenes and objects. Most of the research in the area of texture analysis has focused on methods using gray levels in which only the intensity determines the texture. If we add the chromatic information to the luminance information to determine the different textures our results cover a wider range of parameters. The scope of this study is to implement this method and estimate its performance.

2. *Color Images*

We start with an RGB image, which is transformed to xyY format. This will produce two chromatic components and a luminance component. The luminance component is filtered and smoothed. After this step the image is applied to a similarity based neural network processing resulting in a boundary image. The two chromatic components are used to compute the chrominance histogram and identify multiple thresholds which are used to segment the image into regions. The resulting image of the chrominance processing is a region image. Then the two images (boundary and region) are combined to expand the regions of interest. At this moment we have a Region of Interest (ROI) Image, which is compared to other images at the same level of processing to locate possible scene changes. All this process is represented in the flowchart below.

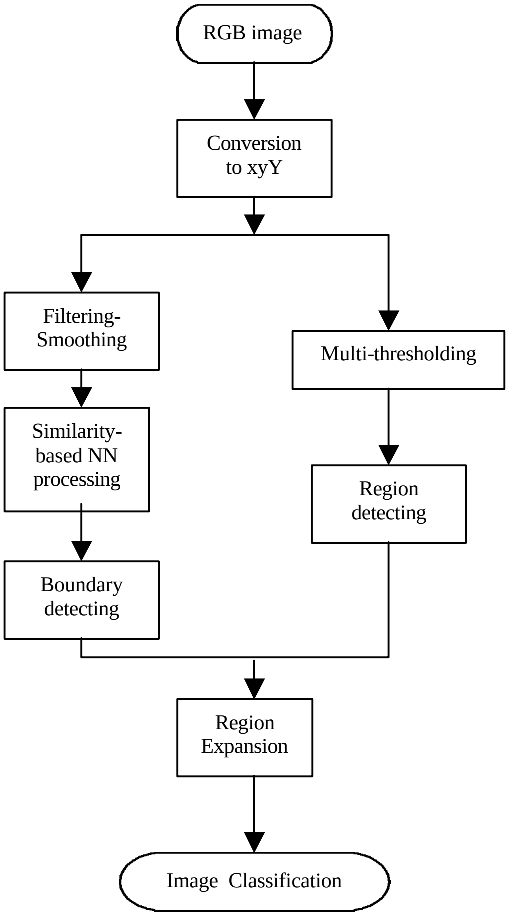

Fig. 1 Flowchart of the color texture processing

3. *RGB→ xyY Conversion*

The original images are given in RGB and are changed to xyY because luminance and chromaticity values give more relevant information in terms of region detection. The values of the converted components in terms of RGB values are given by the following equations[1]:

$$\mathbf{x} = \frac{X}{X + Y + Z}. \qquad (1)$$

$$\mathbf{y} = \frac{Y}{X + Y + Z}. \qquad (2)$$

$$X = 0.607 * R + 0.174 * G + 0.2 * B \qquad (3)$$

$$\mathbf{Y} = 0.299 * R + 0.587 * G + 0.114 * B \qquad (4)$$

$$Z = 0.066 * G + 1.111 * B \qquad (5)$$

With these values the Y component is ready for Luminance Processing but the xy components have to be changed to a one dimensional chrominance value (cv). The xy values are equal or greater than zero and equal or less than one. The interval of the xy values (from 0 to 1) is divided in *k* number of intervals. The xy values are related to an equivalent value within 0 and *k*. Using the new equivalent value for each x and y the new one dimensional chrominance value is given by the following formula:

$$cv = y + k * x \qquad (6)$$

4. *Luminance Processing*

The Luminance Processing extracts the boundary of ROI in the image with the intensity component Y. The Luminance Processing has three stages: 1) Filtering-Smoothing 2) Similarity-based NN processing 3) Boundary detecting.

The Filtering stage is performed by a set of Gabor filters, which are capable of detecting basic image features. The Gabor filters are gaussian-modulated sinusoids.

$$g(m,n) = e^{-(m^{2} + n^{2} / 2\sigma^{2})}\ e^{-2j\pi\phi(m \cos \theta + n \sin \theta)} \qquad (7)$$

Only the real part of the complex exponential is used because it considerably reduces the number of computations.

$$g(m,n) = e^{-(m^{2} + n^{2} / 2\sigma^{2})} \cos[2j\pi\phi(m \cos \theta + n \sin \theta)] \quad (8)$$

where m,n are the pixel coordinates. The orientations (θ) of, 0 and 90 degrees, and the frequencies (ϕ) of 0.25 and 0.50 are used in this paper. This should be fixed based on the types of images to be classified. This means that a total of four filters are used to produce four filtered images ready to be smoothed. The iterative smoothing is done using a 3 x 3 window to remove small variations in the image that can be considered as random features that do not represent a ROI.

The last step in the Luminance Processing is the Boundary Detection. A single-level neural network is used to combine the four filtered images and produce one boundary image. The inputs to a given node correspond to the eight neighbors of a pixel. The input weight for each pixel is determined by the similarity of the central pixel and its neighbors. The measure of similarity is defined next.

$$S_{ij} = \frac{\min (P_i,P_j)\ .}{\max (P_i,P_j)} \qquad (9)$$

where P_i is the central pixel and P_j a neighboring pixel. The functions 'min' and 'max' represent the min and max differences between P_i and P_j respectively. The four values of S_{ij} from the four filtered images are averaged to obtain a combined input weight.

$$W_{ij} = \tfrac{1}{4} \sum_{k=1}^{4} S_{ij}^{k} \qquad (10)$$

k identifies each of the four filtered images.

The neural network uses a hard-limiting nonlinearity which is described as follows:

$$O_N = K(\alpha - \eta) \quad \begin{array}{l} =1, \text{ if } \alpha<\eta\text{; else} \\ = 0 \end{array} \qquad (11)$$

where α is the average node input and η is a specific node threshold that can be found experimentally. Function K produces a node output of either zero or one depending on the similarity of the central pixel and its 8 close neighbor pixels. Zero value means that the specific central pixel is likely to be on the boundary of a region, while one implies that it is in the interior of a region. If we define 0 output as black and 1 as white the resulting boundary image will be a primarily black image with white borders for the different regions.

5. Chrominance Processing

The chrominance processing consists in computing the histogram of the chrominance image. The largest values (peaks) that are obtained correspond to large clusters of pixels having similar chromianance values. With this stage possible ROI are identified. The peaks are paired with a unique integer number to identify a whole cluster region to which the peak belongs. Assuming 8 bits per pixel the possible values for a peak are from 1 to 255. The peak number is assigned in a determined order or direction (left to right usually) starting with **1**, followed by **1 + [255/*np*]**, etc. *np* is the number of histogram peaks. The resulting image (Region Image) is divided into a number of regions equal to the number of histogram peaks.

6. Region Expansion

This stage of the process consists of combining the information from the boundary and the region images to cover certain areas that might not be included in a specific cluster area. The process starts with the identification of a boundary point from the boundary image. The boundary Image saves time because it identifies the regions in which a segment is missing and makes impossible to close the region at the region image. Then the region expansion is done with the region image using the expansion algorithm. The expansion algorithm starts in the pixel determined by the boundary region and moves in each four possible directions to search for similarity between the central pixel and its four neighbors. This process is done by windows (5x5 usually) because doing it in smaller windows would take too long. To state the threshold function a similarity value is defined.

$$sv = \frac{\max - \min}{\max} \qquad (12)$$

where min and max represent the smallest and largest number between the pixel neighborhood average and the seed pixel average.

7. Algorithm Implementation

In the color image processing, first chrominance and then luminance are processed. An example flower image is shown in Fig. 2 (a). Figures 2(b) and (c) show the chrominance and luminance images, respectively.

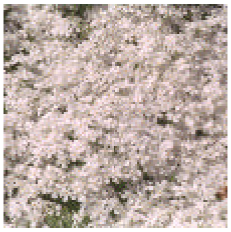

Fig. 2(a) Flower image

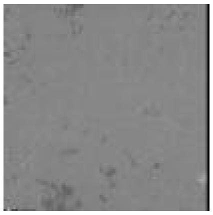

Fig. 2(b) Chrominance image

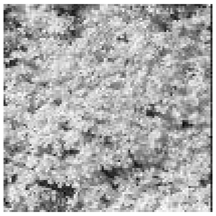

Fig. 2 (c) Luminance image

Training method

Samples of size 32 x 32 are taken from the luminance images and convolved with 4 Gabor filters of the same size. A smoothing of the filtered image over 3x3 windows is done. Additive mask features are computed on the smoothed images. These feature vectors are computed for 64 samples from 6 different color images. This forms the training matrix for the color images.

Classification method

Testing samples of size 32 x 32 are extracted from the color luminance images and the algorithm of Gabor filtering, smoothing are applied. Feature vector is computed for these samples again and a distance Euclidean and Mahalanobis classifiers are used to classify the unknown samples.

Minimum Euclidean distance:

$$d_j(x) = \sum_{q=1}^{Q} \left(x_q - m_{i,q}\right)^2 \qquad (13)$$

Mahalanobis distance

$$d_j(x) = \{(x-m)^t C_j^{-1} (x-m_j)\} \qquad (14)$$

x is the feature vector to be classified, m is the mean value of the training vectors, τ is the transpose, C is the covariance matrix.

The classifiers are used to recognize 32 different unknown samples from each of the 6 color images.

The results are shown comparing performance of Euclidean classifier and Mahalanobis classifier with and without smoothing. The Euclídean classifier performs poorly compared to Mahalanobis with or without smoothing. The results are better with Mahalanobis classifier which is also computationally more efficient. The classification results are shown in Table 1.

TEXTURE	PCC (%)		
	Gabor filtering		**Gabor Filtering Smoothing Euclídean**
	Euclídean classifier	**Mahalanobis classifier**	
Flower	71.85	65.625	90.625
Sand	62.5	78.125	75
Brick	12.5	50	3.125
Water	100	100	100
Grass	59.375	65.625	46.875
Food	28.125	56.25	9.38
Average	**55.73**	**69.27**	**54.17**

Table 1. PCC stands for percentage of correct classification

8. Conclusions and Future Work

In this paper an algorithm for recognition of color textures is presented. The algorithm has been implemented up to the part of Gabor filtering. Some results are presented at this stage which are not very good but promising. The maximum correct classification rate obtained is 69.27% with only luminance processing of the image.

In future work chrominance processing will also be implemented which should help improve the classification rate. More results will be obtained for different types of color images using the algorithm.

9. References

[1] George Paschos and Kimon P. Valavanis, A Color Texture Based Visual Monitoring System For Automated Surveillance, *IEEE Transaction on Systems, Man and Cybernetics*, Vol. 29, No. 1, pp 298-306, 1999.

[2] Si Wei Lu and He Xu, Textured Image Segmentation Using Autoregressive Model and Artificial Neural Network, *Patern Recognition*. Vol 28, No. 12, pp. 1807-1817, 1995.

[3] Rafael E. Gonzalez and Richard E. Woods, Digital Image Processing, Addison-Wesley Publishing Company, pp 413-478, 1992.