

# On the Segmentation of Color Cartographic Images

Juan Humberto Sossa Azuela, Aurelio Velázquez, and Serguei Levachkine

Centro de Investigación en Computación – IPN  
Av. Juan de Dios Bátiz s/n, Esq. Miguel Othón de Mendizábal  
UPALM-IPN Zacatenco, México. D. F. C.P. 07738  
avela@cic.ipn.mx, hsossa@cic.ipn.mx, palych@cic.ipn.mx

**Abstract.** One main problem in image analysis is the segmentation of a cartographic image into its different layers. The text layer is one of the most important and richest ones. It comprises the names of cities, towns, rivers, monuments, streets, and so on. Dozens of segmentation methods have been developed to segment images. Most of them are useful in the binary and the gray level cases. Not to many efforts have been however done for the color case. In this paper we describe a novel segmentation technique specially applicable to raster-scanned color cartographic color images. It has been tested with several dozen of images showing very promising results.

## 1 Introduction

Color cartographic images are very important commercially speaking but complex at the same time. They contain many information usually divided in layers: the text layer, the river layer, the symbol layer, and so on. Segmentation of an image like these into its layers is a complex task because in general the information from the different layers is mixed. Letters might share the same color as text, river traces or street paths.

One of the most important layers in any cartographic map is the text layer. It can help us to identify cities, towns, rives, lakes, monuments, streets, etcetera. One might apply a simple thersholding technique trying to separate the text layer from a color cartographic image. Thresholding has been applied with success for many years to isolate objects of interest from their background. Dozens of techniques have been described in the literature. Refer for example to [1]-[6] for isolated examples and to [7] and [8] for good surveys.

Image thresholding is mostly applicable to the case of high contrasted images (images where objects contrast strongly with respect to the background). In general, however, this is not the case because objects in most gray-level images normally share the same gray-levels. Almost any simple thresholding method will not adequately (as requested by humans beings) isolate the desired objects. Intuitively, trying to do the same with color images will be even more complicated. Several segmentation techniques have developed during these last years for the color images. For a very good survey see [9].

In this paper we introduce a novel technique specially designed to work with color images. The proposed technique has been tested with several raster-scanned color cartographic images with excellent results.

## 2 Preliminaries

Most of the existing color-based image segmentation techniques use the R, G and B information contained in the image to accomplish the task. Our proposal also uses this information plus the average of the combination of two of them, the average of the three, and the transformation of the color image to an intensity image, transforming it into an image YIQ (Y - luminance, I - In-phase and Q - Quadratic) by selecting only the luminance component. The eight images considered by our segmentation algorithm are thus the following:

$$\begin{aligned} M1 &= "R", \\ M2 &= "G", \\ M3 &= "B", \\ M4 &= \text{int}((\text{"R"} + \text{"G"})/2), \\ M5 &= \text{int}((\text{"R"} + \text{"B"})/2), \\ M6 &= \text{int}((\text{"G"} + \text{"B"})/2), \\ M7 &= \text{int}((\text{"R"} + \text{"G"} + \text{"B"})/3), \\ M8 &= \text{color\_to\_gray}(\text{"RGB"}). \end{aligned}$$

Let us denote these images as the 8 M-images of a standard color image.

## 3 The Novel Technique

The proposed technique to segment raster-scanned color cartographic images is divided in three main stages: Image pre-processing, image processing and image post-processing. Each one of these three stages is next described.

### 3.1 Image Preprocessing

During this stage, the goal is to accentuate the pixels of the objects of interest in the image: the text pixels in our case, those coming from alpha-numeric characters. This is done by first decomposing the test image into its 8 M-images. Fig. 1 shows an image and its eight M-sub-images. Each M-sub-image is next adjusted in contrast fit the range of 0 to 255 gray levels. The same original 8 M-sub-image are histogram-equalized. This results in 24 sub-images (8 original M-sub-images, 8 normalized sub-images and 8 equalized sub-images). Figures 2 and 3 show, respectively the 8 normalized sub-images and the 8 equalized sub-images of the 8 M-sub-images of Fig. 1(b).

### 3.2 Image Processing

The goal here is to further emphasize the pixels of interest. This stage is divided in two steps: Image pre-thresholding and image post-thresholding. These steps are next explained.

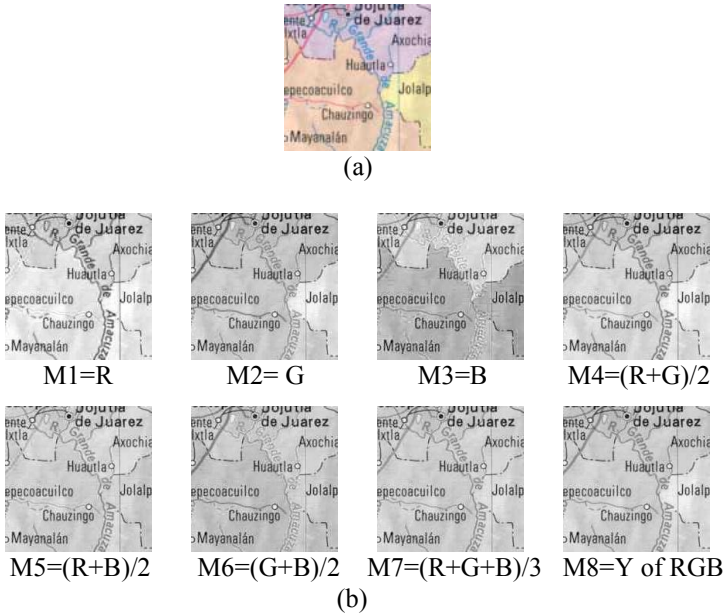


Fig. 1. (a) A color cartographic image and (b) its 8 M-images.

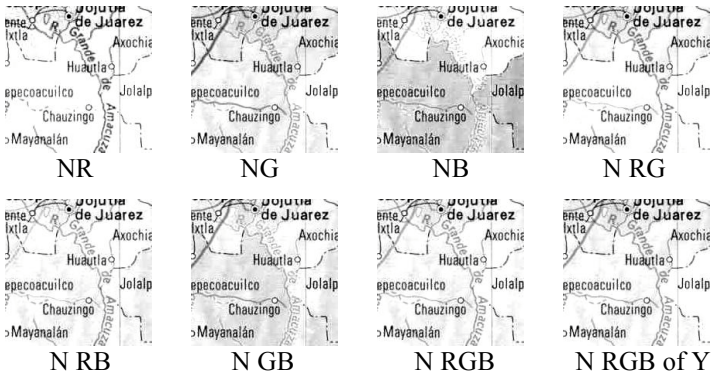


Fig. 2. Normalized sub-images of the eight M sub-images of Fig. 1(b).

#### 3.2.1 Image Pre-segmentation

The 24 images shown in Figs. 1(b), 2 and 3 are next thresholded to get their 24 binary versions. The goal as mentioned before is to strongly emphasize the presence of text in the image. Next we describe how to automatically select the desired threshold.

Before describing how to choose the desired threshold let us say some words. It is well known that threshold selection is not only a critical but also a very difficult task in the general case. A bad threshold would result in very bad results; a good threshold would provide in the contrary very nice results.

One very well known way to obtain the threshold is by using the histogram of the image (see for example [10]). We have used this approach at the beginning of our research with very bad results. We have experimented with several algorithms. After many tries we have retained the following one:

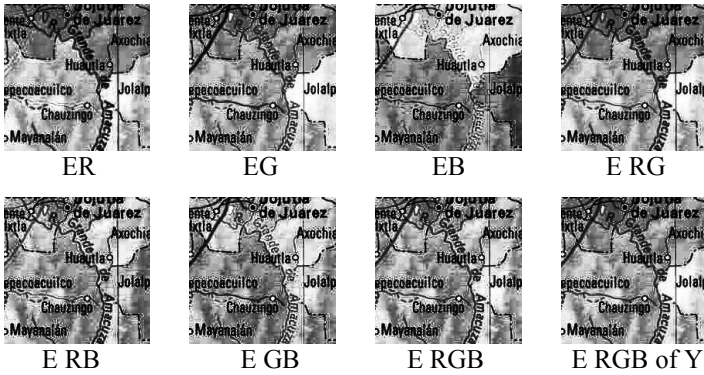


Fig. 3. Histogram equalized sub-images of the eight sub-images of Fig. 1(b)

For each of the 24 for sub-images:

**Step 1.** Convolve sub-image with the following modified Prewit’s edge masks in both the x and the y directions:

$$\begin{matrix}
 & \begin{matrix} -1 & 0 & 1 \end{matrix} \\
 1/6 & \begin{matrix} -1 & 2 & 1 \\ -1 & 0 & 1 \end{matrix}
 \end{matrix}
 \qquad
 \begin{matrix}
 & \begin{matrix} -1 & -1 & -1 \end{matrix} \\
 1/6 & \begin{matrix} 0 & 2 & 0 \\ 1 & 1 & 1 \end{matrix}
 \end{matrix}$$

**Step 2.** Instead of computing the magnitude of the gradient as usual, add pixel-by-pixel the two resulting images to get once again one image, say  $g$ .

**Step 3.** Add all the gray-level values of this image to get just one number.

**Step 4.** Divide this number by the total size of the image (the number of pixels of the image) to get the desired threshold  $u$ . If  $g$  has  $M$  rows and  $N$  columns then,

$$u = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N g(i, j) \tag{1}$$

**Step 5.** Threshold the sub-image with this  $u$ .

Repeat steps 1 to 5 for the 24 sub-images to get the desired binary version.

We have found that a threshold computed like this provides very good results. We could also use another edge detector such as the Roberts or Sobel operators with very similar results.

In the case of histogram-equalized images the applied threshold is the original one  $u$  (obtained by means of Eq. (1)) but divided by seven. If the original threshold was applied directly to an equalized image a completely black image would result. This factor was found empirically. Fig. 4 shows the 24 binary versions of the 24 sub-images shown in Figs. 1(b), 2 and 3.

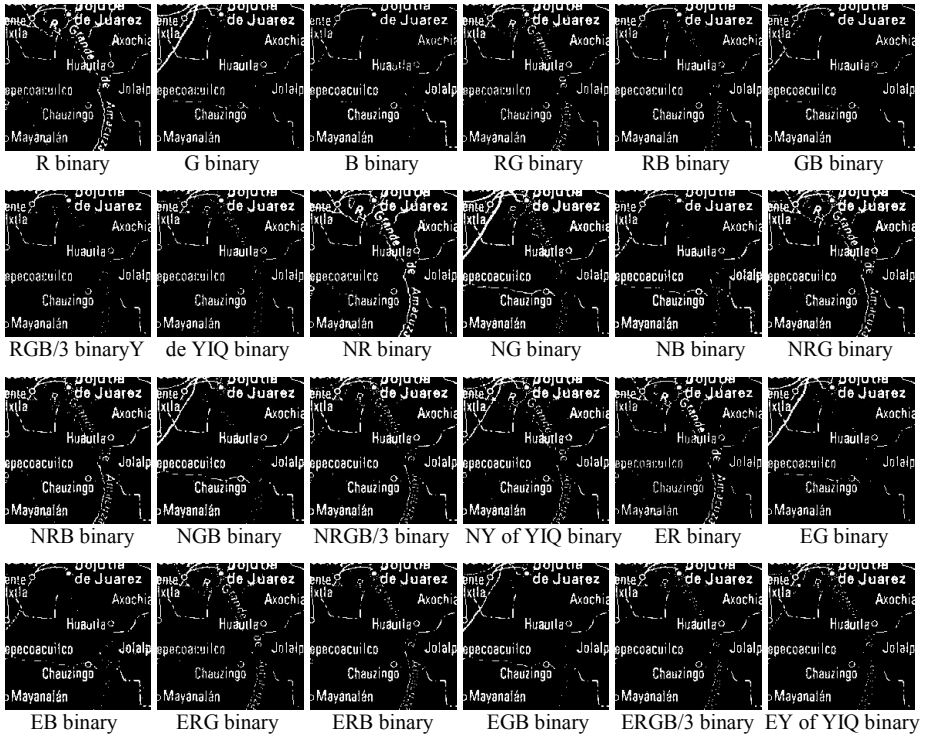


Fig. 4. The 24 binary versions of the 24 sub-images shown in Figs. 1(b), 2 and 3

**A comment.** We have used an edge detector to get the desired threshold due to letters and numerals are associated in their frontiers with abrupt changes with respect to the image’s background. The use of an edge detector would thus accentuate letters and numerals at least in their frontiers. The application of a method like this would also emphasize the presence of rivers, country borders and so on. These, however, could be later eliminated if desired, for example, to only separate the alphanumeric layer of the image.

### 3.2.2 Image Post-segmentation and Region Labeling

The white pixels in each of the 24 binary sub-images are representative of strokes of a letters or numerals, the border of a river, the limit between cities, and so on. They appear however fragmented and not clustered to form connected regions of letters or numerals, in some images appear incomplete, in some others they appear complete.

To solve this problem we have used the following heuristic method. We first obtain one image from all the 24 binary sub-images by simple pixel-by-pixel addition of all

the 24 images. The value of a pixel in this new image will oscillate between 0 and 24. To obtain again a binary image we verify if the sum at a given (x,y) position is greater than 21 (a manually selected value). If it is true we put a 1 into the buffer image and zero otherwise. We have thus another binary image with 1's in the most probably positions representing letters, numerals and other symbols and 0's in the backgrounds areas. Fig. 5(a) shows the resultant binary sub-image.



**Fig. 5.** (a) Binary sub-image after thresholding the image resulting when adding pixel-to-pixel the 24 sub-images shown in Fig 4. (b) Resulting image after filling the gaps on (a). (c) Resulting image after eliminating the pixels on Fig. 1 as explained in the text.

We next apply a standard labeling algorithm [10] to this image to get adjacent regions representing the desired elements (letters, numerals, and other symbols). Let's call this labeled image the P-image.

### 3.3 Image Post-processing

If you take a look to Fig. 5(a), the adjacent connected regions in this image obtained with the procedure just described appear not well emphasized as desired and yet fragmented.

The gaps between isolated regions must be filled to get the complete desired regions. These gaps are filled by means of the following procedure:

#### **Procedure FILLING GAPS**

The input is a P-image.

1. Select a connected region of pixels in the P-image. Dilate this region with a 3x3 structural element, from 3 to 15 times, depending on the number of pixels of the region. Each 10 pixels add one morphological-dilation to the process. This process gives as a result a square window whose size is given by the number of dilations. Let us call this resulting square window the D-mask.
2. AND-Image-mask (see [11], pp. 50 for the details) the 8 M-sub-images with the D-mask obtained in step 1 in order to compute the average gray-level and the standard deviation of this region. Only the pixel values under the 1's in the D mask are taken into account.
3. Turn off all pixels in the D-mask if the corresponding gray-level value if any of the M-sub-image is greater than the gray-level average plus the standard deviation value obtained in step 2. This allows, on the one hand, to eliminate the undesired background pixels added to the alphanumeric character during the

dilation process. On other hand, this step permits to aggregate missing pixels of the character.

4. Apply steps 1-3 to all labeled regions inside the P-image.

The output is another image (the F-image) with the isolated characters with gaps filled.

Fig. 5(b) shows the resulting image after applying the filling gap procedure just described to the image shown in Fig. 5(a). Note how the gaps between isolated regions have disappeared, and the letters and numerals now appear more complete.

Fig 5(c) shows the resulting image after the segmented pixels (those appearing in Fig. 5(b)) were eliminated from the original image (Fig. 1(a)). Fig. 5(c) was obtained by using Fig. 5(b) as a mask. If a pixel in Fig. 5(b) is 1 (white), its corresponding pixel in Fig. 1(a) is substituted by an interpolated value obtained by averaging the surrounding pixels of the pixel.

From Figs. 5(b) and 5(c), you can see that some letters, rivers and other small objects did not were separated from the original image (Fig. 1(a)). They do not appear in white in Fig. 5(b), they do appear however in Fig 5(c). We would like of course these objects to be also segmented. To accomplish this we have applied our technique to other RGB combinations: RG, RB and GB. As we will next see the remaining objects of interest will be accentuated and thus also segmented.

The M-images for an image are not 8 but 5. For the RG combination the 5 M-images are: M1, M2, M4, M7 and M8. For the RB combination were M1, M3, M5, M7 and M8. For the GB combination were M2, M3, M6, M7 and M8. For a given combination now five normalized and five equalized images were obtained. Fifteen thresholded images as explained in Section 3.2.1 were obtained from each of the three combinations and their 10 images. Each set of ten images was processed as explained in Sections 3.3.2 and 3.3 to get the final images with gaps filled. Fig. 6 shows the three resulting images. Note how the missing letters in Fig. 5(a) (appearing in Fig. 5(c)) are now well emphasized in Fig. 6(a). The path of the river under “R. Grande Amacuzá”, also missing in Fig 5(a) appears also accentuated in Fig. 6(a). Note also how the three rivers missing in Fig. 5(a) (appearing in red in Fig. 5(c)) are well emphasized in Fig. 6(c).

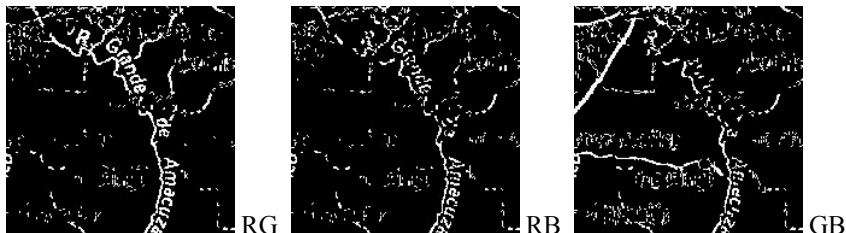


Fig. 6. Resulting images after processing the combinations RG, RB and GB

One important feature of our proposal is that each combination emphasizes different elements of the original image. The RGB combination strongly emphasizes black objects. The RG combination emphasizes mostly blue objects such as river names and their paths and the GB combination emphasizes the red elements such as roads. The RB combination in this case appears to be useless.



As we are interested in text characters, we can now take each character from each resulting image along with its position and stock them into a file for further processing.

## 4 Results

In this section additional testing results are shown. Figure 7 (top) shows six raster-scanned color cartographic images. All of these were processed as described in the previous sections to get their final segmented versions. Only the RGB combination was used. Figure 7 (down) shows these final versions. You can appreciate that the final results are good enough. The methodology has been tested until now with more than 100 images with very promising results.



Fig. 7. Other images (top) and their segmented results only on their RGB combinations

## 5 Conclusions

In this short paper a novel thresholding approach applicable for color images has been proposed. It incorporates three main stages: Image preprocessing, image processing and image post-processing.

This approach has been tested with many raster-scanned color cartographic images giving promising results.

The resulting images are being now analyzed by another module that will allow to separate alphanumeric characters from other objects. The goal is to isolate as much as possible each alphanumeric character and determine its identity by a trained classifier.

## Acknowledgments

The authors would like to express their acknowledgment to CONACYT under grants 34880-A and 32019-A and to the Centro de Investigación en Computación of the IPN from Mexico for their support for the development of this research.



## References

1. N. Otsu, A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man and Cybernetics*, 9(1): 62-66, 1979.
2. J. N. Kapur, P. S. Sahoo and A. K. C. Wong, A new method for gray-level picture thresholding using entropy of the histogram, *Computer Graphics and Image Processing*, 29:273-285, 1985
3. J. Kittler and J. Illingworth, Minimum error thresholding, *Pattern Recognition*, 19:41-47, 1986.
4. P. Sahoo, C. Wilkings and J. Yeager, Threshold selection using Renyi's entropy, *Pattern Recognition*, 30(1):71-84, 1997.
5. L. Li, J. Gong and W. Chen, Gray-level image thresholding based on Fisher linear projection of two-dimensional histogram, *Pattern Recognition*, 30(5):743-749, 1997.
6. X. J. Wu, Y.J. Zhang and L. Z. Xia, A fast recurring two-dimensional entropic thresholding algorithm, *Pattern Recognition*, 32:2055-2061, 1999.
7. J. S. Weska, A survey of threshold selection techniques, *Computer Graphics and Image Processing*, 7:259-265, 1978.
8. P. S. Sahoo, . Soltani, A. K. C. Wong and Y. Chen, A survey of thresholding techniques, *Computer Graphics and Image Processing*, 41:233-260, 1988.
9. H. D. Cheng, X. H. Jiang, Y. Sun and J. Wang, Color image segmentation: advances and prospects, *Pattern recognition*, 34(12):2259-2281, 2001.
10. R. C. Gonzalez and R. E. Woods, *Digital image processing*, Addison Wesley Pub. Co. 1993.
11. S. E. Umbaugh, *Computer Vision and Image Processing: A practical Approach using CVIPtools*, Prentice Hall PTR, NJ, 1998.