A System to Segment Text and Symbols from Color Maps

Partha Pratim Roy[1], Eduard Vazquez[1], Josep Lladós[1], Ramon Baldrich[1], and Umapada Pal[2]

[1] Computer Vision Center, Universitat Autònoma de Barcelona, 08193, Bellaterra (Barcelona), Spain
[2] Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata - 108, India

Abstract. Automatic separation of text and symbols from graphics in document image is one of the fundamental aims in graphics recognition. In maps, separation of text and symbols from graphics involves many challenges because the text and symbols frequently touch/overlap with graphical components. Sometimes the colors in a single character are gradually distributed which adds extra difficulty in text and symbol separation from color maps. In this paper we proposed a system to retrieve text and symbol from color map. Here, at first, we separate the map into different foreground layers according to color features and then in each layer, connected component features and skeleton information are used to identify text and symbol from graphics on the basis of their geometrical features. Lastly, segmentation results of the individual layers are combined to get final segmentation results. From the experiment we obtained encouraging results.

1 Introduction

Automatic discrimination of text/graphics in document image is one of the fundamental aims in graphics recognition [2],[3],[6]. Here, the aim is to segment the document into two layers: a layer assumed to contain text and symbols and the other one containing the rest of graphical objects such as street, river, border of the regions etc. The problem has received a great deal of attention in the literature because of the different processing approach of text and graphics. At the component level the problem is not too intensed. The spatial distribution of the components and their sizes, can be measured in a number of ways, and fairly reliable classification can be obtained. Difficulties arise however, when either there is text and symbol embedded in the graphics components, or text and symbol touched with graphics. It includes, frequent intersection of text and symbols with graphical lines and curves and segmentation of such documents is very difficult. The separation problem of text and graphics intersections has not yet been dealt with successfully although there exist many pieces of published paper.

Fletcher and Kasturi [6] proposed an algorithm to separate text-string from mixed graphics. They used simple heuristics based on the characteristics of text characters. The method is insensitive in text font style, size and orientation. One

W. Liu, J. Lladós, and J.-M. Ogier (Eds.): GREC 2007, LNCS 5046, pp. 245–256, 2008.

of the assumptions was that the text character should not touch with graphics or other characters and each text character forms an isolated component. Another assumption of the method is that, the character components of a string are aligned straight. Luo et al. [1]uses the directional mathematical morphology approach for separation of character strings from maps. The idea is to separate all long linear segments by directional morphology and histogram analysis of these segments. Long segments are considered as part of graphics; effectively leaving small text character segments. Tan et al. [7] illustrates a system to extract text strings from a mixed text/graphics image using *Pyramid* structure. Multi-resolution representations of such a pyramid structure help to select different regions for segmentation. Cao and Tan [3] proposed a method of detecting and extracting text characters that are touched to graphics. It is based on the observation that the constituent strokes of characters are usually short segments in comparison with those of graphics. They looked for the lines in the overlapped region on the vectorized image to interpret intersection of text and graphics. More consolidated method is proposed by Tombre et al. [2]. This method is based on the analysis of the connected components. The algorithm has covered a number of improvements to make it more stable for graphics-rich documents.

In color maps the color carries a lot of information which is useful for separation. Using color features, complexity of text/graphics separation can be reduced drastically. Because, in color maps, different colors are used to represent text and symbol from graphics. By color properties, one can separate touching/overlapped text symbols from graphics without exploring much graphics recognition methodologies. There exist color segmentation methods for separating colors used in a document. Recent work includes a variety of techniques: for example, morphological watershed based region growing [12], JSEG segmentation [13], Mean Shift algorithm [15] etc. If an image contains only homogeneous color regions, clustering methods in color space [14,16] are sufficient to handle the problem. However, in color text documents sometimes even, it appears at first appearance, that the character seems to be printed in a single color, but actual measurements reveals that the colors in a single character are gradually distributed. This degradation effect of color in text layer sometimes makes it more difficult to segment. It causes over/under -segmentation and such over/under -segmentation creates problem in separating foreground layer (text graphics layers of different colors) efficiently. Here it is quite challenging to remove the noise and to extract the intended characters. Among color segmentation methods, clustering is one of the simplest, and has been widely used. The method proposed by Coleman and Andrews [16] is a clustering algorithm based on k-means which operates in an "unsupervised" mode and does not require training prototypes. Since, k-means algorithm requires the number of cluster to initiate, in order to have this information, the knowledge about the image data is necessary. This method iterates on a number of clusters, and evaluate the quality of the clusters by within-cluster and between-cluster scatter matrices. Moreover, the influence of resolution of the input image is an important factor in color document analysis. When a color document is scanned at high resolution, mesh noise occurs in the

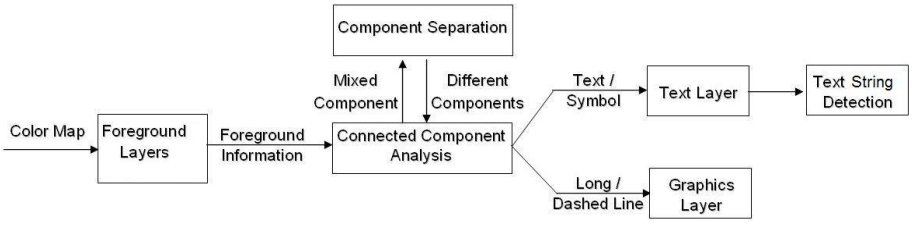


Fig. 1. (a)Block diagram of our approach

digital color image resulting over-segmentation, when a segmentation algorithm is applied. Hase et al. [17] studied offset printing color documents such as book covers, posters. They discussed a method to absorb the variation of color distribution of color segmentation. Their algorithm tries to prevent over-segmentation and fusion with the background while maintaining real-time usability. They have described a selected local color averaging technique to remove the problem of mesh noise. Dhar and Chanda [10] presented a method for extraction and recognition of text and symbol from topographic maps. Here, color segmentation is done by a supervised clustering algorithm and in each color layer, symbols are recognized on the basis of symbol-specific geometrical features.

There are various types of color maps, and they contain thousands of text and symbol in different shapes. So, we need a sophisticated process, which can be used in general without many heuristic measures. The problems are mainly due to presence of graphical components with texts and symbols in different color layers. To handle such situations, the objective of this paper is to combine different features like color information, connected component analysis, skeleton information etc. to get better segmentation results. Our proposed technique is language and font independent. Also, our method does not depend on number of colors present in the document.

In our proposed method, the color map is first analyzed using color information and separated into different layers. Text/graphics components are extracted from these layers. A new algorithm, combining connected component and skeleton analysis, has been proposed to identify the isolated character, joined character, dash and long line components from each layer. The components in which both character and long line present due to overlapping are considered as mixed components. Using Hough transform and skeleton analysis, these mixed components are analyzed for their segmentation into character and line. Extracted character components are grouped into string according to their color and proximity features. Block diagram of our approach has been shown in Fig.1.

2 Color Segmentation and Foreground Layer Selection

The problem of foreground detection in color maps could be defined as that of detecting layers containing text, symbols and graphical objects. If the text

layer is assumed to be of dark color in a light background, then the problem can be solved by converting the RGB color-space to YIQ color-space, and applying a threshold in Intensity (Y channel) image. In color degraded image, this method is not efficient to separate foreground layer. For our map handling, due to image degradation, we performed the color segmentation to get different color layers and this is done by a combination of color feature and spatial information. This is followed by selection of foreground layer considering the features of text/graphics information. It is done by applying a heuristic measures on color volume and edge information of each color layers. The detail of color segmentation is discussed as follows.

First, we apply the method of Vazquez et al. [4] to find dominant colors in a d-dimensional histogram Ω_d. The method proposed is a two-steps operator. The creaseness operator, `MLSEC-ST`[5] is introduced in order to spurn non-representative data as well as to enhance meaningful information. This process assigns a high creaseness value at the center of elongated objects by means of divergence calculation of the Structural Tensor Field.

Formally, given a symmetric neighborhood of size σ_i centered at point x, say, $N(x, \sigma_i)$ the Structure Tensor is defined as:

$$S(x, \sigma_i) = N(x, \sigma_i) * (w(x) \cdot w^t(x)) \tag{1}$$

where a gaussian of standard deviation σ_d is used to make the calculation of ω. Afterward, if $\omega'(x, \sigma_i)$ is the eigenvector corresponding to the largest eigenvalue of $S(x, \sigma_i)$ then, the dominant gradient vector in the neighborhood of size proportional to σ_i centered at x is:

$$\overline{w}(x, \sigma_i) = sign(\omega'^t(x, \sigma_i) \cdot \omega(x))\omega'(x, \sigma_i) \tag{2}$$

Next, the creaseness value is associated to a point x_k, $\forall k \in \Omega$ as follows:

$$k(p) = -Div(\overline{w}_p) = -\frac{d}{r}\sum_{k=1}^{r} \overline{w}_k^t(\sigma_i) \cdot n_k \tag{3}$$

As a result, we have Ω'_d, a representation of Ω_d where each point is represented by its creaseness value. Next, we find the dominant structures of the histogram by applying a ridges extraction algorithm [11]. The ridges extraction procedure will join several points with a high creaseness under one unique ridge. Therefore, we will have as many ridges as dominant structures. This is done as follows:

If a given point $x \in \Omega_d$ is a local maxima, then we visit the neighbor of x which becomes a local maxima when we remove x from Ω_d. The process stops when it reaches a flat region.

Formally, let $\tau(\Omega'_d)$ be the set containing all local maxima in Ω'_d, and $neigh(x)$ be the r-connected neighborhood of a point x with a zero-crossing in its gradient for a given direction, i.e. we calculate the gradient values not for all points but for ridge points and its closer neighbors. We also define $\eta(x, n_j)$ as the common neighbors between x and n_j , where $x \in \Omega'_d$ and $n_j \in neigh(x)$:

$$\eta(x, n_j) = \{neigh(x) \bigcap neigh(n_j)\} \tag{4}$$

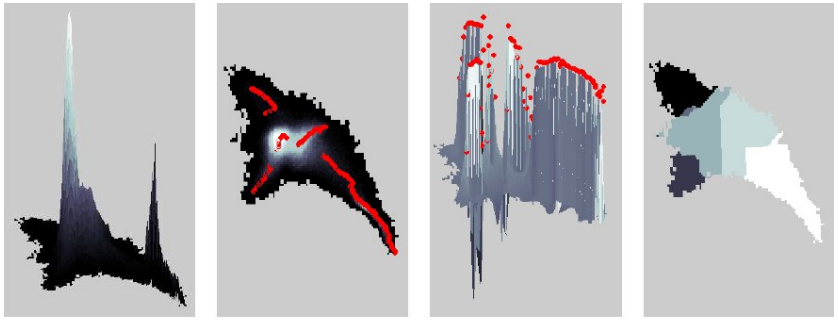


Fig. 2. (a)Ω_3, original 3-dimensional distribution. (b)Ω'_3 Creaseness representation of a) and ridges found are represented with dots. (c)Ridges found in d)fitted on original distribution. 2-Dimensional view. (e)Clustering of the original distribution.

It is worth to note that neither x nor n_j are included in $\eta(x, n_j)$. Then, we define the ridge points in a creaseness representation image Ω'_d, as:

$$\tau_z(\Omega'_d) = \tau_{z-1}(\Omega'_d) \bigcup \{n \in neigh(l) \mid l \in \tau_{z-1}(\Omega'_d), \mu(l, n) = 0\} \tag{5}$$

$$\mu(x, n_j) = \sharp \{y \in \tau(x, n_j) \mid \Omega'_d(y) \geq \Omega'_d(n_j)\} \tag{6}$$

Next, we perform a clustering to this distribution. For this, the points in $\tau_z(\Omega'_d)$ are used as marks in a watershed procedure. To avoid the drawbacks of the watershed algorithm [11], the method is combined with a Voronoi spatial partitioning. This will perform a clustering of the original histogram Ω_d in as many regions as dominant colors are found. Fig. 2 shows an example of the whole process. Concretely, Fig. 2(a) shows a synthetic 3-dimensional distribution, i.e., Ω_3. Creaseness representation of it, Ω'_3, is depicted in Fig. 2(b) where dots are the ridges found. The same ridges obtained in Ω'_3 are fitted in the original distribution in 2(c). Finally, the clustering of Ω_3 is showed in Fig. 2(d) and there are 5 different cluster in this figure. Based on the number of cluster, color volume of each cluster and edge information of each color layer, the number of foreground layers is decided.

3 Text and Symbol Separation

After selecting different foreground layers we separate text and symbol from graphics in each of the layers. Different steps used in each foreground layers are discussed as follows.

3.1 Connected Component Analysis with Skeleton Information

Text, symbol and graphical lines are present in foreground layer. We considered the connected component analysis developed by Tombre et al. [2] for initial

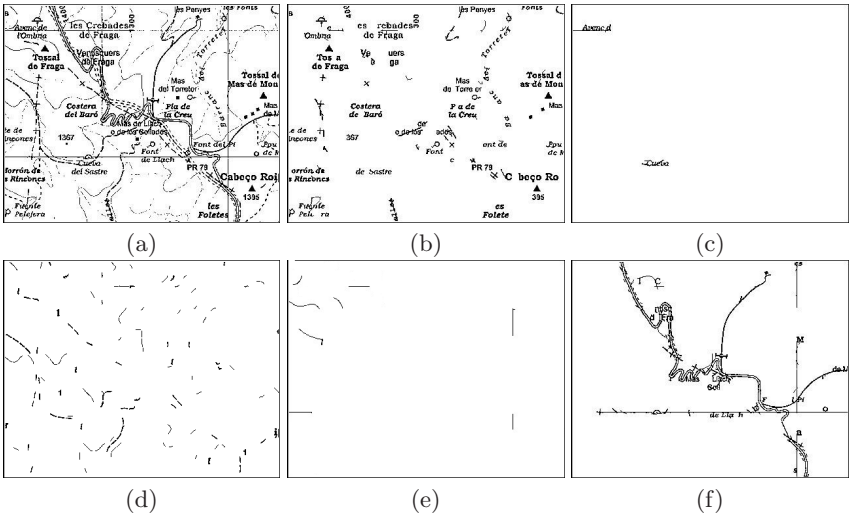


Fig. 3. (a)Example of a foreground Layer (b)Isolated text and symbol (c)Joined characters (d)Small elongated components (e)Long line (f)Mixed components

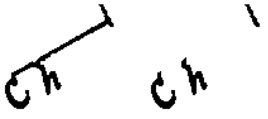
segmentation. A few criteria based on geometrical features of the connected component are good enough to group a component into one between text or graphics layer. But, there are some constraints. For example, some characters cannot be split due to touching. We will call them as "joined characters". If joined characters touch with long lines, they will not be separated by simple rules, because their features will be different from isolated character. We integrate skeleton information along with geometrical features to detect the long segments and to analyze them accordingly. We separate the components into 5 groups namely, Isolated characters, Joined characters, Dash components, Long components and Mixed components. In skeleton image of each component, if there exists no long segment, then the component is included into one of the isolated character/symbol or joined character or dash component group. Otherwise, it is considered as mixed or long component. The description of each of them is given below. The Fig. 3 shows different components of a foreground layer.

In *Isolated Character* group, normal text alphabets and small symbols are included. These are selected by the connected component size histogram analysis [2]. *Joined Characters* consists of the components where more than one isolated character touch each other. These connected components have larger aspect ratio than isolated characters and do not contain long skeleton segments. *Dash Characters* are mainly small elongated components. These include the dash segments from the dash line along with some isolated characters, such as "1","l" etc. These characters are combined into dash character group, because, at the pixel level analysis, they hold the same property as dash segments. The *Long Line* is the graphics layer of our algorithm. The segments obtained from the

skeleton of this component are all larger compared to the size of text characters. Straight and curve both types of line can be possible. *Mixed Component* consists of the components where both long line and isolated/joined characters are present. This happens due to overlapping with each other and we can not separate such mixed component during component labelling. Segmentation of the mixed components is done in two stages. In the first stage long straight lines are removed from the mixed components. Next,long curve parts are detected and removed from the remaining part of the image. These two stages are discussed as follows.

3.2 Long Straight Line Removal

We perform Hough Transform to detect the straight lines present in the binarized image. In Hough space, all the collinear pixels of a straight line will be found intersecting at the same point (ρ, θ), where ρ and θ identify the line equation. Depending on accumulation of pixels, the straight lines are sorted out. Some characters may touch with this straight line portion(see Fig. 4(a)). Hence our objective is to remove the non-character part from the straight line. We used stroke-width information in our approach to get the line width (L_w). Stroke width of a straight line segment is computed using the statistical mode of the black run-lengths, obtained by scanning the segment in horizontal, vertical and two diagonal directions. The portions of the line where the width is more than L_w, are separated from straight line. Fig. 4(b) shows the remaining part of the image after straight line removal in Fig. 4(a) by Hough Transform analysis.

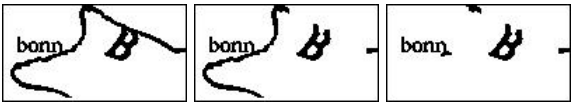


(a) (b)

Fig. 4. (a)Characters are joined with a long line in a Mixed component (b)Isolated characters after removal of straight long line

3.3 Long Curve Line Removal

According to text and graphics feature, it is assumed that the length of segments of the characters are smaller compared to that of graphics. The mixed-component segmentation method proposed by Cao and Tan [3] which is based on the continuation of the strokes in the skeleton works well for documents, where the text and lines are of more or less thin in nature. But, there are some limitations. When a line touches a symbol or text of blob like shape (dense pixels), the thinned image is always not perfect for the arrangement of segments. It needs post-processing, which is a difficult job. To overcome the drawback we compute the skeleton and all the segments are decomposed at the intersection point of



(a) (b) (c)

Fig. 5. (a)A Mixed component (b)Long straight part is removed (c)Extracted part after removing the long curve lines

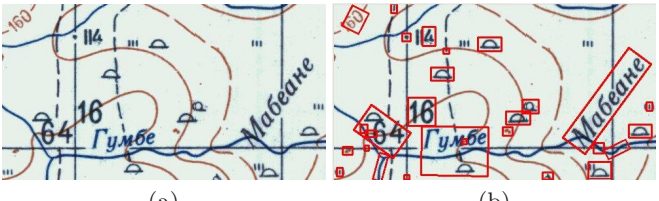
the skeleton. Based on the bounding box (BB) information of a segment, the major axis (L_s) is calculated as:

$$L_s = Max(Height_{BB}, Width_{BB}) \tag{7}$$

The segments having L_s larger than average character height (here, average height is obtained by averaging the heights of isolated characters) are chosen for elimination. The remaining portion after removal of long straight and curve line are considered as either isolated characters or joined characters according to their feature. For example, Fig. 5(a) demonstrates an initial mixed component with touching characters. The component after removing long straight line is shown in Fig. 5(b). After doing curve line removal, the remaining portions are shown in Fig. 5(c).

3.4 Character String Extraction

After passing through different separation methods, the mixed components will get separated. The long lines of mixed component will be in the graphics layer. Text and symbol will be in isolated or joined character layer. These isolated and joined characters are combined to get all the text and symbol components. Now we cluster the isolated characters into individual words. In general, the gap T_w between two words is larger than the gap T_c between two characters in a word and the grouping is formed by the characters of similar colors. Using this positional information and color information of different isolated character/symbol we cluster different words. For example see Fig. 6(b), where text and symbols are marked by rectangular box.



(a) (b)

Fig. 6. (a)Original Image (b)Segmented text and symbol layer. Here, segmented text and symbol parts are marked by rectangular box.

4 Experimental Results and Discussion

We have taken maps from different scripts to test our method. 26 maps are selected from "Spanish","English","Russian", and "Bengali" and the average size of the test maps are 350x450 pixels. We considered a large varieties of data for our experiment and some examples of such data are discussed as follows. The background can be of single color (Russian) or multi-color. Foreground contains numerous colors to represent text and symbol or graphical lines. The text characters are connected in Bengali maps. In other maps these are generally isolated, but sometimes joined text strings are also found due to printing or noise issue. In graphics part, lines can be dashed or continuous. There exist both straight and curve lines. The long continuous lines are touched/overlapped with text in many places. The overlapped text and lines sometimes are in similar color or in different color. In text string, the arrangement of characters are of both linear and curvilinear. The maps are not noise free always. In Bengali maps the noise/dithering effect is prominent.

Our proposed methodology combines color information, connected component analysis and skeleton information for segmentation. The thinning was done by the algorithm proposed by Ahmed and Ward [9] which works in rotation invariant nature. The length of a segment is used to classify it either as long lines or as small segments. String construction and joining missing characters are done using morphological operation. Scale invariance is also incorporated by computing histogram analysis of the components' size, considering aspect ratio of components.

From the experiment of color separation, we noted that our system shows very good results when different colors are distinct. For example, we obtained 100% color segmentation result in Russian map, because Russian maps contain distinct color layers. But from Bengali and Spanish map, we obtained 75% (on an average) results because of the color degradation. We computed this percentage on pixel label. In our dataset, after color separation, we found total 55 mixed components where long lines exist. Among them 22 were straight and rest were curve. The component label accuracy of removal of long straight lines are 100% and for long curve line it is 85%.

In Fig. 7c(2), it is interesting to see that we have recovered the number "79" from a curve line. This extracted number is shown by dotted box in Fig. 7c(2) and this number was touched with the curve line. There are 1400 isolated character and 60 joined characters in our dataset. From the experiment, we noted that more than 98% cases our method segment text and symbol of a map into isolated and joined character group. We also noted that, if a long line is not fully straight and it contains a sufficient straight part, this part will be detected by Hough Transform and will be removed leaving the other small parts. These small segments will fall into isolated and joined character group. This is same for long curve line. In skeleton analysis, a curve line may not be fully removed, if it visits many junctions in the travel path. This will result false alarms. For example, see near right-bottom part of Fig. 6(b), where we can see some false alarms, which have been generated because of this problem. In our present work, grouping

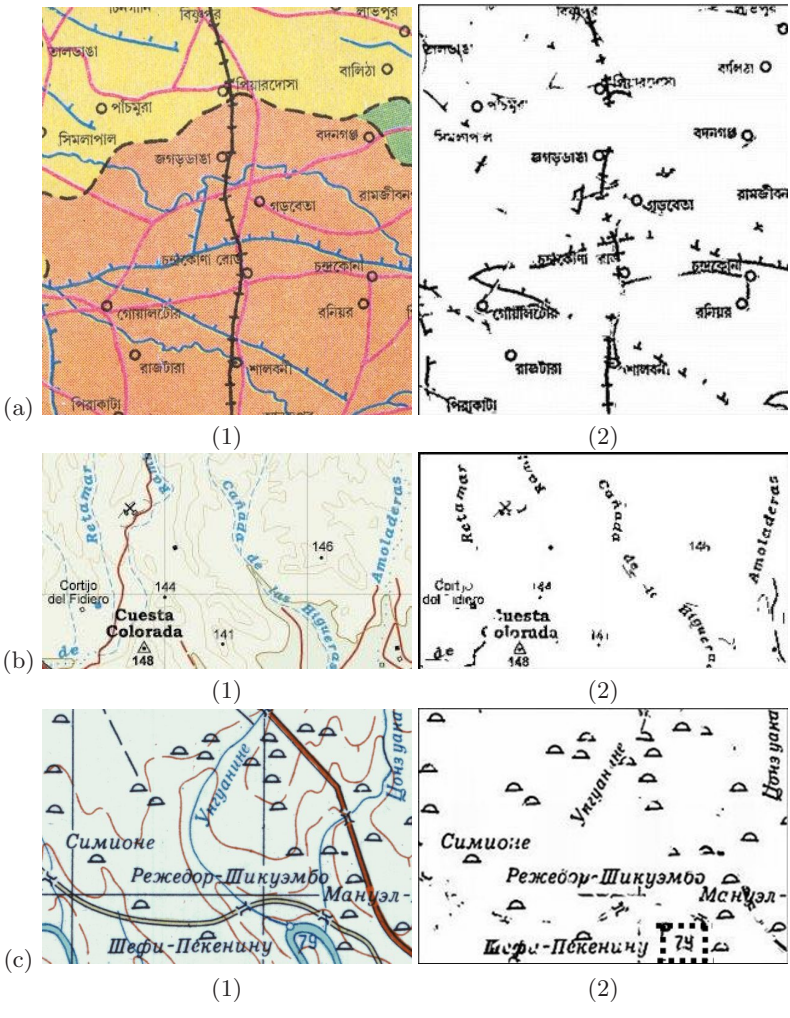


Fig. 7. Images in different scripts (a)Bengali (b)Spanish (c)Russian. In each script, (2) shows the extracted text and symbols of the corresponding color images (1).

of selected dash-like components into text/symbol layer is not considered. Due to this, missing of dash components like text characters may be obtained (see Fig. 6(b)). In future, we plan to use context information to solve these problem as follows. The isolated dash components are likely to be of dash lines, if they are arranged in a linear fashion. The other dash-shaped characters may be included for text part. For the false alarms generated by skeleton analysis, we should combine the neighborhood text region information to include them in text layer.

Almost all the previous approaches in text/graphics separation either used binarized image or converted the color image to binarized image for this purpose. The separation of text/graphics layer in degraded color image using color analysis

is not an easy task. There exists no methodology to evaluate the correctness of color segmentation result. The validity of the results vary according to human perception and thus focus for measuring in terms of qualitative rather than quantitative. Color separation analysis using creaseness operator reduces the amount of variability of color information effectively. In our test maps, where the color degradation is less, it outperforms. But in maps of "Spanish" and "Bengali" the noise is very prominent and we got some over segmentation. Here, we selected the foreground layers manually and used these layers for our text/graphics separation purpose. To get the idea of segmentation results of different scripts, see Fig. 7, where text and symbols are extracted from the color images.

5 Conclusion

In this paper we proposed a language and font insensitive system to retrieve text and symbol from color maps. Here, at first, we separated the maps into different layers according to color features and then in each layer, connected component features, skeleton information, geometrical features are used to identify text and symbol from graphics. We tested our method on documents of different languages like English, Spanish, Russian, Indian etc. and from the experiment we obtained encouraging results. In future we plan to test our system on more documents of different languages. Also, we plan to use the contextual information to remove small non-text part that are included into isolated and joined character group, mistaken by our approach.

Acknowledgement

This work has been partially supported by the Spanish projects TIN2006-15694-C02-02 and CONSOLIDER-INGENIO 2010 (CSD2007-00018).

References

1. Luo, H.Z., Agam, G., Dinstein, I.: Directional mathematical morphology approach for line thinning and extraction of character strings from maps and line drawings. In: ICDAR 1995, Washington, DC, USA, vol. 1, p. 257 (1995)
2. Tombre, K., Tabbone, S., Peissier, L., Lamiroy, B., Dosch, P.: Text /graphics separation revisited. In: Lopresti, D.P., Hu, J., Kashi, R.S. (eds.) DAS 2002. LNCS, vol. 2423, pp. 200–211. Springer, Heidelberg (2002)
3. Cao, R., Tan, C.L.: Text/graphics separation in maps. In: Proceedings of 4th IAPR International Workshop on Graphics Recognition, Kingston, Ontario, Canada, September 2001, pp. 44–48 (2001)
4. Vazquez, E., Baldrich, R., Vazquez, J., Vanrell, M.: Topological histogram reduction towards colour segmentation. In: Lecture Notes in Computer Science - Pattern Recognition and Image Analysis, pp. 55–62 (2007)
5. López, A.M., Lloret, D., Serrat, J., Villanueva, J.J.: Multilocal Creaseness Based on the Level-Set Extrinsic Curvature. Computer Vision and Image Understanding: CVIU 77(2), 111–144 (2000)

6. Fletcher, L.A., Kasturi, R.: A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images. IEEE Transactions on PAMI 10(6), 910–918 (1998)
7. Tan, C.L., Ng, P.O.: Text extraction using pyramid. Pattern Recognition 31(1), 63–72 (1998)
8. Roy, P.P.: An Approach to Text / Graphics Separation from Color Maps. M.S. thesis, CVC, UAB, Barcelona (February 2007)
9. Ahmed, M., Ward, R.: A Rotation Invariant Rule-Based Thinning Algorithm for Character Recognition. IEEE Transactions on PAMI 24(12), 1672–1678 (2002)
10. Dhar, D.B., Chanda, B.: Extraction and recognition of geographical features from paper maps. IJDA 8(4), 232–245 (2006)
11. Lopez, A.M., Villanueva, J.J., Lumbreras, F., Serrat, J.: Evaluation of methods for ridge and valley detection. IEEE Transactions on PAMI 21(4), 327–334 (1999)
12. Shafarenko, L., Petrou, M., Kittler, J.: Automatic watershed segmentation of randomly textured color images. IEEE Transactions on Image Processing 6(11), 1530–1544 (1997)
13. Deng, Y., Manjunath, B.S., Shin, H.: Color image segmentation. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 1999, vol. 2, pp. 446–451 (1999)
14. Comaniciu, D., Meer, P.: Robust Analysis of Feature Spaces: Color Image Segmentation. In: IEEE Conf. on CVPR, pp. 750–755 (1997)
15. Comaniciu, D., Meer, P.: Mean Shift Analysis and Applications. In: ICCV 1999: Proceedings of the International Conference on Computer Vision, Washington, DC, USA, vol. 2, p. 1197 (1999)
16. Coleman, G.B., Andrews, H.C.: Image segmentation by clustering. Proceedings of IEEE 67, 773–785 (1979)
17. Hase, H., Yoneda, M., Tokai, S., Kato, J., Suen, Y.: Color segmentation for text extraction. IJDAR 6(4), 271–284 (2003)