International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-4 Number-2 Issue-15 June-2014

Combination of Morphological Operations with Structure based Partitioning and grouping for Text String detection from Natural Scenes

Vyankatesh V. Rampurkar[1], Gyankamal J. Chhajed[2]

Abstract

Text information in natural scene images serves as important clues for many image-based applications such as scene perceptive, content-based image retrieval, assistive direction-finding and automatic geocoding. Now days different approaches like countours based, Image binarization and enhancement based, Gradient based and colour reduction based techniques can be used for the text detection from natural scenes. In this paper the combination of morphological operations with structure based partitioning and grouping is used for efficient string detection from any natural scene images. A new framework based on image partition and connected components grouping is used to locate text regions embedded in those images. These described techniques are based on sibling's method i.e. adjacent character grouping method and Text line grouping method. Text line grouping method can locate text strings situated at arbitrary orientations. In adjacent character grouping detected string is represented by a rectangle while in text line grouping detected string is represented by fitted line. Described techniques can find text strings by using structure-based partition and grouping methods.

Keywords

Connected Components Analysis, Adjacent Character grouping, Image partition, Text line grouping, Text string detection, Color reduction technique.

1. Introduction

Text information in natural scene images serves as important clues for many image-based applications such as scene perceptive, content-based image retrieval, assistive direction-finding and automatic geocoding etc. However, locating text from a complex background with multiple colors is a challenging task. A lot of objects on which characters are written exist in our living environment [1]. We humans get much information from these texts.

It is expected that robots act in our living environment and support us in the future. If robots can read text on objects such as packages and signs, robots can get information from them, and they can use it in their activation and support for us. Owing to the progress of OCR, computers have been able to read text in images [4]. However, images have many non-character textures, and they make it difficult to read text by OCR. To cope with that problem, we need to extract character string regions from images. Natural scene images contain text information which is often required to be automatically recognized and processed. Localization of text and simplification of the background in images is the main objective of automatic text detection approaches [6]. Identifying location of text in complex and bad quality images is difficult process as variety of fonts, backgrounds, colors, sizes are used. It is easy to locate text in images which satisfy text and background contrast is high [5].

2. Related Work

Algorithm based on closed contour method and thickness of characters: One of the characteristics of common characters in real images is that most of them are capable of producing closed contour when edge extraction process is applied. So Tomohiro Nishino [3] takes an approach to detect closed contours from images. In this the assumption is that string is aligned in straight line in any direction where closed contours have regular arrangement. Assuming that characters included in a character string are aligned horizontally, string regions can be extracted by detecting horizontally aligned closed contours. Tomohiro Nishino explains how to detect the horizontally aligned closed contour. First, a circumscribed rectangle of a closed contour is calculated. Next, the rectangle is slid to right by some pixels as much as the width of the rectangle. If the rectangle includes the center of a circumscribed

Manuscript received on May 15, 2014.

V.V.Rampurkar is currently Assistant Professor in Information Technology department at VPCOE Baramati Pune University, Maharashtra, India.

Gyankamal is currently Assistant Professor and Head in computer engineering at VPCOE Baramati, Pune University, Maharashtra, India.

rectangle of another closed contour, these two closed contours are assumed to be aligned horizontally and to be included in the same character string. Closed contours which are isolated are assumed not to be characters. Circumscribed rectangles of each character string are assumed to be string regions. By this process, string regions of horizontally aligned closed contours are extracted. Both closed and unclosed contours which lie left or right of character string regions are extracted, and each thickness is calculated. These contours are added to the character string region if they have the similar thickness to that of characters in that region.

Algorithm based on image binarization and connected component analysis: Basilios Gatos [2] produces gray level image and inverted gray level image. Then, calculate the two corresponding binary images using an adaptive binarization and image enhancement technique. In the sequel, the proposed technique involves a decision function that indicates which image between binary images contains text information [7].

Algorithm based on the application of a color reduction technique, and a method for edge detection: In this, an algorithm that automatically locates horizontally aligned text more efficiently in images and digital video is discussed. This approach uses edge detection, color reduction technique along with projection profile analysis and geometrical properties.

Step 1: Image Preprocessing- If the image data is not represented in YUV color space; it is converted to this color space by means of an appropriate transformation. After that, luminance value thresholding is applied to spread luminance values throughout the image and increase the contrast between the possibly interesting regions and the rest of the image.

Step 2: Edge Detection- This step focuses the attention to areas where text may occur. Julinda Gllavata employs a simple method for converting the gray-level image into an edge image. As a result, all character pixels as well as some non-character pixels which also show high local color contrast are registered in the edge image. In this image, the value of each pixel of the original image is replaced by the largest difference between itself and its neighbors (in horizontal, vertical and diagonal direction). Despite its simplicity, this procedure is highly effective. Finally, the contrast between edges will be increased by means of a convolution with an appropriate mask. Step 3: Detection of Text Regions- The horizontal projection profile of the edge image is analyzed in order to locate potential text areas. Since text regions show high contrast values, it is expected that they produce high peaks in horizontal projection.

Step 4: Enhancement and Segmentation of Text Regions- First, geometric properties of the text characters like the possible height, width, and width to height ratio are used to discard those regions whose geometric features do not fall into the predefined ranges of values. All remaining text candidates undergo another treatment in order to generate the so called text image where detected text appears on a simplified background. The binary edge image is generated from the edge image, erasing all pixels outside the pre-defined text boxes and then binarizing it. After this the task of gap filling is done if contrast color pixel is sandwiched in any direction like horizontally , vertically or diagonally and finally same color is given to sandwiched pixel . The gap image is used as a reference image to refine the localization of the detected text candidates. Text segmentation is the next step to take place. Then, the segmentation process concludes with a procedure which enhances text to background contrast on the text image.

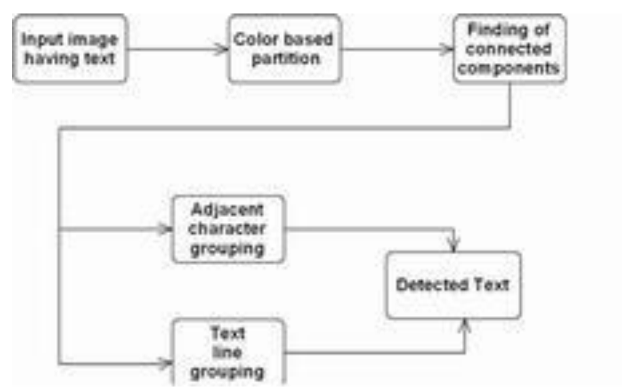

Figure 1: The flowchart of the framework

The proposed framework consists of two main steps, given here.

Step 1): Image partition to find text character candidates based on gradient feature and color uniformity. In this step, two methods are proposed to partition scene images into binary maps of nonoverlapped connected components: gradient-based method and color-based method. After this removing of connected components which are not text character checking by size, aspect ratio or inner holes is done as post processing. Step 2): Character candidate grouping to detect text strings based on joint structural features of text characters in each text string such as character sizes, distances between two neighboring characters, and character alignment. In this step, two methods of structural analysis of text strings are proposed: adjacent character grouping method and text line grouping method.

3. System Implementation

Introduction:

Figure 1 shows system architecture of Text string detection system from natural scenes. System works in four modules: Thresholding and basic morphological operations, finding of connected components, Adjacent character grouping and Text line grouping. First input image is converted into binary image. Then some morphological operations are applied on it. Then connected components are found out. After finding connected components height, width, centroid and area of each connected component are calculated. These parameters are required for the implementation of next modules. Then adjacent character grouping and text line grouping methods are implemented. Adjacent character grouping and Text line Groupings are methods that can detect text string from natural scene images.

Figure 2: GUI created for implementation of the system

Thresholding:

Thresholding is the simplest method of image segmentation. From a greyscale image, thresholding can be used to create binary images. Following is an example of Thresholding image of a given color image.

Figure 3: Thresholding of given color image

Dilation and Erosion:

The most basic morphological operations are dilation and Erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. Following is an example of erosion and dilation when applied to above binary image.

Figure 4: an example of erosion

Figure 5: an example of dilation

Connected Components Labeling:
Connected-component labeling is an algorithmic application of graph theory, where subsets of connected components are uniquely labeled based on a given heuristic. Connected-component labeling is used in computer vision to detect connected regions in binary digital images, although color images and data with higher dimensionality can also be processed.

Two-pass Algorithm for getting connected components labeling:
The two-pass algorithm iterates through 2-dimensional, binary data. The algorithm makes two passes over the image: one pass to record equivalences and assign temporary labels and the second to replace each temporary label by the label of equivalence class. A faster-scanning algorithm for connected-region extraction is presented below

On the first pass:
1 Iterate through each element of the data by column, then by row (Raster Scanning)
2 If the element is not the background

1. Get the neighbouring elements of the current element
2. If there are no neighbours, uniquely label the current element and continue
3. Otherwise, find the neighbor with the smallest label and assign it to the current element
4. Store the equivalence between neighboring labels

On the second pass:
1. Iterate through each element of the data by column, then by row
2. If the element is not the background

1. Relabel the element with the lowest equivalent label

Adjacent Character Grouping:
Text strings in natural scene images usually appear in alignment, namely, each text character in a text string must possess character siblings at adjacent positions. The structure features among sibling characters can be used to determine whether the connected components belong to text characters or unexpected noises. Here, following five constraints are defined to decide whether two connected components are siblings of each other.

1. Considering the capital and lowercase characters, the height ratio falls between T1 and 1/T1.
2. Two adjacent characters should not be too far from each other despite the variations of width, so the distance between two connected components should not be greater than T2 times the width of the wider one.
3. For text strings aligned approximately horizontally, the difference between Y-coordinates of the connected component centroids should not be greater than T3 times the height of the higher one.
4. Two adjacent characters usually appear in the same font size, thus their area ratio should be greater than 1/T4 and less than T4.
5. If the connected components are obtained from gradient based partition, the color difference between them should be lower than a predefined threshold T5 because the characters in the same string have similar colors.

In our system we set T1=T4=2, T2=3, T3=0.5 and T5=40
When C and C' are grouped together, their sibling sets will be updated according to their relative locations. That is, when C is located on the left of C'•, C' •will be added into the right-sibling set of C, which is simultaneously added into the left-sibling set of C'• the reverse operation will be applied when C is

located on the right of C'•. To create sibling groups corresponding to complete text strings, here we merge together any two sibling groups SG (C1) and SG (C2) when their intersection contains no less than two connected components. At this point, each sibling group can be considered as a fragment of a text string. Repeat the merge process until no sibling groups can be merged together. Text string in scene images can be described by corresponding adjacent character groups. To extract a region containing a text string, we calculate rectangle covering all of the connected components in the corresponding adjacent character group.

Text Line Grouping:

In order to locate text strings with arbitrary orientations, here developed text line grouping method. Centroid as the descriptor of each connected component is used for grouping connected components which may not be in straight line horizontally or vertically. Given a set of connected component centroids, groups of collinear character centroids are computed, as shown below.

M= {m1, m2, m3...} Where m is centroid of connected component, M denotes centroids of all connected components obtained from image partition

L= {G1, G2, G3...} where G is a subset of M and | G |>=3, these are character centroids and collinear and L denotes the set of lines joining text character centroid. Here designed an efficient algorithm to extract regions containing text strings. At first, we remove the centroids from the set M if areas of their corresponding connected components are smaller than the predefined threshold Ts. Then, three points mi, mj, mk are randomly selected from the set to form two line segments. Then we calculate the length difference, and incline angle difference between line segments (mi, mj) and (mj , mk) as shown

$\Delta d = D(mi, mj) / D(mj, mk)$

$\Delta\Theta = |\Theta ij - \Theta jk|, \quad \text{if } |\Theta ij - \Theta jk| <= \Pi/2$

$= |\Theta ij - \Pi - \Theta jk|, \text{ if } |\Theta ij - \Theta jk| > \Pi/2$

The three centroids are approximately collinear if $1/T6 <= \Delta d <= T6$ and $\Delta\Theta <= T7$ $T6=2$, $T7=\Pi/12$ Thus, they compose a preliminary fitted line lu = {mi, mj, mk}, u =index of fitted line.

Other collinear centroids along lu can be added into the end positions to form a complete text string increasingly. For now, each text string is described by a fitted line. The location and size of the region containing a text string is defined by the connected components whose centroids are cascaded in the corresponding fitted line.

Important Functions Used:

1. getImage(): getImage() function saves a bit image of specified region into memory, region can be any rectangle.
2. getWidth(): This method returns the width of displaying images in pixels.
3. getHeight(): This method returns the height of displaying images in pixels.
4. getRGB(x,y): The x and y coordinates represent the pixel location, x going from left to right and y going from top to bottom (both starting at 0). So (0, 0) is the upper left pixel, and (width-1, height-1) is the bottom right pixel. The integer value returned has the R, G, B and (potentially) Alpha values packed into a single int.
5. acos():The method returns the arccosine of the specified double value.
6. drawLine(int x1, int y1, int x2, int y2): Returns DrawLine for line starting at (x1, y1) and ending at (x2, y2).
7. drawRect(int x, int y, int width, int height): Draws the outline of the specified rectangle.
8. getRed(): getRed method gets the red component Of this Color object.
9. getGreen(): getGreen method gets the green component of this Color object.
10. getBlue(): getBlue method gets the blue component Of this Color object.

4. Result Analysis

Result Snapshots:

Upper part of the following figure 6 shows the input image to the system and lower part of the same shows detected string by the Adjacent character grouping method in which detected string is represented by cyan color within rectangle. Upper part of the following figure 7 shows the input image to the system and lower part of the same shows detected string by the Adjacent character grouping method in which detected string is represented by cyan color within rectangle. Upper part of the following figure 8 shows the input image to the system and lower part of the same shows detected string by the Adjacent character grouping method in which detected string is represented by cyan color within rectangle.

Upper part of the following figure 9 shows the input image to the system and lower part of the same shows detected string by the Adjacent character grouping method in which detected string is represented by cyan color within rectangle. Left part of the following figure 10 shows the input image to the system and right part of the same shows detected string by the Text line grouping method in which detected string is represented by fitted line. Upper part of the following figure 11 shows the input image to the system and lower part of the same shows detected string by the Adjacent character grouping method in which detected string is represented by cyan color within rectangle. Left part of the following figure 12 shows the input image to the system and Right part of the same shows detected string by the Adjacent character grouping method in which detected string is represented by cyan color within rectangle. Left part of the following figure 13 shows the input image where Chucai Yi and Ying Li Tian failed to detect the text string but here right part of the same shows detected string by the Adjacent character grouping method in which detected string is represented by cyan color within rectangle.

Performance measurement for following first five images is described next in table 1.

Here our results produce precision value up to 0.70

Figure 6: an image 1 and its equivalent output for adjacent character grouping

Analysis:

Adjacent character grouping is performed to combine the candidate text characters into text strings which contain at least three character members in alignment. In order to locate text strings with arbitrary orientations text line grouping method is developed. Adjacent character grouping is supported by the information of text orientations while text line grouping is performed for arbitrary text orientations, so its calculation cost is more expensive. In this method to group together the connected components that correspond to text characters in the same string which is probably non horizontal, we use centroid as the descriptor of each connected component. Here a solution is to search for satisfied centroid groups in the power set of M, so the complexity of this algorithm will be O $(2^{|M)}$ where |M represents the number of centroids in the set M. To evaluate the performance, here calculated two metrics, precision p and recall r. Here, precision is the ratio of area of the successfully extracted text regions to area of the whole detected region, i.e. precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. It is usually expressed as a percentage. Recall is the ratio of area of the successfully extracted text regions to area of the ground truth regions. i.e. recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is also usually expressed as a percentage. Recall and precision are inversely related.

Figure 7: an image 2 and its equivalent output for adjacent character grouping

Figure 8: an image 3 and its equivalent output for adjacent character grouping

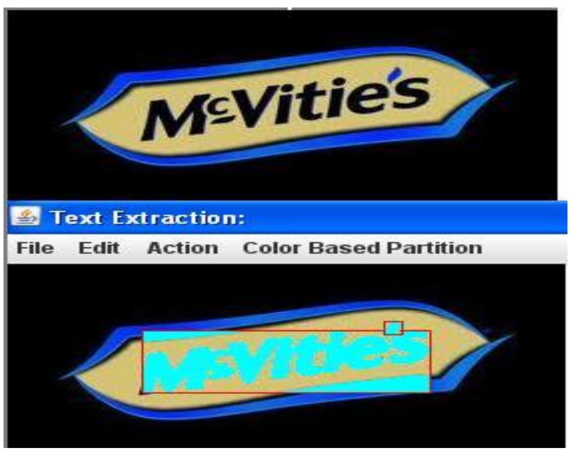

Figure 9: an image 4 and its equivalent output for adjacent character grouping

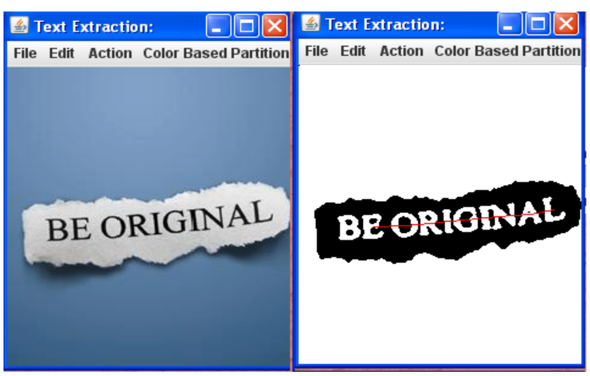

Figure 10: an image 5 and its equivalent output for Text Line grouping

Figure 11: an image 6 and its equivalent output for adjacent character grouping

Figure 12: an image 7 and its equivalent output for adjacent character grouping

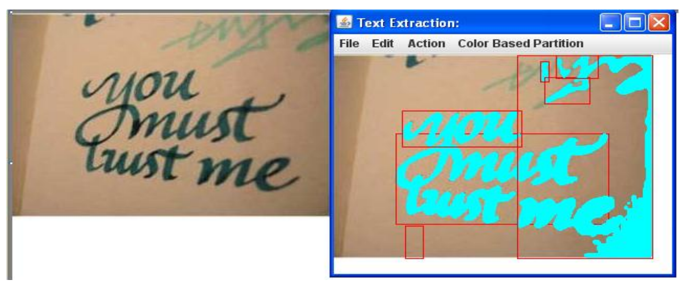

Figure 13: an image 8 and its equivalent output for adjacent character grouping

5. Conclusion

In this paper text detection is done from natural scene images. A new framework based on image partition and connected components grouping is used to locate text regions embedded in those images. To label a region of connected pixels with similar colors as a connected component, here used color-based partition method. Then by using 8-connectivity algorithm all connected components are found out. The adjacent character grouping method as string segments and then merges the intersecting sibling groups into text string. The text line grouping is able to extract text strings with arbitrary orientations. In adjacent character grouping detected string is represented by a rectangle while in text line grouping detected string is represented by fitted line. As described above obtained results is having precision and recall value in between 0.77 to 1. Some input images where Chucai Yi and YingLi Tian failed to detect the text string gives precision values up to 0.69 to 0.70 by proposed system as discussed above.

6. Future Scope

Here future work will focus on developing learning based methods for text extraction from complex backgrounds and text normalization for OCR recognition. First, we plan to employ an OCR system to check the recognition performance for the text images produced by the proposed algorithm. Second, the approach will be extended to also work with video sequences instead of still images. Finally, we plan to implement a hybrid system where connected component-based methods are combined with texture-based methods to possibly obtain further performance improvements.

References

[1] Chucai Yi and YingLi Tian, Senior Member, IEEE "Text String Detection from Natural Scenes by Structure-based Partition and Grouping", IEEE Trans Image Process. Sep 2011; 20(9): 2594–2605.

[2] "Extraction of Character String Regions from Scenery Images Based on Contours and Thickness of Characters" in MVA2011 IAPR Conference on Machine Vision Applications, June 13 15, 2011, Nara, JAPAN.

[3] J. Banerjee, A. M. Namboodiri, and C. V. Jawahar,"Contextual restoration of severely degraded document images," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2009.

[4] X. Chen and A. L. Yuille, Detecting and reading text in natural scenes, in Proc. IEEE Conf. Comput. Vis.Pattern Recognit., 2004, pp. 366 373.

[5] J. S. Kim, S. H. Kim, H. J. Yang, H. J. Son, and W. P. Kim, "Text extraction for spam-mail image filtering using a text color estimation technique", in Proc. 20th Int. Conf. Ind., Eng., Other Applic. Appl. Intell. Syst., 2007, pp. 105.

[6] P. K. Kim, "Automatic text location in complex color images using local color quantization", in Proc. IEEE TENCON, 1999, vol. 1, pp. 629 632.

[7] C. Mancas-Thillou and B. Gosselin, "Spatial and color spaces combination for natural scene text extraction", in Proc. IEEE Conf. Image Process., 2006, pp. 985 988.

Vyankatesh V. Rampurkar has received his Bachelor of Engineering in 2008 from VPCOE, BARAMATI. He has completed ME degree, in Computer Engineering at VPCOE, BARAMATI. Vyankatesh has participated and received certificate of participation in workshop on "Database Management Systems" and "Writing Effective Conference Papers" organized by MHRD and IIT, Bombay. He is life member of ISTE. Lecturer, B.E.(Computer),M.E.(Computer) Research Interest are Image processing applications for finding hidden data features, Data Mining and applications.

Gyankamal obtained Engineering degree (B.E.) in Computer Science and Engineering in the year 1991-95 from S.G.G.S.I.E.T, Nanded and Postgraduate degree (M.Tech.) in Computer Engineering from College of Engineering, Pune (COEP) in the year 2005-2007 . She is approved Undergraduate and Postgraduate teacher of Pune university and having about 16 yrs. of experience.
Gyankamal authored a book and has 10 publications at the national, international level for Conferences and Journal. She judged paper presentation event at National Conference , delivered expert talk and reviewed paper for IEEE international conference . She is life member of the ISTE & International Association IACSIT.