Uday Modha, Preeti Dave / International Journal of Engineering Research and Applications (IJERA)
ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 2,Mar-Apr 2012, pp.930-932

Image Inpainting-Automatic Detection and Removal of Text From Images

Uday Modha*, Preeti Dave**
*(Student of Master of Engineering in Information Technology, Shantilal Shah Engineering College, Bhavnagar, Gujarat, India)
** (Department of Electronics and Communication, Shantilal Shah Engineering College, Bhavnagar, Gujarat, India)

Abstract
Image Inpainting is the art of filling in missing data in an image. The purpose of inpainting is to reconstruct missing regions in a visually plausible manner so that it seems reasonable to the human eye. There have been several approaches proposed for the same. The novel contribution of the paper is the combination of the inpainting techniques with the techniques of finding text in images and a simple morphological algorithm that links them. This combination results in an automatic system for text removal and image restoration that requires no user interface at all. Examples on real images show very good performance of the proposed system and the importance of the new linking algorithm.

***Index Terms*—** inpainting, text detection, mathematical morphology.

1. INTRODUCTION
Inpainting is the process of filling-in missing or damaged image information. Its applications include the removal of scratches in a photograph [1], repairing damaged areas in ancient paintings, recovering lost blocks caused by wireless image transmission, image zooming and super-resolution [2],removing undesired objects from an image [3], or even perceptual filtering [4]. In the inpainting problem, the user has to select the area to be filled in, the inpainting region, since the areas missing or damaged in an image cannot be easily classified in an objective way. However, there are some occasions where this can be done. One such example, are the text characters printed in an image. Detecting and recognizing these characters can be very important in bi- modal search of internet data (image and text), and removing these is important in the context of removing indirect advertisements, and for aesthetic reasons.

In this paper we deal with the problem of *automatic* inpainting-based image restoration after text detection and removal from images. That is, given an image with text characters how to detect the exact position of the characters in the text, remove them and fill-in the resulting gaps via inpainting methods.

Our contribution is a new system for automatic text detection and removal from images, that requires no user interaction. This system is important because the selection of the area to be inpainted has been done manually by previous inpainting systems.

2. BACKGROUND AND REVIEW OF INPAINTING
For the inpainting problem it is essential to proceed to the discrimination between the structure and the texture of an image. As *structure* we can define the main parts - objects of an image, whose surface is homogeneous without having any details. As *texture* we can define the details on the surface of the objects which make the images more realistic.

3. OVERVIEW ON TEXT DETECTION
Much research work has been carried out in the area of text detection and localization in both, images and video. Therefore, for a better understanding of the different methods, the main character features are described. Although hardly ever all of them are taken into account in the same method, some of them, such as contrast or colour homo geneity, are always present. Text extraction from images and video sequences finds many useful applications in document processing [5], detection of vehicle license plate, analysis of technical papers with tables, maps, charts, and electric circuits [6], identification of parts in industrial automation [7],and content- based image/video retrieval from image/video databases [8], [9]. Educational and training video and TV programs such as news contain mixed text-picture-graphics regions.Region classification is helpful in object-based compression, manipulation and accessibility. Also,text regions may carry useful information about the visual content. However due to the variety of fonts, sizes,styles, orientations, alignment effects of uncontrolled illuminations, reflections, shadows,the distortion due to perspective projection as well as the complexity of image background, automatic localizing and extracting text is a challenging problem.Characters in a text are of different shapes and structures.

Text extraction may employ binarization [10], [5],[13] or directly process the original image [12], [13], [14]. In [9], a survey of existing techniques for page layout an analysis is presented.Mathematical morphology is a topological and geometrical based approach for image analysis. It provides powerful tools for extracting geometrical structures and representing shapes in many applications. Morphological feature extraction techniques have been efficiently applied to character recognition and document analysis, especially if dedicated hardware is used.In this paper, we propose an algorithm for text extraction based on morphological operations

4. SYSTEM FLOW

In this paper, we propose a text detection algorithm that consists of following steps:

1) Morphological edge detection
2) Text feature filtering
3) Text region binarization
4) Applying Inpainting Algoritmms

A. Step 1 - Morphological Edge Detection

To perform the edge detection algorithm, we first convert the input RGB color image to a gray-scale intensity image Y using (1), where R, G, and B represent red, green and blue components of the input image.

$$Y = 0.299R + 0.587G + 0.114B \quad (1)$$

The gray-scale image is then blurred using open-close and close-open filters to reduce false edge noise and over-segmentation. Structuring element used for this operation is a 3 by 3 8-connected element. Next, a morphological gradient detection operation is performed on the blurred image Ybl, as shown in (2).

$$Y2 = MG(Ybl,B) = dilation(Ybl)-erosion(Ybl) \quad (2)$$

In order to get the threshold level of Y2, we use a global nonhistogram-based thresholding technique. The threshold level is determined by (3), where s is an edge detector obtained by applying central difference edge detection filter to *Y2*. [1]

$$\gamma = \frac{\sum Y_2 \cdot s}{\sum s}$$

B. Step 2 - Text Feature Filtering

In order to reduce the number of connected components that have to be analyzed, a close operation with a 5 by 5 structuring element is performed to the binary edgeimage obtained from Step 1. After the close operation, all connected components of the edge image are Screened with their position, size,and area information. A candidate of letter should meet a set of constraints in size and shape. In our algorithm,we select connected components as letter candidates if the following requirements are met:

1) Width of the bounding box < 0.5 image width
2) Height of the bounding box < 0.3 image height
3) 0.1 < center width of the bounding box < 0.9
4) 0.3 < center height of the bounding box < 0.7
5) Width vs. height ratio < 10
6) Width of the bounding box > 10 pixels
7) 0.1 < Connected component filled area over (width height of the bounding box) < 0.95
8) Width of the bounding box > 10 pixels
9) 0.1 < Connected component filled area over (width height of the bounding box) < 0.95

After the first round filtering, it is expected that most of the non-letter components would be removed. So the majority of the remaining candidates should be letters with the same font and size. Based on this condition,we calculate the mean height hm of the bounding box of the remaining components, and remove any connected component with its height smaller than 0:6hm or greater than 1:8hm.

C. Step 3 - Text Region Binarization

Each remaining bounding box is used as a mask to the original gray-scale image. Otsu’s method [15] is used to obtain the threshold of the masked gray-scale image for binarization. Since each bounding box is relatively small compared to the size of the entire image, no further adaptive thresholding method is implemented. Theoretically after this step, only stroked letters are left as the foreground, 1, and the rest of the image would go to background, 0.

D. Step 4 – Applying Inpainting Algorithms

As a Final Step of our system,We now use these positions derived above as the inpainting area and apply any of the inpainting technic to remove the text from images.Result of step 3 is a binary image. Which works as a mask in inpainting technic.

Figure 1 Original Image

Figure 2 Text Detection

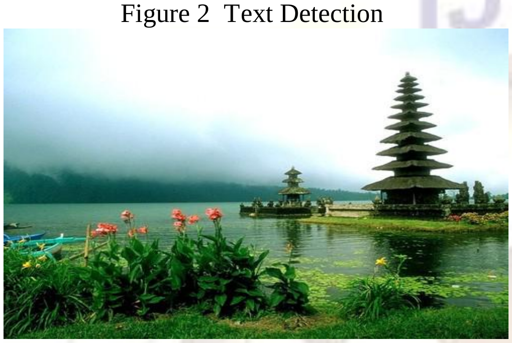

Figure 3 Inpainted Image

5. EXPERIMENTAL RESULTS

We present results of the system that we described above. It is important to note that, the system requires no user interaction,since both the text detection (inpainting region) and the inpainting process are being done automatically.Fig. 1 is the Original Image which is to be inpainted In Fig. 2 we used Morphological Text Extraction algorithm from [16] to detect the text position in the images. And works as a mask in inpainting algorithm.In Fig. 3 is the Final inpainted image after application of appropriate inpainting algorithms. In general, an automatic system is needed to decide about the appropriate inpainting method.

6. CONCLUSIONS

In this paper we dealt with the combined problems of inpainting and finding text in images. We have proposed a new simple morphological algorithm inspired from the reconstruction opening operation. This algorithm aptures all the pixels that correspond to text characters and thus its output can be considered as the inpainting region. By applying then an appropriate inpainting method, we have developed a system for automatic text removal and image inpainting.

7. REFERENCES

[1] M. Bertalmio, V. Casseles, C. Ballester, and G. Sapiro, Image Inpainting, ACM Press, 2000.

[2] T. Chan and J. Shen, “Mathematical models for local nontexture inpaintings,” SIAM J. on Appl. Math., vol. 62, pp. 1019–1043, 2002.

[3] A. Criminisi, P. Perez, and K. Toyama, “Object removal by exemplar based inpainting,” in Proc. IEEE-CVPR, 2003, vol. 2,pp. 721–728.

[4] M. Dimmiccoli and P. Salembier, “Perceptual filtering with connected operators and image inpainting,” in Proc. ISMM, 2007. [5] K. Jain and B. Yu, “Document representation and its application to page decomposition,” IEEE Trans. Pattern Anal. Machine Intell.,vol. 20, pp. 294–308, Mar. 1998.

[6] K. Jain, and B. Yu, “Automatic Text Location in Images and Video Frames”, Pattern Recognition, 31 (12) (1998) 2055-2076.

[7] K. Jain, and Y. Zhong, “Page Segmentation using Texture Analysis”, Pattern Recognition, 29 (5) (1996) 743-770.

[8] Chitrakala Gopalan and D. Manjula, “Contourlet Based Approach for Text Identification and Extraction from Heterogeneous Textual Images”, International Journal of Computer Science and Engineering 2(4) (2008) pp.202-211.

[9] D. Crandall, S. Antani, and R. Kasturi,“Robust Detection of Stylized Text Events in Digital Video”, Proceedings of International Conference on Document Analysis and Recognition, 2001, pp. 865-869.

[10] K. C. Fan, L. S. Wang, and Y. K. Wang, “Page segmentation and identification for intelligent signal processing,” Signal Process.,vol. 45, pp.329–346, 1995.

[11] H. M. Suen and J. F.Wang, “Text string extraction from images of colorpriented documents,” Proc. Inst. Elect. Eng. Vis., Image, Signal Process., vol. 143, no. 4, pp.210–216, 1996.

[12] R. Lienhart, “Indexing and retrieval of digital video sequences based on automatic text recognition,” in Proc. ACM Int. [13] L. Wang and T. Pavlidis, “Direct gray-scale extraction of features for character recognition,” IEEE Trans. Pattern Anal. Machine Intell., vol. 15, pp. 1053–1067, Oct. 1993.

[14] Y. Zhong, K. Karu, and A. K. Jain, “Locating text in complex color images,” Pattern Recognit., vol. 28, no. 10, pp. 1523–1535, 1995.

[15] Nobuyuki Otsu, A threshold selection method from gray-level histograms. IEEE Trans.Sys.,Man., Cyber 9(1):62-66

[16] Yassin M. Y. Hasan and Lina J. Karam 2000 “Morphological Text Extraction from Images” IEEE Transaction on Image Processing vol. 9,no. 11,Nov. 2000.