Colored Rubber Stamp Removal from Document Images

Soumyadeep Dey*, Jayanta Mukherjee, Shamik Sural, and Partha Bhowmick

Indian Institute of Technology, Kharagpur
{soumyadeepdey@sit,jay@cse,shamik@sit,pb@cse}.iitkgp.ernet.in

Abstract. Rubber stamps on document pages often overlap and obscure the text very badly, thereby impairing its readability and deteriorating the performance of an optical character recognition system. Removal of rubber stamps from a document image is, therefore, essential for successfully converting a document image into an editable electronic form. We propose here an effective technique for rubber stamp removal from scanned document images. It is based on the novel idea of a single feature obtained by projecting the pixel colors of the image foreground along the eigenvector corresponding to the first principal component in *HSV* color space. Otsu's adaptive thresholding is used to segment out the stamp impressions from the text by exploiting the discriminative power of the aforesaid feature. Experimentation and subjective evaluation on a variety of scanned document images demonstrate the strength and effectiveness of the proposed technique.

Keywords: Rubber stamp removal, document cleaning, colored document processing.

1 Introduction

Rubber stamps, also called seals, are used to cast distinctive and lasting impressions on document pages. Their purpose is to certify a document for various reasons, such as authorization, authentication of source, etc. The seal essentially comprises a suitably molded or engraved 'pattern'. A usual practice before pressing the seal against a document page is to smear its 'pattern' with a specially made ink or dye, so that the required impression is properly transferred to the concerned page.

While scanning the stamp-containing pages, the stamp impressions also get scanned along with the actual data content, which poses severe problem in converting a document into an electronic text form by an optical character recognition (OCR) system. A stamp may be present in a page either in some blank space or in overlap with a text segment. The latter case is more problematic, since the performance of an OCR system falls drastically in presence of text-overlapping stamp regions. Our work is particularly focused on removing the text-overlapping

* Corresponding author.

P. Maji et al. (Eds.): PReMI 2013, LNCS 8251, pp. 545–550, 2013.

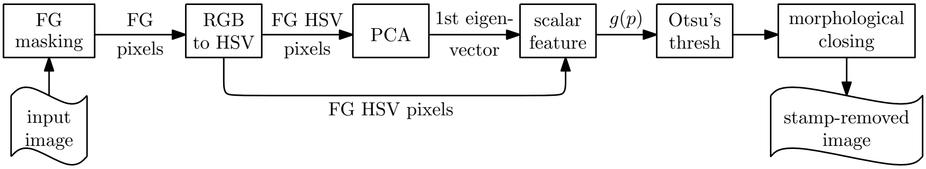


Fig. 1. Proposed stamp removal algorithm (FG = foreground)

stamp impressions from the concerned document image without affecting its text content. The OCR performance improves significantly once the stamp impressions are removed from the document image, as shown in this paper.

Till date, no work has been reported that addresses rubber stamp removal from document images, although there are some literature regarding rubber stamp detection in document images. In the stamp detection approach by Zhu and Doermann [7], stamps are limited to oval shape only. In another work, Zhu *et al.* [6] proposed an automatic logo detection algorithm, using a boosting strategy across multiple image scales. Forczmanski *et al.* [3] proposed a stamp detection algorithm based on color profile and shape analysis. All these methods are not appropriate for rubber stamp removal from document images, especially when the stamps do overlap with the text part. This has motivated us in designing an effective technique to remove rubber stamp impressions from document images. The work in this paper explicates this technique, which is based on principal component analysis in HSV space. One of its premises is that the text is written in a particular color of ink on a uniform background, and the stamps are of different color(s).

2 Methodology

The stages required for rubber stamp removal from document images are shown in Fig. 1. The input image is first classified into foreground and background. The foreground pixels are taken into HSV color space and PCA is performed on the converted foreground pixels. The foreground pixels in three-dimensional color space are then mapped onto an one-dimensional space, using the eigenvector corresponding to the first principal component. The obtained foreground pixels are classified into stamp and text pixels by usual thresholding technique after doing the histogram analysis of the resultant image.

2.1 Foreground Masking

An input color image I is first converted to a gray-scale image, I_g. The gray-scale image is binarized using an adaptive image binarization technique with window size 100 × 100 [2]. Binarization of the gray-scale image is obtained by Eq. 1, where $I_g(x, y)$ and $I_b(x, y)$ indicate the respective gray value and binary value at a pixel with coordinates (x, y) of the sub-image, and T represents the dynamically obtained threshold value for the sub-image.

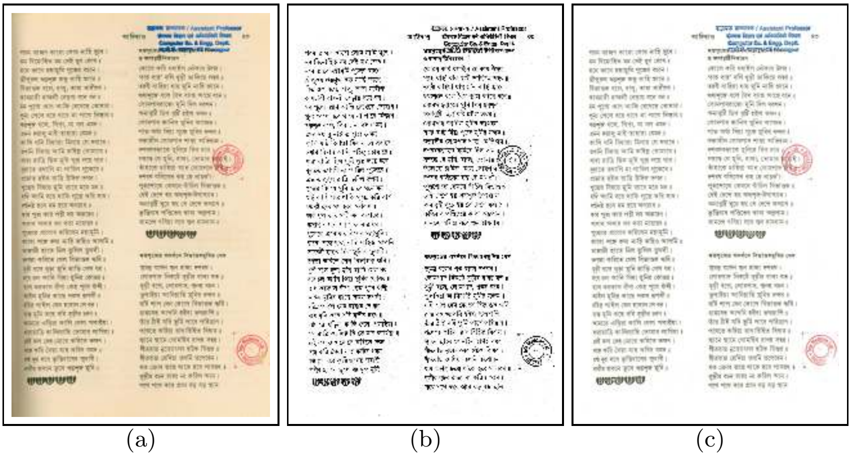


Fig. 2. (a) An input image; (b) binary image; (c) foreground masked image

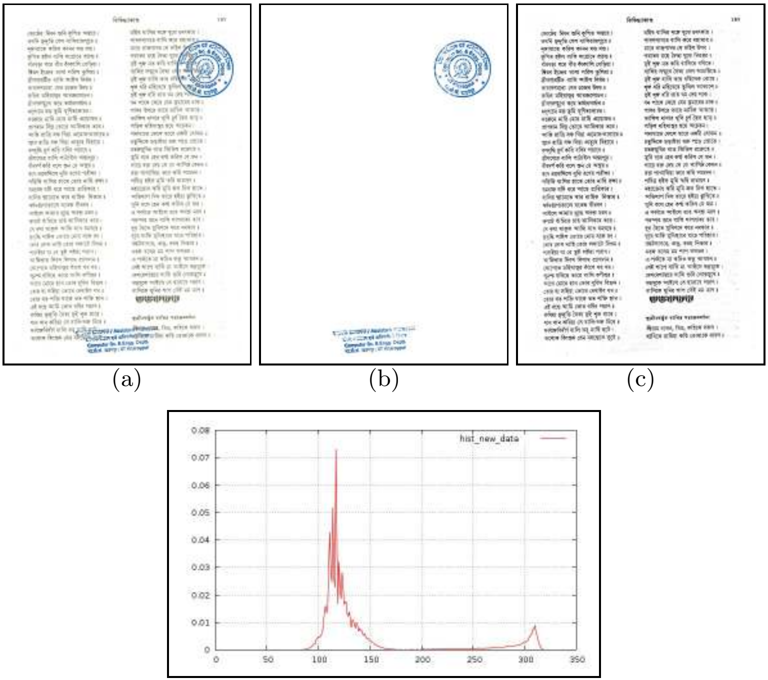
$$I_b(x,y) = \begin{cases} 255 & \text{if } I_g(x,y) > T \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Let F and B denote the respective sets of foreground pixels and of background pixels. The set F is used for masking the source color image on the binary image. This masking operation is used to get back the color information of the foreground pixels from the source image. The image, thus formed, is known as *foreground masked image*, has a uniform background and is used for further processing. An example of sample input image, its corresponding binary image, and foreground masked image are shown in Fig. 2.

2.2 Stamp Removal and Output Image Generation

The foreground masked image is an image in RGB color space. The foreground pixels of the foreground masked image are converted from RGB to HSV color space [4]. In HSV color space, a feature is selected for segmentation of stamp from the text part.

Principal component analysis (PCA) is a mathematical procedure to convert a set of correlated variables into a set of linearly uncorrelated variables called principal components, using an orthogonal transformation. Principal component analysis is carried out by computing the eigenvectors of the 3×3 covariance matrix in HSV space. We consider a feature based on the first principal component of the foreground data. Hence, the corresponding *unit eigenvector* is computed. Let the obtained unit eigenvector be $\overrightarrow{u} = a\hat{i} + b\hat{j} + c\hat{k}$. Let us denote the color vector at pixel p as $\overrightarrow{f(p)} = h(p)\hat{i} + s(p)\hat{j} + v(p)\hat{k}$, where $h(p)$, $s(p)$, and $v(p)$ denote the hue, saturation, and intensity values of p in HSV color space. Then the foreground three-dimensional data is converted into one-dimensional (scalar) data along the first principal component by projecting $\overrightarrow{f(p)}$ along $\overrightarrow{u}$ according to the following equation.



(a) (b) (c)

(d)

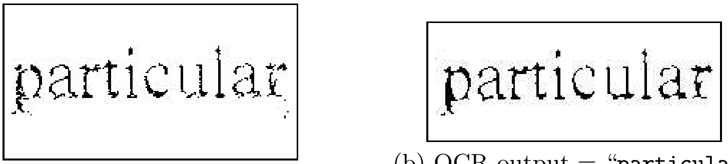
Fig. 3. (a) A foreground masked image; (b) stamp part of the image; (c) text part of the image after stamp removal; (d) histogram of foreground data ($g(p)$)

$$g(p) = \overrightarrow{f(p)} \cdot \overrightarrow{u} \tag{2}$$

For segmentation, the histogram of $g(p)$ is computed and analyzed as a bimodal histogram. Its highest peak corresponds to the text part, and the other peak to the stamp region(s). Otsu's thresholding algorithm [5] is used to classify the foreground pixels into two classes, one representing the text region and the other representing the stamp region.

Figure 3 shows a result produced by our algorithm. The foreground masked image is shown in Fig. 3(a). The histogram obtained using Eq. 2 on the foreground pixels is shown in Fig. 3(d). The segmented stamp regions and the text regions after segmenting the foreground pixels, using Otsu's thresholding method on the obtained histogram data, are shown in Fig. 3(b) and Fig. 3(c) respectively.

To join the characters, which got broken due to stamp removal from overlapped regions, morphological closing is performed by applying a dilation operation followed by an erosion [4]. The objective of this operation is to fill the holes in the text components without distorting their boundaries as much as possible. One example of a word with broken characters and its closed image obtained by using 3 × 3 morphological kernel, along with their corresponding OCR output, are shown in Fig. 4.



(a) OCR output = “`jparticulasz`”

(b) OCR output = “`particular`”

Fig. 4. (a) Binary image and its OCR output; (b) image after morphological closing and its OCR output

3 Results

For evaluation of the proposed rubber stamp removal algorithm, experimentation is conducted on different scanned pages. These pages are in different languages and contain rubber stamps of different shapes and colors. All the pages are scanned in 300 dpi resolution and given as input to the rubber stamp removal system. The input images and stamp-removed images are used for evaluation. The documents are read one at a time. The number of words readable before stamp removal and that after stamp removal are counted for the evaluation purpose. For documents written in English language, the OCR performance is reported before and after rubber stamp removal using a standard OCR system [1]. The OCR performance is not shown for Bengali documents, as its performance on our Bengali data set is very poor.

In Table 1, reading performances for stamp-overlapping text regions are shown, before and after stamp removal from the document images. Here, n_w refers to the number of words overlapping with the stamp regions, and Acc_{ws} and Acc_{sr} denote the respective accuracies for stamp-overlapping text regions before and after stamp removal. It may be noticed from this table that the reading accuracies improve significantly for both English and Bengali document images, once the stamp impressions are removed. The OCR performance also has an encouraging improvement for the English documents, as reflected in the corresponding word-level and character-level accuracies shown in Table 2. The accuracies of the OCR on English words and characters are estimated by manually counting the number of words and their associated characters which are overlapped with stamps. Some sample stamp-overlapping text regions and their outputs after stamp removal are shown in Fig. 5.

Table 1. Reading performance for English and Bengali documents

document	n_w	Acc_{ws}(%)	Acc_{sr}(%)
Bengali	264	28.41	98.11
English	221	39.82	99.09

Table 2. OCR performance on English documents

data type	number	Acc_{ws}(%)	Acc_{sr}(%)
words	221	1.81	84.62
characters	958	18.68	91.96

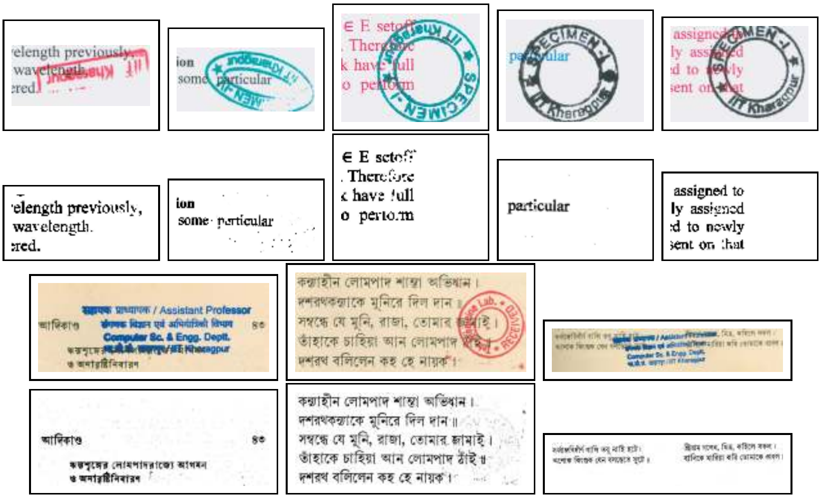


Fig. 5. Sample stamp removal results

4 Conclusion

We have proposed a stamp removal technique from document images, which significantly improves the document reading accuracy, as evidenced by relevant experimentation. The OCR performance for English documents in particular is found to be improved to a significant extent. The proposed technique can, therefore, be included in the document cleaning stage to improve the performance of an OCR system.

Acknowledgment. This work is funded by TCS research scholar program and Ministry of Communications & Information Technology, Government of India, Ref.: MCIT 11(19)/2010 HCC (TDIL) dt. 28-12-2010. We are thankful to our colleagues and lab members for preparing the data sets.

References

1. Free online OCR, `http://www.newocr.com/`
2. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
3. Forczmański, P., Frejlichowski, D.: Robust stamps detection and classification by means of general shape analysis. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) ICCVG 2010, Part I. LNCS, vol. 6374, pp. 360–367. Springer, Heidelberg (2010)
4. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. PHI (2009)
5. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. SMC 9(1), 62–66 (1979)
6. Zhu, G., Doermann, D.: Automatic document logo detection. In: Proc. ICDAR 2007, pp. 864–868 (2007)
7. Zhu, G., Jaeger, S., Doermann, D.: A robust stamp detection framework on degraded documents. In: SPIE Conf. Doc. Recog. & Retrieval, pp. 1–9 (2006)