Enhancement of Monochrome Text Quality in Color Copies

Nathir A. Rawashdeh, Mohamed N. Ahmed, Lexmark International, Inc., Lexington, KY, USA

Abstract

One aspect of copy image quality is the sharpness and color of copied text. Printing monochrome or black text as a color object will cause the text to appear too light, blurred, or having objectionable color hues. In this paper, we present a method to improve the appearance of black and monochrome text when copied on a color multi function printer (MFP), by detecting the presence of monochrome text characters in documents containing monochrome and color content. Initially, the scanned image is segmented into text, image, and background classes. This step is based on the analysis of edge, texture, and geometrical pixel-level features extracted at different scales. Once detected, text regions are analyzed for their chromatic content in order to discern whether the text is colored or merely a color scan of black text. Based on this classification, pixels belonging to black text are sharpened appropriately and printed using only black toner rather than a mixture of cyan, magenta, yellow, and black

Introduction

The process of copying documents on a color multi function printer can introduce quality degrading artifacts such as blur, aliasing, moiré, color infidelity, spatial distortion, and color fringing. Image processing techniques can mitigate most of these problems; however, with varying success, depending on the type of image content. For example, sharpening may be appropriate for text and line art content, but not for images. The user can usually manually select the copy mode that best fits the type of document content at hand. In office environments, it is important to preserve text sharpness, darkness, and color. In a document with mixed content, containing text and images for example, it is not possible to reproduce all the content optimally. Image processing parameters must be tuned to a compromise that ensures the page as a whole is optimally processed.

This paper introduces algorithms for document segmentation and that detect monochrome text characters. This enables special processing of monochrome text with the aim of improving its reproduction quality. Figure 1 is a close-up image of a black text character copied twice – in color and monochrome modes. The color reproduction, shown in Figure 1 (b), of the black character in Figure 1 (a) illustrates how it was printed using a combination of process black (a combination of cyan, magenta, and yellow) and real black. This reproduction shows colored dots outside the character boundary. At regular viewing distance, such text can appear less sharp, too light, and of an objectionable hue. Conversely, when the text is copied in monochrome mode, such as shown in Figure 1 (c), the characters are sharper, darker, and color neutral. We present a solution that enables the detection of monochrome text in the scanned document image, and printing the text using only black toner or ink. Several approaches for document segmentation have been proposed [1-8]. These techniques can be broadly categorized as bottom-up or top-down. Bottom-up methods start from the pixel level and merge regions together into larger and larger components. Top-down techniques apply a priori knowledge about the page to hypothesize and split the page into blocks which are subsequently identified and further subdivided. Top-down approaches work well with pre-specified layouts such as technical papers; however, the performance of these techniques degrades significantly when different components are touching or overlapping.

Among bottom-up approaches, texture-based schemes have attracted much attention [8-10]. These methods treat different components of a document image as different textures. The scanned document images are convolved with a set of masks to generate feature vectors. Each feature vector is then classified into different classes using a pre-trained classifier. One problem associated with these approaches is the mask size for extracting local features. If the mask size is too small, it is difficult to detect large scale textures such as large fonts. On the contrary, if a large mask is chosen, the computational complexity will increase dramatically [8].

The presented document segmentation technique is based on fuzzy c-means clustering [2]. To classify pixels as belonging to text or images, each pixel is assigned a feature vector, extracted from edge information and gray level distribution. The feature pattern is then assigned to a specific region using a fuzzy c-means approach. In the classical fuzzy c-means algorithm, the objective function is minimized when high membership values are assigned to pixels with feature vectors close to the centroid of their particular classes, and low membership values are assigned when the pixel data is far from the centroid of its class. This formulation, however, does not take into account the spatial relationships between pixels, which could lead to inconsistent classification.

(a) *(b)* *(c)*

Figure 1. *Digital images of a 10 point text character. (a) Original from a typical text book (b) printed color copy (c) printed monochrome copy.*

Inspired by Markov Random Field (MRF) image modeling [5], the presented algorithm is formulated by modifying the objective function of the standard FCM algorithm to allow the labeling of a pixel to be influenced by the labels in its immediate neighborhood. The new cost function conforms with MRF neighborhood modeling through the use of cliques. The objective function is minimized when the center pixel has a high membership value in a certain class at the same time that its related cliques possess low membership values in the other classes [2].

Feature Extraction

To distinguish text from images, we selected two features obtained from the gray level distribution in a window W of dimension $w \times w$, centered around each pixel.

$$\mu = \frac{1}{w^2} \sum_{(x,y)\in W} I(x,y) \quad , \tag{1}$$

and

$$\sigma^2 = \frac{1}{w^2 - 1} \sum_{(x,y)\in W} (I(x,y) - \mu)^2. \tag{2}$$

Standard FCM Objective Function

The standard FCM objective function for partitioning $\{x_k\}_{k=1}^N$ into c clusters is given by

$$J = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^p \| x_k - v_i \|^2 , \tag{3}$$

where $\{v_i\}_{i=1}^c$ are the prototypes of the clusters and $[u_{ik}] = \mathcal{U}$ is an array that represents a partition matrix $U \epsilon \mathcal{U}$, namely

$$\mathcal{U} \{ u_{ik} \in [0,1] \mid \sum_{i=1}^{c} u_{ik} = 1 \;\; \forall k \quad and \quad 0 < \sum_{k=1}^{N} u_{ik} < N \;\; \forall i \} \; . \tag{4}$$

The parameter p is a weighting exponent on each fuzzy membership and determines the amount of fuzziness of the resulting classification. The FCM objective function is minimized when high membership values are assigned to voxels whose intensities are close to the centroid of its particular class, and low membership values are assigned when the voxel data is far from the centroid. It is apparent from Eq. (3) that the FCM objective function does not take into consideration, any spatial dependence between observations. Thus, the computed membership functions can exhibit sensitivity to noise in the observed image. One obvious approach to compensate for this sensitivity is to smooth the image before applying FCM; however, standard smoothing filters can result in a loss of important image detail. More importantly, there is no way to rigorously control the trade-off between the smoothing and the clustering result that is obtained.

Spatial Constraint Term

We propose a modification to Eq. (3) by introducing a term that allows the labeling of a pixel to be influenced by the labels in its immediate neighborhood. As mentioned before, the neighborhood effect acts as a regularizer and biases the solution towards piecewise-homogeneous labeling. Such regularization is useful in segmenting scans corrupted by salt and pepper noise. The modified objective function is given by

$$J_m = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^p \| x_k - v_i \|^2 + \frac{\alpha}{N_R} \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^p \left(\sum_{m=1 m\neq i}^{c} \sum_{x_r \in N_k} \phi(\Delta I_{kr}) u_{mr}^p \right) , \tag{5}$$

where N_k stands for the set of neighbors that exist in a window around x_k and N_R is the cardinality of N_k and

$$\phi(\Delta I_{kr}) = e^{-(\Delta I_{kr})^2} \quad , \quad \Delta I_{kr} = | x_k - x_r | \; . \tag{6}$$

The effect of the neighbors term is controlled by the parameter α and $\phi(\cdot)$. The relative importance of the regularizing term is inversely proportional to the signal to noise ratio (SNR) of the image signal. Lower SNR would require a higher value of the parameter α . Also, If the edge strength is high, the value of $\phi(\cdot)$ is small, which prevents the inclusion of pixels across object boundaries. Formally, the optimization problem comes in the form

$$\min_{U, \{v_i\}_{i=1}^c} J_m \quad U \epsilon \mathcal{U} \quad subject\ to \quad U \epsilon \mathcal{U} \quad . \tag{7}$$

Parameter Estimation

The objective function J_m can be minimized in a fashion similar to the standard FCM algorithm. Taking the first derivatives of J_m with respect to u_{ik} , v_i , and setting them to zero results in two necessary but not sufficient conditions for J_m to be at a local extremum

$$u_{ik}^* = \frac{1}{\sum_{j=1}^{c} \left(\frac{D_{ik} + \frac{\alpha}{N_R}\gamma_i}{D_{jk} + \frac{\alpha}{N_R}\gamma_j} \right)^{\frac{1}{p-1}}} \tag{8}$$

and

$$v_i^* = \frac{\sum_{k=1}^{N} u_{ik}^p x_k}{(1+\alpha)\sum_{k=1}^{N} u_{ik}^p} \quad , \tag{9}$$

where $\gamma_i = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^p \left(\sum_{m=1 m\neq i}^{c} \sum_{x_r \in N_k} \phi(\Delta I_{kr}) u_{mr}^p \right)$, and

$$D_{ik} = \| x_k - x_r \|^2 \; .$$

Color Processing

Typically, when a page is scanned, a scan bar moves down the page while illuminating the portion of the page in front of the scan bar sensor. The scanner collects the RGB intensities of pixels in the scan band and starts processing them immediately, before the scan bar reaches the end of the page. This approach is cost effective because the scan bar sensor is a relatively expensive element and can be kept small. In addition, the small scan bar sensor minimizes the amount of data that must be processed per unit of time. This reduces memory requirements, and simplifies the processing hardware.

The multi-function printer used in this experiment processes the scan band pixels in the YCbCr opponent color space because it encodes color information more efficiently than the RGB space [11]. The following equation is used to convert scan band pixels from the RGB to the YCbCr color space:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2989 & 0.5866 & 0.1145 \\ -0.1687 & -0.3312 & 0.5000 \\ 0.5000 & -0.4183 & -0.0816 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}, \quad (10)$$

where *R*, *G*, *B*, *Y*, *Cb*, *Cr* are 8-bit values ranging from 0 to 255. The RGB values are not standardized and vary with the scanner model and hence the resulting YCbCr gamut is also scanner specific. The *Y* channel is equivalent to a grayscale (luma) image, and the color information is encoded in the *Cb* and *Cr* channels. For example, *Cb* and *Cr* values less than 128 indicate that the color is more yellow and green than blue and red. The neutral axis passes through the center of the plane, i.e., where *Cb* = 128 and *Cr* = 128. Equivalent to metric chroma in the CIE L*a*b* color space, we define colorfulness of a pixel *P* as the Euclidean distance ($\boldsymbol{\ell}_2$ norm) from the neutral center in the CbCr plane as follows:

$$C(p) = \sqrt{(Cb(p) - 128)^2 + (Cr(p) - 128)^2}\ , \quad (11)$$

where *Cb(p)* is the *Cb* component of pixel *P*, and *Cr(p)* is the *Cr* component. With this definition of colorfulness, it is possible to classify a pixel as colorful or not given an appropriate threshold that is based on example scans of monochrome and color documents.

Colorfulness Threshold Model

In order to classify a pixel as color or monochrome, we compute its colorfulness according to Eq.(11) and compare it to a threshold. A pixel is classified as color it if is more colorful than the threshold, and conversely, it is classified as monochrome if it is less colorful than the threshold. The choice of colorfulness threshold requires the consideration of variables such as, scanner technology, media type, and expected document content. Rather than use trial and error in tuning the threshold, we modeled the cumulative colorfulness histogram of 16 color and monochrome test pages that are representative of the type of documents expected to be processed on the device. The document contents included images, graphics, line art, text, solid patches, and color ramps. Figure 2 shows the resulting cumulative colorfulness histogram. It is bimodal with a high peak indicating the presence of many pixels that are minimally colorful. The smaller peak shows the presence of colorful pixels that are less in number and with a wider colorfulness distribution.

The optimal threshold should separate the peaks, i.e. classes, while minimizing the probability that a pixel is misclassified given its colorfulness value. This is achieved through modeling the cumulative colorfulness histogram using a mixture of Gaussian distributions *P(x)*, as follows:

$$P(x) = \sum_{i=1}^{c} \pi_i\, p(x|C_i)\ , \quad (12)$$

where $0 < x < 128$ is the pixel colorfulness bin center, c is the number of normal mixtures (2 in this case), π_i is the mixing proportion of the C_i mixture, and the distributions are:

$$p(x|C_i) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-(x-\mu_i)^2/2\sigma_i^2}\ , \quad (13)$$

where $\theta_i=\{\mu_i,\sigma_i\}$ are C_i parameters. In order to estimate the mixture parameters (means, variances, and mixing proportions), we employ the Expectation-Maximization (EM) algorithm [12-16].

The results of applying EM with a mixture of two Gaussians to the cumulative colorfulness histogram are shown in Figure 2. The cumulative colorfulness histogram data is shown in black. The peak between 0 and 10 is the contribution of all the less colorful, or monochrome, pixels in the test scans. The distribution of more colorful pixels lies between 10 and 50.

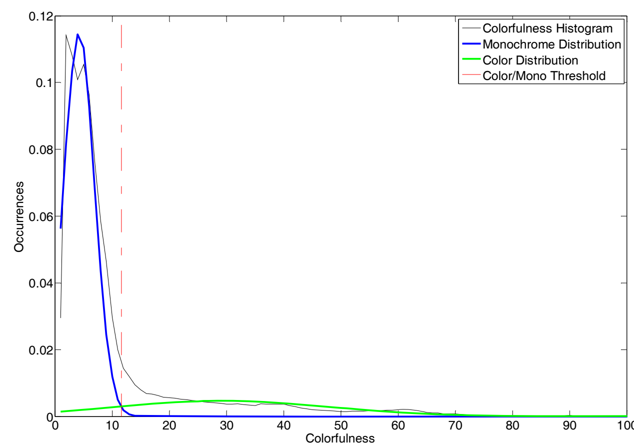

Figure 2*. Gaussian Mixture model of the cumulative colorfulness histogram.*

A mixture of two Gaussians was chosen to model the histogram data since there are two classes of colorfulness of a pixel – color and monochrome. Applying the EM algorithm with an initial guess of the distribution parameters yields the two Gaussians shown. The class of monochrome pixels (blue distribution) is centered at $\mu_1 = 4.24$, with a standard deviation of $\sigma_1 = 2.71$ and a $\pi_1 = 0.78$ mixing proportion. The class of color pixels (green distribution) has a mean of $\mu_2 = 29.35$, a standard deviation of $\sigma_2 = 18.85$, and a $\pi_2 = 0.22$ mixing proportion.

Given the Gaussian mixture model, it is possible to calculate the probability that a pixel is monochrome or color given its colorfulness following Eq.(11). For example, a pixel with colorfulness equal to 20 is more likely to be a color pixel than a monochromatic one – according to the Gaussian mixture model. For implementation in the scanner hardware, it is faster to simply use a threshold value for classification, such that a pixel is classified as color if its colorfulness is lager or equal to the threshold value and mono otherwise. The histogram model yields this optimal threshold as being equal to *11.68*. This is the value where the two distributions intersect, as indicated by the vertical dashed line in Figure 2.

After the page segmentation algorithm detects the locations of the text characters in the scanned document, it reports them as rectangular image regions that are bounding boxes around the text characters. These regions are then analyzed for colorfulness by counting the number of colorful pixels in the regions. A text character is deemed monochrome only if the percentage of colorful pixels in the bounding box region is smaller than a certain threshold. This approach is robust to color noise and provides a tunable threshold, possibly by the user, to set the MFP's sensitivity to the ratio of color pixels in the bounding box of text characters, which can vary with the type of font.

Results

We applied the algorithm to several test images using an MFP with a contact image sensor scan bar at 300 dpi. A scan of a test page is shown in Figure 3. It contains text of various font types, sizes and colors. In addition, there are fields of solid color, as well as graphic elements. This mixture of content is representative of documents found in office environments where good text quality is important. Due to the large amount of color and the variety of content, such a page is likely to be copied in color mixed mode. This means that the black text will be reproduced using a mixture of real and process black, as illustrated in Figure 1, which reduces its quality. As shown in Figure 4, the algorithm segments the page and marks the objects in the page using various label colors. Text characters are detected by the size of the bounding boxes. Bounding boxes that are larger than a threshold will not be classified as locations of text. The top portion of the page, for example, contains a large area with graphics and line art. The algorithm detected it and reported a large bounding box around it. Within it, there are graphics elements, text and lines. The algorithm successfully detects some of the elements. It does not detect the white text characters within the area, similarly in the horizontal band at the bottom of the page. The algorithm detected the band as an object, but it considers white text characters to be part of the background rather than objects that need to be printed. On the other hand, the algorithm successfully detected all the text characters located on the white background in the middle of, and the bottom of the page. With text of very small font, however, characters can appear connected in the scan image, and thus, are considered a single object and are bound collectively within a single bounding box. After the page segmentation results identify text characters, the algorithm proceeds to classify the text as either monochrome or color. A color text character is found when its bounding box contains a large proportion of colorful pixels. This threshold can vary with font type and can be made user tunable. As described earlier, a colorful pixel is defined as having a colorfulness value, according to Eq.(11) that is larger than the colorfulness threshold. The results of this step are illustrated in Figure 5. The enlarged region shown contains several lines of black text of varying font size and style, in addition to a line of red text – the second line from the top. The bounding boxes have been modified to to illustrate the detection of monochrome text (with black bounding boxes) and color text (with gray bounding boxes). Given this classification, it is possible for the MFP to selectively print black and gray text using only real black toner, while preserving the print settings of the other objects on the page.

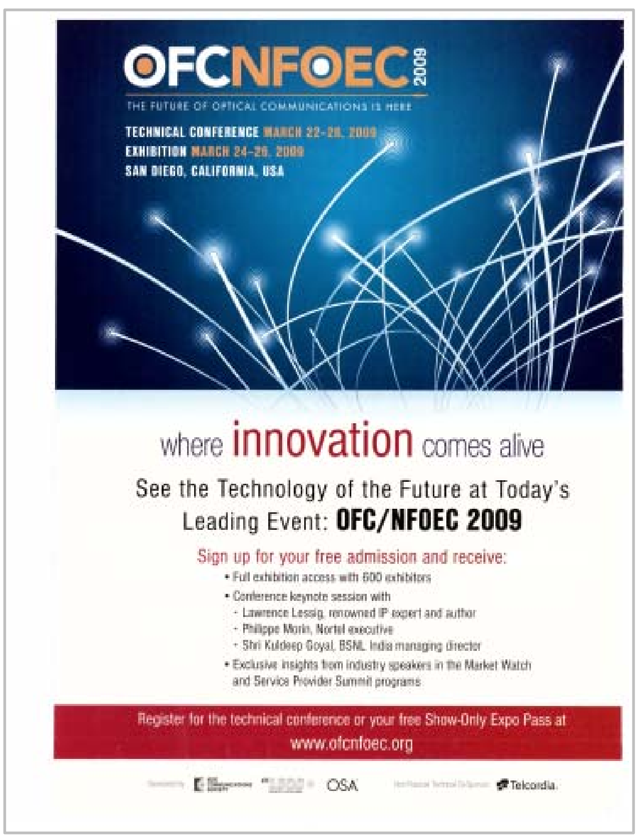

Figure 3. RGB scan of a color test page with mixed content.

Figure 4. Segmented page with bounding boxes on object.

Figure 5. *An area showing detected monochrome and color text characters.*

Conclusions

This paper presents an algorithm for the segmentation of document content into background, text and image pixels. A bounding box is placed around the detected objects. Furthermore, once text characters are detected, they are classified as either monochrome or color through analysis of the proportions of colorful pixels within their bounding boxes. The page segmentation employed a modification of the standard FCM algorithm, and the classification of pixels as color or monochrome is based on the modeling of colorfulness histograms of several test pages as a Mixture of two Gaussians that are found through the Expectation Maximization algorithm. The results show how the algorithm detects monochrome text characters in a page with mixed content, which enables improved copies of monochrome text of color MFP's.

References

[1] M. N. Ahmed, Ahmed Eid, Parameter Optimization for Content-based Image Enhancement, Proc. of the International Conference on Digital Printing Technologies, NIP 23, (2007).

[2] M. N. Ahmed, Brian E. Cooper, Segmentation and Enhancement of Digital Copies Using a New Fuzzy Clustering Method, Proc. of Electronic Imaging conference, (2006).

[3] J. Luo, R. Queiroz, Z. Fan, ''A Robust Technique for Image Descreening Based on the Wavelet Transform," IEEE Trans. on Signal Processing, Vol. 46, No. 4, (1998).

[4] S. Hasib, C. Bouman, ''Training-based Descreening," IEEE Trans. on Image Processing, Vol. 16, No. 3, (2007).

[5] S. Kumar, R. Gupta, N. Khanna, S. Chaudhury, S. Hoshi, ''Text Extraction and Document Image Segmentation Using Matched Wavelets and MRF Model," IEEE Trans. on Image Processing , Vol. 16, No. 8, pp. 2117-2128, (2007).

[6] C. Carson, S. Belongie, H. Greenspan, J. Malik, ''Blobworld: Image Segmentation Using Expectation- Maximization and its Application to Image Query," IEEE Trans. on Pattern Matching and Machine Intelligence (PAMI) , Vol. 24, No. 8, pp. 1026-1038, (2002).

[7] M. Ahmed, B. E. Cooper, S. Love, Document Image Segmentation Using a Two-stage Neural Network, Proc. of SPIE, Vol. 3962, pp. 25-33, (2000).

[8] H. Cheng, C. Bouman, J. Allebach, Multiscale Document Segmentation, Proc. of IS&T 50th Annual Conference, pp. 417-425, (1997).

[9] K. Etemad, D. Doermann, R. Chellapa, Page Segmentation Using Decision Integration and Wavelets Packets, Proc. Int. Conference of Pattern Recognition, Vol. 2, pp. 345-349, (1994).

[10] A. K. Jain, Y. Zhong, "Page Segmentation using texture analysis, " Pattern Recognition, Vol. 29, No. 5 pp. 343-770, (1996).

[11] Recommendation ITU-R BT.601-6, "Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide Screen 16:9 Aspect Ratios," International Telecommunications Union, (1982, 1986, 1990, 1992, 1994, 1995, 2007).

[12] N. A. Rawashdeh, M. N. Ahmed, Detection of Monochrome Pages in Color Document Scans, Proceedings of the 25'th International Conference on Digital Printing Technologies, NIP25, (2009) .

[13] J. C. Handley, Scanned Color Document Image Segmentation Using the EM Algorithm, International Congress of Imaging Sciences, pp. 675-678, (2006).

[14] Geoffrey J. McLachlan and Thriyambakam Krishnan, The EM Algorithm and Extensions, 2nd Edition, (Wiley, NY 2008).

[15] A. P. Dempster, N. M. Laird, D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm". Journal of the Royal Statistical Society. Series B, Vol. 39, No. 1, pp. 1–38, (1977).

[16] C. Reddy, H. Chiang, B. Rajaratnam, ''TRUST-TECH-Based Expectation Maximization for Learning Finite Mixture Models," IEEE Trans. on Pattern Matching and Machine Intelligence (PAMI), Vol. 30, No. 7, pp. 1146-1157, (2008).

Author Biography

Nathir A. Rawashdeh received the Ph.D. and B.S. degrees in electrical engineering from the University of Kentucky in 2007 and 2000 respectively. He received the M.S. in electrical and computer engineering from the University of Massachusetts, Amherst in 2003. Currently, he is a Senior Software Engineer in the Color Science and Imaging Department at Lexmark International. He was with The MathWorks between 2000 and 2003, where he worked on DSP-target tools. His research interests include image quality analysis, color science, and pattern recognition.
Dr. Rawashdeh is a member of the IEEE, ARRL and SPIE.