Networked Document Imaging with Normalization and Optimization

Hirobumi Nishida

Document Lab, Software R&D Group, Ricoh Co., Ltd.,
1-1-17 Koishikawa, Bunkyo-ku, Tokyo 112-0002, Japan
`hn@src.ricoh.co.jp`

Abstract. A system architecture is presented for document imaging in an open, distributed environment over networks, where various kinds of imaging devices can be interconnected remotely. The key components are two sets of image processing operations to transform input images to (1) canonical image representations to absorb different visual appearance due to characteristics of imaging devices or image acquisition conditions (normalization), and (2) optimal image representations according to tasks and preferences of individual users (optimization). Images captured through a diversity of input devices can be delivered to remote sites through networks, and then will be used for a variety of tasks such as printing on paper sheets, browsing on displays, and editing. These diversities can be resolved systematically by placing the normalizations at an upper end (routing servers) and the optimizations at a lower end (clients) of the data flow over networks. In view of this architecture, we describe some instances of the normalizations and optimizations associated with a particular task of highly legible printing of scanned document images. Three essential algorithms are mentioned for optimizing document images: adaptive tone mapping with background cleaning, text super-resolution, and text color clustering. The optimization process is mentioned for highly legible printing, along with some other potential applications and tasks.

1 Introduction

Various kinds of imaging devices can be interconnected remotely in open, distributed environments over networks. Images captured through a diversity of input devices can be delivered to remote sites through networks, and then the stored images will be retrieved for *repurposing* such as printing on paper sheets, browsing on PC, PDA, or cellular phones, and editing, as shown in Fig. 1. This new environment can also be referred to as *asynchronous* in the following sense:

- The sender does not know how, why, or when the delivered image will be used, or the characteristics of output devices at the recipient site (resolution, display area, color gamut, etc.).
- The recipient of delivered images would not know the identity or characteristics (MTF and color) of the input device or the image acquisition conditions at a remote site over the network.
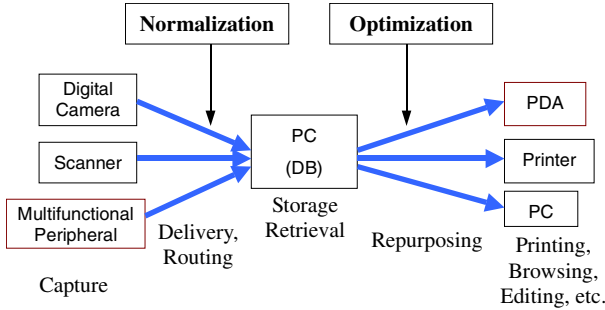
Fig. 1. Document imaging in open, distributed environments

These facts impose new technical challenges on document imaging. Taking an example task of printing out a scanned document image delivered through networks, we often observe the following phenomena and problems in terms of image quality:

- Contrast between text and background is poor.
- Background is colored on the printed image, while it should be white ideally.
- Background needs to be cleaned up to remove "show-through" or noise.
- Moiré phenomena are observed.
- Legibility of small text is poor because of insufficient image resolution.

To resolve these problems, we need to apply some image enhancement suitable for document images. Traditional methods for text image enhancement can be classified into four categories: filtering, contrast enhancement, model-based image restoration, and resolution expansion. Filtering approaches include noise removal using mathematical morphology [6] and second order filters for noise removal without blurring details [11]. Nonlinear tone mapping based on local window statistics [12] is an instance of contrast enhancement. A model-based method of image restoration is proposed based on cluster analysis of causes of OCR errors [5]. Resolution expansion approaches include text bitmap clustering and averaging for generating outlines in arbitrary resolution [2], restoration of optimal high resolution images in terms of an inverse problem based on an objective function composed of bimodality of distribution, smoothness, and intensity [14], and combinations of interpolation and binarization [7]. By combining these techniques, image defects can be resolved and document images can be transformed through various image-processing operations into optimal representations for various tasks (display, print out, OCR, etc.) or preferences of individual users.

There are also new representations and uses of document images in addition to traditional binarized images for OCR and raster images for printing. For instance, the Mixed Raster Content (MRC) Imaging Model, decomposing the document image into foreground (high resolution to be preserved) and background (resolution to be reduced) along with foreground colors (color palettes for text), is now used as a standard (JPM in JPEG2000) for scanned document images [3,4]. The reflowing technique [1] rearranges and reconstructs on Web browsers the document components (text, pictures, etc.) extracted from the image so that the text can be most readable on the specific display.

As can be seen, there could be a diversity of technologies involved in networked document imaging. It is now time to explore system architectures so that we can cope with this diversity in a systematic and extendable way. A key observation is that the diversity can be classified into two factors:

Normalization: transforms the input images into *canonical representations* absorbing specific differences due to the characteristics of input devices and the image acquisition conditions.

Optimization: transforms the normalized input images into *optimal representations* according to *preferences and tasks* of individual users.

The notions of normalization and optimization are quite general. In this paper, as a case study, we focus on the task of highly legible printing of scanned document images. This task of "scan-to-print" is the most common in networked document imaging, because scanned document images are stored in repositories and will be printed by *some* users at a remotely located office many days (or years) later. We describe some instances of the normalization and optimization associated with this particular task.

This paper is organized as follows: In Section 2, we introduce the notions of normalization and optimization in networked document imaging. In Section 3, we outline some algorithms for optimizing scanned document images. In Section 4, the task of highly legible printing of scanned document images is illustrated based on the optimizations. Section 5 is the conclusion.

2   Normalization and Optimization

In this section, we introduce the notions of normalization and optimization in networked document imaging. We also mention image processing techniques for normalization to absorb differences of image appearance due to specific characteristics of individual scanners. Furthermore, some *objective functions* along with image processing algorithms are described for optimizing scanned document images.

Normalization transforms the input images into *canonical representations* absorbing specific differences due to the characteristics of input devices and the image acquisition conditions.

- The canonical representation should be objective and physically determined, because specific differences have been introduced due to artifacts associated with particular input devices.
- Normalized images should enjoy fair quality when output or displayed anywhere, because particular artifacts have been removed through the normalizations.
- No loss of information should be introduced through the normalizations, because only particular artifacts are to be removed.
- The normalizations should be free of parameters specified by end users, because recipients would not know the characteristics of the input device or the image acquisition conditions at a remote site over the network.

- The normalizations should be placed at an upper end (routing servers) of the data flow with batch modes (no user interactions), because the normalizations resolve individual differences of image appearance regardless of tasks or preferences of the recipients.

In contrast to normalization, *optimization* transforms the normalized input images into *optimal representations* according to *preferences and tasks* of individual users

- Preference and tasks are subjective and even psychological.
- Optimal representations also depend on characteristics (resolution, display area, color gamut, etc.) of image output devices.
- The optimizations should be adjustable easily with a few simple parameters specified by end users, because *objective functions* for a particular optimization differ according to tasks, preferences, or characteristics of image output devices.
- The optimizations should be placed at a lower end (clients) of the data flow with interactive modes.

Particular components for image processing and analysis depend on the types of input devices (for instance, flatbed scanners, digital cameras) and tasks in consideration. In this paper, we address the most common task associated with document images: highly legible *printing* of *scanne*d document images.

2.1 Normalization

Normalization transforms the input images into *canonical representations* absorbing specific differences due to the characteristics of input devices and the image acquisition conditions. As far as scanners are used as input devices, the following factors possibly introduce specific difference of image appearance:

- Diversities of input devices
  - ➢ Moiré phenomena and graininess of image appearance due to different frequency characteristics (MTF).
  - ➢ Different color characteristics.
- Diversities of image acquisition conditions
  - ➢ Skew
  - ➢ Spherical distortions (bound volumes)
  - ➢ Homographic (perspective) distortions (for camera-type scanners)
  - ➢ Non-uniform illumination

As solutions to image defects due to image acquisition conditions, *rectification* techniques have been investigated extensively in document image analysis. Furthermore, color management has recently been incorporated into many networked imaging devices to maintain color consistency among different devices.

Techniques for suppressing moiré and graininess are often referred to as *inverse halftonning* that restores ideal continuous-tone representations from halftone dot patterns (Fig. 2). Such processing is now included as options in most scanner drivers. However, in network environments, inverse halftoning needs to be applied to image data obtained through an unknown scanner located at a remote site. Therefore, *blind*
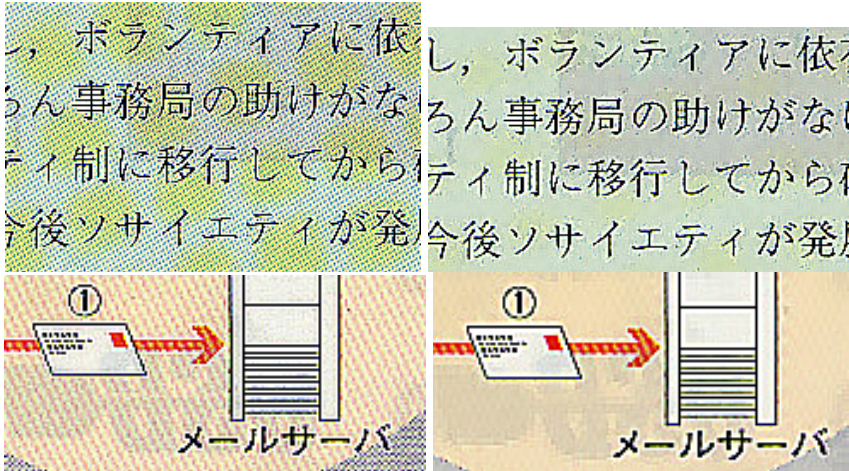
Fig. 2. Original images (left) and result of the inverse halftoning (right)

methods are required without depending on the type or properties of halftoning processes. Practical processing speed must be achieved under software implementation so that images can be processed on routing servers during the image delivery process. A blind, fast algorithm to meet these requirements can be found in Nishida [9].

2.2   Optimization

In general, an optimization problem consists of an objective function and its domain along with constraints. A document image is composed of different types of zones (domains), namely text, pictures, drawings, tables, and rulers, for which different objective functions can be considered in view of image quality and legibility. Typical constraints are the computational resources available. Ideally, computation for optimizations should take only a few seconds with users' interactions on client PCs.

The objective functions also depend the document imaging model. We consider optimization of scanned document images based on the Mixed Raster Content (MRC) imaging model that has attracted much attention with the recent adoption of JPEG2000/Part6 (ISO 15444-6) [4] as a new worldwide standard for encoding color document images.

The MRC (Mixed Raster Content) model is composed of separate background and foreground layers with separate compression methods, and an image mask to blend them together. The scanned document images are first segmented into the picture components and the text components, and the text is divided up again into the binary image and the color. The resulting three layers are compressed separately from each other. Lossless compression is applied to the text layer (a binary image), and therefore, high legibility and OCR accuracy can be achieved. The other two layers (color images) are compressed separately in JPEG2000/Part1 (the image compression within JPEG2000) [3].

Following the MRC model, we find that the following objective functions can be introduced for the three types of image layers:

- Background: The background color (paper color) should be uniform or ideally white. "Show-through" from backside also needs to be cleaned up.
- Binary images for foreground: Legibility of small text should be improved if the image is input in insufficient resolution.
- Color for foreground: Text colors should be clustered into a small number of pallets.

Furthermore, contrast should be enhanced over the entire image to improve text legibility.

As algorithms of image processing and analysis for optimization, the following can be considered on top of layout analysis for page segmentation:

- Adaptive tone mapping with background cleaning for contrast enhancement and background color optimization.
- Text super-resolution for improving legibility of small text.
- Color clustering for text color pallets.

In Section 4, we outline these three algorithms in consideration of computational resources, in particular, to compute image components for optimization within a reasonable time (a few seconds) on PCs.

3   Outline of Algorithms for Optimization

In this section, we outline image processing algorithms for optimizing the above three types of objective functions.

3.1   Adaptive Tone Mapping and Background Cleaning

We address the problems with contrast enhancement between text and background, adaptive control of the tone mapping according to image content, and background cleaning to remove show-through components and noise. A tone mapping is constructed based on statistics of text and background colors, and it is then modified so that black enhancement can be suppressed if the picture portion is large. Furthermore, the background can be cleaned up (or whitened) by mapping background colors to their representative color (or white).

First of all, we need to estimate precisely the text and background colors on the document image. Color document images often contain many photographs and pictures with various colors. Therefore, color or tone histograms obtained from such images are composed of mixture distributions or multi-modal distributions containing many significant valleys and peaks. It is usually difficult to estimate precisely the text and background colors from such mixture distributions, because the histogram also contains spurious peaks and valleys. To avoid analysis of histograms composed of mixture distributions, we note that a color document image is locally composed of two types of regions: foreground and background. Based on this composition, the estimation problem for text and background colors can be reduced to a simple two-class problem:

- Based on some objective function, a subregion containing the true text and background colors is found out.
- The subregion is classified into two classes: foreground and background.
- From the distributions of foreground and background colors, the representative colors and their variations are estimated.

Furthermore, to maintain tone continuity of the midtone areas, black enhancement is suppressed if the picture portion is large. Areas whose colors lie outside the distributions for the text and background correspond to midtone areas, and therefore, the portion of picture areas over the entire image can be estimated based on the intensity distributions of the text and background.

The advantage of the proposed method is that no information on the characteristics of input devices is required; it can estimate parameters specifying the tone mapping based on statistical analysis of image features instead of using knowledge on the characteristics of input devices. This is essential when processing image data input from unknown input devices on remote sites, because *a priori* distributions of text or background colors are not available. Actually, the signal levels for simple white background range widely between 200 and 255 (in 8-bit grayscale) according to tone characteristics and parameter setting (e.g., gamma) of scanners. Examples of the adaptive tone correction and background cleaning/whitening are given in Fig. 5.

3.2   Restoring High-Resolution Binary Images for Text

Following the success of the MRC model, we find that a natural approach to improving legibility of text is to restore high-resolution binary images from the original low-resolution image. It is known theoretically that a gray-scale image in low resolution has the same amount of information as a high-resolution binary image. In practice, we need a method that

- can deal with characters with complex shape structures, such as Kanji,
- can improve OCR accuracy and legibility of text for the human eye (smoothness and linearity of contours and restoration of strokes) at the same time, and
- entails simple computation.

To satisfy the above three requirements, Nishida [10] presents a method for restoring high-resolution binary images from gray-scale images in low resolution. An
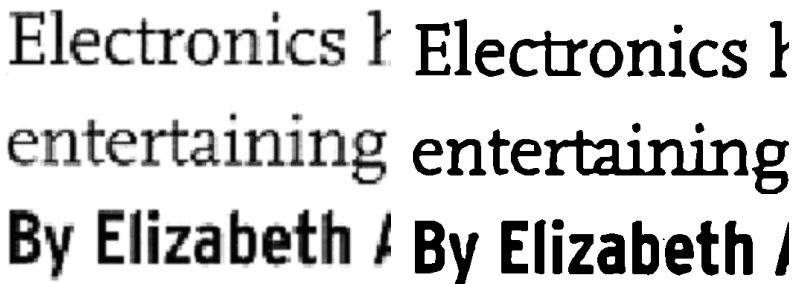

Fig. 3. Original low-resolution image and its restored high-resolution binary image

effective approach to tackling this complex problem is to utilize and integrate various types of information. The new method consists of the following four components:

(1) Generation of the initial high-resolution image by interpolation and local statistics (adaptive binarization)

(2) Complementing missing strokes based on topographic features

Regarding a gray-scale image as a surface, we examine the topographic features ("ridge", "valley", "peak", "pit", "hillside"). They are preserved even on low-resolution images, and therefore, can be used as cues for extracting strokes from text images. Missing strokes are complemented based on the topographic features.

(3) Contour modification: gradient magnitudes and curvatures along the contours

Even if missing strokes can be complemented, the resulting image is still poor in terms of text image quality because of fluctuations along contours. The quality can be improved if it is modified so that curvatures are small and the contour passes through pixels where gradient magnitudes are locally maximized. Contours are modified with the Active Contour Model or Snake.

(4) Contour beautification

Contours are beautified to correct distortions around right angles and fluctuation along horizontal and vertical directions.

Fig. 3 shows an example of the restoration of high-resolution binary images from original low-resolution images [10].
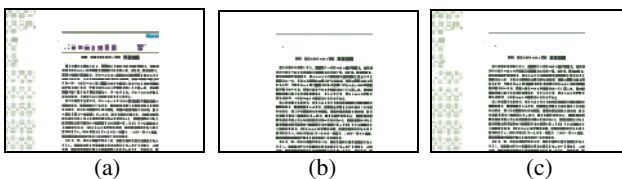
3.3 Text Color Clustering

The specific problem in text color clustering for document images is that text consists mostly of black fonts along with several other colors for emphasizing particular sentences or phrases. Typical profiles of color distributions show that there are a large cluster composed of a majority of colors (corresponding to the dominant color, namely black) and some small clusters, as can be observed from the example of Fig. 4a. We need to pay a particular attention so that small clusters can be picked up to keep them from being merged to the dominant cluster. Furthermore, computational efficiency is also important.

Color clustering has been investigated in the document image analysis area, with emphasis on color-based text extraction from webs or covers of books, magazines, and CDs [13]. There are fewer characters to be considered in these applications than in our problem, and the cluster size is well balanced across the distinct colors.

We take the following approach to the problems of ill-balanced cluster size and computational efficiency:

- Histogram-based clustering is applied separately to each component of a transformed color space, where the components are correlated only weakly.

(a)   (b)   (c)

Fig. 4. (a) Text color plane. (b) The largest color cluster. (c) The second largest cluster.

- The extraction of dominant clusters is followed by the extraction of small clusters corresponding to colors for emphasis or decoration.

We use the YCbCr color space because of its simplicity in the coordinate transformation and independency among the three components. The clustering technique based on Matas and Kittler [8] is applied to each component of YCbCr.

The extraction of dominant clusters proceeds as follows:

(1) The largest clusters (1-D intervals) are extracted from each component of YCbCr.

(2) A 3-D rectangular parallelepiped is generated by taking the Cartesian product of the intervals on Y, Cb, and Cr coordinates. If the number of samples inside this parallelepiped is larger than some specified ratio over the entire samples, a cluster for a dominant color is constructed in the following step (3). Otherwise, the extraction process completes.

(3) A cluster is defined as the parallelepiped, with its representative color being the Cartesian product of the representative values on each.

(4) Histograms on each component of YCbCr are constructed by excluding the colors within the constructed cluster. Repeat from step (1).

For example, the largest cluster is extracted as

$$\{(y, Cb, Cr): y \in [7,14], Cb \in [14,16], Cr \in [14,18]\},$$

including 69.4% of text colors as shown in Fig. 4b. The second largest cluster, occupying 23.2% of text colors as shown in Fig. 4c, is extracted as the three-dimensional parallelepiped [22, 25] x [14, 16] x [14, 18].

Small clusters corresponding to colors for emphasis are extracted in a similar way, but three-dimensional parallelepipeds are constructed for all combinations of extracted clusters on each of Y, Cb, Cr components. Table 1 shows the final results of the whole clustering procedure. There are two dominant clusters (#1 and #2) and four small clusters (#3 to #6).

Table 1. Results of text color clustering

	Cluster Parallelepipeds YCbCr (32levels)	Cluster Centers			Ratio
		YCbCr	RGB (255levels)	Color	
1	[7, 14] x [14, 16] x [14, 18]	(9, 15, 15)	(61, 79, 58)		69.4%
2	[22, 25] x [14, 16] x [14, 18]	(24, 14, 15)	(181, 202, 164)		23.2%
3	[10, 13] x [14, 16] x [17, 18]	(11, 18, 16)	(88, 83, 116)		3.5%
4	[15, 17] x [15, 15] x [15, 16]	(17, 15, 16)	(136, 138, 122)		1.5%
5	[17, 17] x [12, 12] x [16, 16]	(17, 12, 16)	(136, 147, 80)		1.5%
6	[14, 15] x [20, 20] x [9, 10]	(14, 20, 9)	(34, 140, 168)		0.9%

4   Optimization for Highly Legible Printing

We present an example task of highly legible printing of scanned document images. This task of "scan-to-print" is the most common in networked document imaging, because scanned document images are stored in repositories and will be printed by

Fig. 5. Examples of optimized images for printing. (a) Normalized original image in 200dpi. (b) Optimized image. (c) Normalized original image. (d) Optimized image.

some users at a remotely located office many days (or years) later. A typical situation is that the document image acquired with a color scanner in low resolution (200 or 300dpi) was stored in a document image database, and is printed out later through a high-resolution printer in 600dpi. This is an asynchronous situation in that the characteristics of the input and output devices are inconsistent and that the person having scanned the document does not know how, why, or when the image will be used in the future.

Regardless of particular tasks or preferences of the user, the normalizations are applied before the image is stored in a canonical representation absorbing specific differences due to the characteristics of input devices and the image acquisition conditions. The optimizations depend on the particular task in hand, the preferences of the user, and also the characteristics of the output device. The user can decide the parameters or options of the optimizations in the following way:

- Text super-resolution: The magnification factor depends on the resolutions of the original image and the output devices. For example, if the image is to be printed out from a printer in 600dpi, the magnification factor should be three and two for the image in 200dpi and 300dpi, respectively. However, the text super-resolution is ineffective and unnecessary when the resolution of the image happens to be higher than that of the output device (e.g., LCD displays).
- Text color clustering: Text super-resolution is associated with text color clustering.
- Tone mapping along with background cleaning and whitening: It depends on the user's preference whether the background color should be preserved or reduced to white.

The output image can be generated from a combination of these three components for optimization. Examples of optimized images for printing are given in Fig. 5. Physical size of these images is 15mm by 15mm. Fig. 5a shows a portion of a scanned magazine page in 200dpi after the normalizations are applied. This image is now optimized for printing in 600 dpi. Naturally, we apply text super-resolution by a factor of three (600dpi), followed by text color clustering. Furthermore, background whitening is selected for tone mapping because the paper color looks originally white. The optimized image for Fig. 5a is shown in Fig. 5b. Another example is given in Fig. 5c. The optimized image is shown in Fig. 5d with background cleaning, text super-resolution by a factor of three (600dpi), and text color clustering.

The optimization processes introduced in this paper are quite simple, but can be applied to and shared with various tasks such as retrieval (OCR and indexing), communication (compression), and browsing on displays or PDA, in addition to highly legible printing. High-resolution binary images obtained by text super-resolution can be used for OCR to improve recall and precision of indexing and retrieval, when the resolution of the input image is insufficient. The optimized images can also be transmitted efficiently if they are compressed with JPM (part of JPEG2000 [4]), to which the optimization processes conform. Adaptive tone mapping is still necessary when the images are browsed on displays and PDA to enhance contrast text and background. Text super-resolution will be a key technology to help senior or visually handicapped people read documents easily on displays, because they can enjoy fine visual quality after text is magnified, as illustrated in Fig. 5.

5   Conclusion

There could be a diversity of technologies involved in networked document imaging. It is now time to explore system architectures so that we can cope with this diversity in a systematic and extendable way. A key observation is that the diversity can be classified mainly into two factors: *Normalization* and *Optimization*. This paper has presented a system architecture for document imaging in an open, distributed environment over networks, composed of the above two sets of image processing operations. In network and Internet environments, digital image data captured through any image acquisition device can be transmitted to remote sites over networks, and the receiver can utilize the image data delivered via networks from unknown input

devices. Visual appearance of the document image is quite different according to the characteristics of image acquisition devices and the conditions of image acquisition. Furthermore, different types of image enhancement are required according to users' tasks or preferences. These diversities can be resolved systematically by placing the normalization at an upper end (routing servers) and the optimization at a lower end (clients) of the data flow over networks.

In view of this architecture, we have described some instances of the normalization and optimization associated with a particular task of highly legible printing of scanned document images. In particular, three essential algorithms have been mentioned for optimizing document images: adaptive tone mapping with background cleaning, text super-resolution, and text color clustering. The optimization process has been illustrated for highly legible printing, along with some other potential applications and tasks.

References

[1] T.M. Breuel, W.C. Janssen, K. Popat, H.S. Baird, "Paper to PDA," in *Proc.16th Int. Conf. Pattern Recognition* (Quebec City, Quebec, Canada), vol. 1, 476-479, 2002.

[2] J.D. Hobby and T.K. Ho, "Enhancing degraded document images via bitmap clustering and averaging," in *Proc. 4th Int. Conf. Document Analysis and Recognition*, August 1997.

[3] ISO/IEC 15444-1, Information technology -- JPEG 2000 image coding system -- Part 1: Core coding system, 2000.

[4] ISO/IEC 15444-6, Information technology -- JPEG 2000 image coding system -- Part 6: Compound image file format, 2003.

[5] M.Y. Jaisimha, E.A. Riskin, R. Ladner, and S. Werner, "Model-based restoration of document images for OCR," *Proc. SPIE*, **2660**, 297-308, 1996.

[6] L. Koskinen, H. Huttunen, and J.T. Astola, "Text enhancement method based on soft morphological filters," *Proc. SPIE*, **2181**, 243-253, 1994.

[7] H. Li, O.E. Kia, and D.S. Doermann, "Text enhancement in digital video," *Proc. SPIE*, **3651**, 2-9, 1999.

[8] J. Matas and J. Kittler, "Spatial and feature space clutering: applications in image analysis," in *Proc. Int. Conf. Computer Analysis of Images and Patterns*, 162-173, September 1995.

[9] H. Nishida, "Adaptive inverse halftoning for scanned document images through multiresolution and multiscale analysis," *Pattern Recognition,* **38**(2) 251-260, 2005.

[10] H. Nishida, "Restoring high-resolution binary images for text enhancement," in *Proc. Int. Conf. Image Processing* (Genova, Italy), vol. II, 506-509, September 2005.

[11] G. Ramponi and P. Fontanot, "Enhancing document images with a quadratic filter," *Signal Processing,* **33**, 23-34. 1993.

[12] Y.C. Shin, R. Sridhar, V. Demjanenko, P.W. Palumbo, and J.J. Hull, "Contrast enhancement of mail piece images," *Proc. SPIE*, **1661**, 27-37, 1992.

[13] K. Sobottka, H. Kronenberg, T. Perroud, and H. Bunke, "Text extraction from colored book and journal covers," *Int. J. Document Analysis and Recognition*, **2**, 163-176, 2000.

[14] P.D. Thouin and C.-I. Chang, "A method for restoration of low-resolution document images," *Int. J. Document Analysis and Recognition*, **2**, 200-210, 2000.