

Thresholding Technique for Document Images using a Digital Camera

Sadao Takahashi
Research and Development Group, Ricoh Co., Ltd.
Yokohama, Japan

Abstract

In recent years high-resolution digital cameras have become widespread. They can be used not only for landscapes and portraits, but also for documents. Although 24 bits are required for storing, viewing, and printing landscape and portrait images, only 1 bit is required for text images. Images captured with a digital camera are usually saved in a JPEG format in a limited memory card inserted into the camera. Therefore, implementing a function to binarize a document image to a digital camera is a very useful way of saving storage space. However, images captured with a digital camera generally have fluctuating luminance and therefore can not be binarized easily. The algorithm described in this paper uses the segmenting-and-interpolating scheme, which operates quickly to determine threshold values and create high-quality binary images. Experimental results show that the quality of the characters in the images thresholded using this algorithm is superior and therefore they can be input into an optical character recognition (OCR) software.

Introduction

CCDs with 3 megapixels are now on the market. And some kinds of digital cameras with such type of CCD will be released in the near future. The resolution of an image, in which the area of a letter-size paper is captured with this kind of digital camera, is equivalent to 200 dpi. It is almost equal to that of G3-standard facsimile. Therefore, the camera can be used as a mobile device to capture documents. Usually, images captured with a digital camera are saved in a JPEG format in a limited memory card inserted into the camera. JPEG images, including characters, should not be highly compressed since a high compression rate makes the decoded image unreadable. Although 24 bits are required for the storing, viewing, and printing of landscape or portrait images, only 1 bit is sufficient for text images. Therefore, implementing a function to binarize a document image to a digital camera is a very useful way of saving memory. However, digital camera images usually have fluctuating luminance which can not be binarized easily since the digital camera does not have a shading-correction system and captured images are also affected by external light sources. Figure 5(b) shows an example of an

image thresholded with a fixed value. When flash is shined on the image, the center of the image is clear but the rest of the image is black.

Many adaptive thresholding techniques^{1,2} have been developed in order to properly binarize fluctuating images. But they are often too complicated and need a lot of calculation in order to implement them into a digital camera. The algorithm suitable for digital cameras described in this paper uses the segmenting-and-interpolating scheme that achieves both fast thresholding operation and binary text images of high quality, even if the images contain fluctuating luminance.

The details of the proposed algorithm are described in the following section.

Algorithm

In Figure 1, the block diagram of the thresholding technique is shown. A JPEG image from a digital camera is assumed as the input image here. The color space of the input image is RGB, YCbCr, or grayscale. If the input image is color, the color component used in this algorithm is G or Y. G is preferable to Y since G has the highest resolution of all.

At first, the edge of the image data is enhanced. Usually, the edge is appropriately enhanced for the landscape or the portrait. Therefore, some additional edge enhancement is required to binarize the character image. The edge-enhancing method used in this algorithm is a conventional digital filter, as shown in Figure 2.

Then the edge-enhanced image is segmented into square regions. The size of the region depends on the image size. As mentioned in the experimental section, the size of the region is 128 x 128 pixels when the whole image has 2048 x 1536 pixels. In each region, an average of pixel values is calculated. Figure 3 shows the flowchart of averaging. While calculating the average, image data is sampled so that the calculation time is reduced. Then, sampled data are examined to see whether they are more than the lower limit Lth. If a sampled data is more than Lth, it is used to calculate the average in the region. Otherwise it is not. Since the purpose of the thresholding used in the proposed algorithm is to extract the background level and segment between the foreground, or characters and the background, extracting the background properly is important. Because of this, dark and large characters (such

as headings that include one of the regions) are not regarded as the background but as the foreground. Let sampling interval be T, the number of summed data be N, and the pixel value more than Lth at (x,y) position in a region be p(x,y). A(m,n), the average of the region(m,n) is described in equation (1)

$$A(m,n) = \frac{\sum_j \sum_i p(iT, jT)}{N} \quad (1)$$

If the image has 2048 x 1536 pixels, T is equal to 8. Since every 8th pixel is used to average the pixel values in a region, the calculation time is 64-times faster than that of the case when all pixels are used. The sampling interval gets longer if the image size gets larger. And the calculation time of the average does not get longer even if the size of the input image increases. If all data in one of the regions is equal to Lth or less, A(m,n) can not be determined by equation (1). In this case A(m,n) is set to 0.

After the calculation of A(m,n), the threshold value for the region Bth(m,n) is determined by equation (2)

$$Bth(m,n) = A(m,n) \times Cm \quad (2)$$

where Cm is the multiplying coefficient. Cm is adjusted so that characters of ordinary density (1.0 or more) are extracted. And Bth(m,n) is compared with Lth.

$$Bth(m,n) = \begin{cases} Bth(m,n) & (\text{if } Bth(m,n) > Lth) \\ Lth & (\text{otherwise}) \end{cases} \quad (3)$$

By applying equation (3), it is possible to binarize actual dark areas in the image as *black*.

Then the threshold value for each pixel is interpolated by using the threshold values for the regions. With this threshold value, each pixel in the image data is thresholded. The four regions that come in contact with each other can be seen in the left part of Figure 4. These four regions have Bth(m,n), Bth(m+1,n), Bth(m,n+1), and Bth(m+1,n+1), respectively. Each threshold value for the region is set as the threshold value for each corner pixel of the square R as shown in the right part of Figure 4. With these threshold values for corner pixels, each threshold value for each pixel in the square R is interpolated. Let the pixel position from the upper left corner of the square be (u,s), and that from the lower right corner be (v,t). Region size S is described in equation (4)

$$S = u + v = s + t \quad (4)$$

Before determining Pth(u,s) of a threshold value for pixel(u,s), Pth(0,s) and Pth(S-1,s) are interpolated.

$$Pth(0,s) = \frac{Bth(m,n)t + Bth(m,n+1)s}{S} \quad (5)$$

$$Pth(S-1,s) = \frac{Bth(m+1,n)t + Bth(m+1,n+1)s}{S}$$

Since threshold values for both ends of s-th horizontal line have already been determined, threshold values for all pixels on s-th horizontal line are interpolated with Pth(0,s)

and Pth(S-1,s). This interpolation strategy is much faster than the direct interpolation with four threshold values at the corners. Pth(u,s) is interpolated as described in equation (6).

$$Pth(u,s) = \frac{Pth(0,s)v + Pth(S-1,s)u}{S} \quad (6)$$

On the borders of the image, only one or two Bths are available. For example, only Bth(0,0) is available at the most upper-left corner of the image. In this case Bth(-1,-1), Bth(0,-1) and Bth(-1,0), are extrapolated from Bth(0,0).

$$Bth(-1,-1) = Bth(0,-1) = Bth(-1,0) = Bth(0,0) \quad (7)$$

At the upper-middle region between region(m,0) and region(m+1,0), Bth(m,0) and Bth(m+1,0) are available. In this case Bth(m,-1) and Bth(m+1,-1) are extrapolated as described in equation (8).

$$Bth(m,-1) = Bth(m,0) \quad (8)$$

$$Bth(m+1,-1) = Bth(m+1,0)$$

On other border areas in the image, similar extrapolation is executed and the threshold value for each pixel is determined by equations (5) and (6). Finally, the pixel value p(u,s) is thresholded with Pth(u,s).

$$p(u,s) = \begin{cases} white(1) & (\text{if } p(u,s) > Pth(u,s)) \\ black(0) & (\text{otherwise}) \end{cases} \quad (9)$$

After all pixels are thresholded, a binary image is created, and the operation is finished.

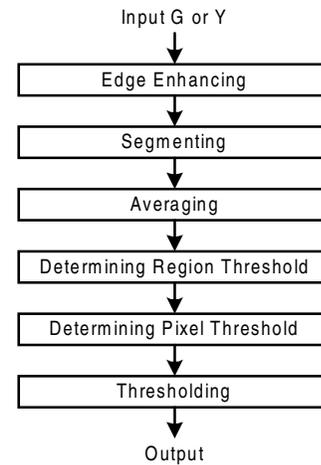


Figure 1. Block diagram of the proposed algorithm.

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	48	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

X 1/32

Figure 2. Edge-enhancing filter.

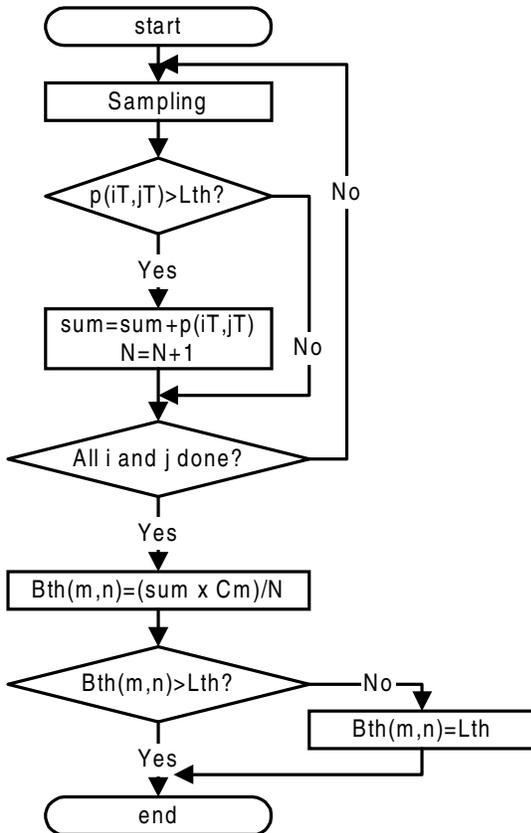


Figure 3. Flowchart of averaging.

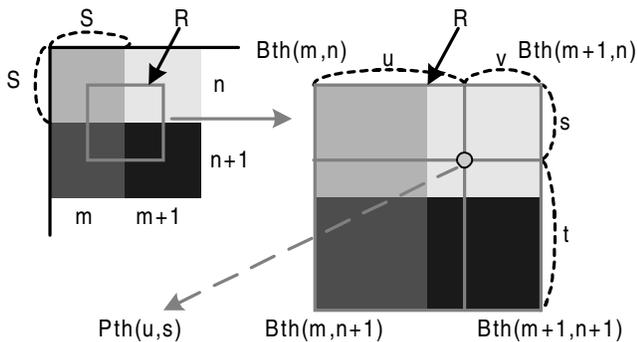


Figure 4. Interpolation of threshold for a pixel.

Experimental Results

In this section the experimental results of the proposed algorithm are shown. Capturing systems used in these experiments are RDC-4200 (Ricoh's 1.3-megapixel digital

camera), RDC-5000 (Ricoh's 2.3 megapixel digital camera), and the experimental capture system with 3.3-megapixel CCD. The parameters for these systems are shown in Table 1. Since the experimental capture system with 3.3-megapixel CCD does not have automatic exposure control and gamma correction functions, the parameters for this system are different from the parameters for other systems. The first example of images is shown in Figure 5. The original image in Figure 5(a) was captured with RDC-4200 with its flash. The fluctuation in the luminance in the image increases if the flash is used. Therefore, this example clearly illustrates that the proposed algorithm is an efficient way to take images with a flash

Low Contrast between Characters and Background

The second example shown in Figure 6 is a series of images in which the contrast between the characters and the background is low. Original images were captured with RDC-5000. Although binarized characters of 0.5-density are faint and insufficient, binarized characters with a density of 0.7 or more are clearly visible. By adjusting C_m in order to threshold the low-contrast image properly, characters with a density of 0.5 may be binarized clearly.

Low-Brightness Environment

The third example shown in Figure 7 is a pair of images captured with RDC-5000 in low-brightness environments. There are two purposes in this experiment: One is to examine the algorithm in an environment where users usually take pictures of documents, such as an office or library. The other is to examine the robustness of the algorithm for low S/N images captured in a lower-brightness environment. In this experiment the flash is prohibited and the shutter speed is fixed to 1/45 seconds since the surface reflection of the flash on the glossy paper and the slow shutter cause bad image quality. The exposure level is adjusted automatically by an automatic gain control circuit. Images taken in the office are thresholded properly as shown in Figure 7(a). Although the original image of Figure 7(b) shows worse S/N than that of Figure 7(a), the thresholded image of Figure 7(b) includes a little background noise in the printed area and characters are clearly visible.

Input to an OCR Software

If the image from the high-resolution digital camera is of good quality when input to the OCR software, the digital camera can be a useful mobile document scanner. We examined whether the quality of binary images from the experimental capture system with a 3.3-megapixel CCD is enough for OCR. The object was a document of letter size that includes 10-pt. Japanese characters printed by a 600-dpi laser printer. The resolution of the thresholded image shown in Figure 8 is equivalent to 200 dpi. A recognition rate of 99.4 % is achieved with Ricoh's OCR software "Yomitori Monogatari Ver. 3". This result shows that a 3-megapixel digital camera can be used as a mobile scanner for documents.

90年度関連メーカー月別売上金額表 単位：千円

	A社	B社	C社	D社	E社	合計
90/04	230000	360000	192300	120065	145328	1047693
90/05	300000	430000	164100	135264	161205	1190569
90/06	295000	440000	181200	143356	182699	1242251
90/07	270000	390000	230020	158322	248632	1296974
90/08	297000	412000	265900	169938	224488	1369326
90/09	240000	370000	255500	180002	210032	1255534
90/10	230000	395000	246800	194444	196632	1262876
90/11	267000	450000	212000	213222	152331	1294553
90/12	320000	490000	288120	230012	132487	1460619
91/01	242000	300000	142000	241654	168940	1094594
91/02	220000	320000	111000	276332	196323	1123658
91/03	195000	365000	156320	305641	241119	1263080
	3106000	4722000	2445260	2368252	2260212	14901724

(a)

90年度関連メーカー月別売上金額表 単位：千円

	A社	B社	C社	D社	E社	合計
90/04	230000	360000	192300	120065	145328	1047693
90/05	300000	430000	164100	135264	161205	1190569
90/06	295000	440000	181200	143356	182699	1242251
90/07	270000	390000	230020	158322	248632	1296974
90/08	297000	412000	265900	169938	224488	1369326
90/09	240000	370000	255500	180002	210032	1255534
90/10	230000	395000	246800	194444	196632	1262876
90/11	267000	450000	212000	213222	152331	1294553
90/12	320000	490000	288120	230012	132487	1460619
91/01	242000	300000	142000	241654	168940	1094594
91/02	220000	320000	111000	276332	196323	1123658
91/03	195000	365000	156320	305641	241119	1263080
	3106000	4722000	2445260	2368252	2260212	14901724

(b)

90年度関連メーカー月別売上金額表 単位：千円

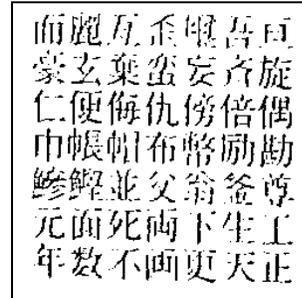
	A社	B社	C社	D社	E社	合計
90/04	230000	360000	192300	120065	145328	1047693
90/05	300000	430000	164100	135264	161205	1190569
90/06	295000	440000	181200	143356	182699	1242251
90/07	270000	390000	230020	158322	248632	1296974
90/08	297000	412000	265900	169938	224488	1369326
90/09	240000	370000	255500	180002	210032	1255534
90/10	230000	395000	246800	194444	196632	1262876
90/11	267000	450000	212000	213222	152331	1294553
90/12	320000	490000	288120	230012	132487	1460619
91/01	242000	300000	142000	241654	168940	1094594
91/02	220000	320000	111000	276332	196323	1123658
91/03	195000	365000	156320	305641	241119	1263080
	3106000	4722000	2445260	2368252	2260212	14901724

(c)

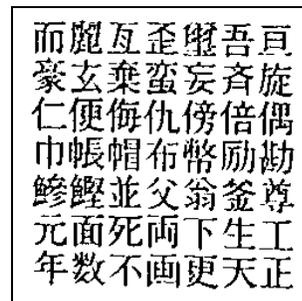
Figure 5. Original image and thresholded image. Each image size is 1280 x 960. (a) Original image, (b) thresholded image with a fixed threshold, and (c) thresholded image obtained by the proposed algorithm.

Table 1: Parameters and Values

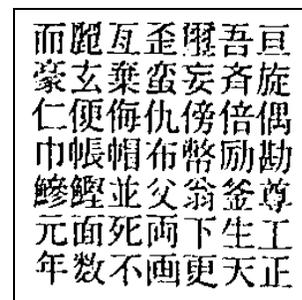
Parameters	Camera		
	RDC-4200	RDC-5000	3.3M CCD
Lth	10	10	24
Cm	0.84	0.84	0.66
S	64	64	128



(a)

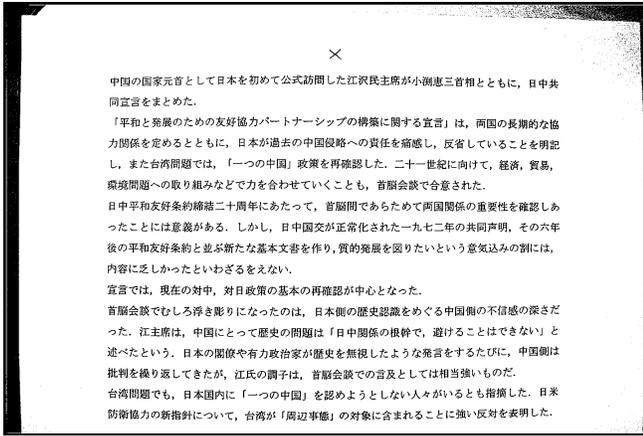


(b)

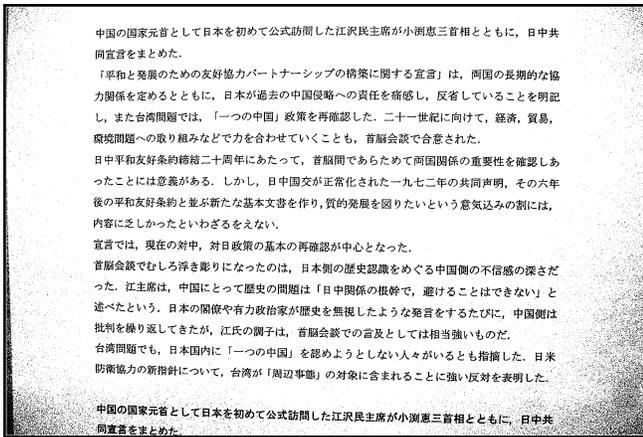


(c)

Figure 6. Low-contrast images. Each image size is 256 x 256. (a) density 0.5, (b) density 0.7, and (c) density 1.0.



(a)



(b)

Figure 7. Thresholded image captured in low-brightness environments. Each image size is 1792 x 1200. (a) $L_v=8.5EV$ and (b) $L_v=7.0EV$.

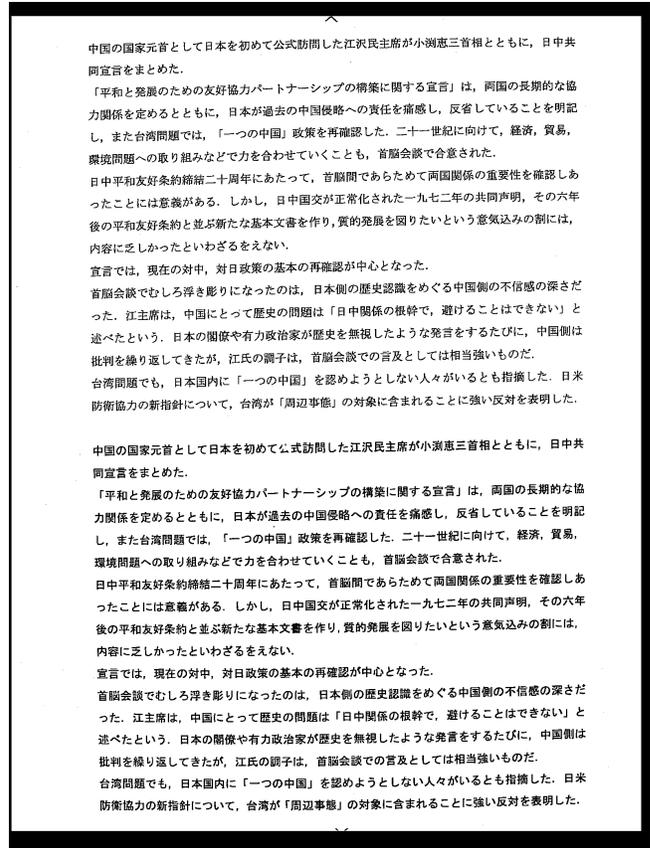


Figure 8. Thresholded image from 3.3-megapixel CCD.

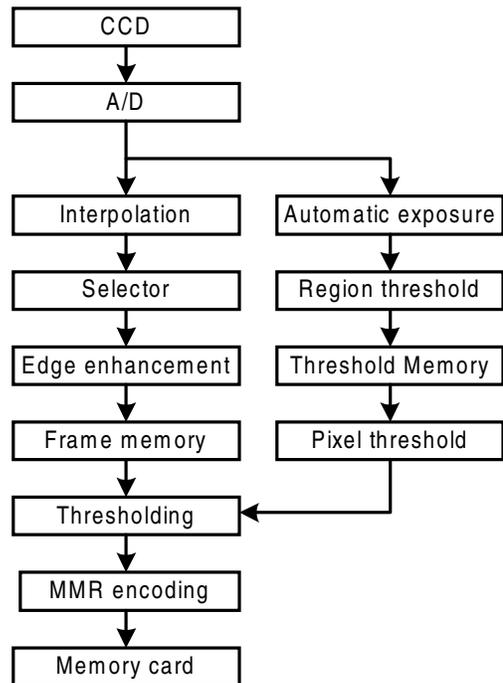


Figure 9. Implementation of the algorithm in the digital camera.

Implementation of the Algorithm in the Digital Camera

The block diagram of the proposed algorithm implemented in the digital camera is illustrated in Figure 9. At first, the analog image data from CCD is converted into digital data. Next, the digital data is interpolated and the full RGB data is created. From the RGB data, the G data is selected and the edge of G is enhanced. Then the edge-enhanced G is saved in the frame memory. At the same time, averages of the RGB data for each small region (which are the result of segmenting the image) are respectively calculated in the automatic exposure (AE) circuit. The AE circuit achieves the function described in equation (1). After the average of G is multiplied by the coefficient C_m and compared with L_{th} , B_{th} is determined in the region threshold (RT) circuit. The RT circuit achieves the function described in equations (2) and (3). Then B_{th} is saved in the threshold memory. In the pixel threshold (PT) circuit that has the function of equations (5) and (6), P_{th} is interpolated with B_{th} . By using P_{th} , the image data read out from the frame memory is thresholded. The thresholded image is then encoded by means of MMR so that it can be compatible with a facsimile. Finally, the encoded image is stored in the memory card in TIFF format, which most image-processing software applications support.

Conclusion

A thresholding technique for document images, which is suitable for implementation to a digital camera has been presented. The algorithm can properly binarize text images that have fluctuating luminance. Experimental results show that a 3-megapixel digital camera can be used as a mobile document scanner or mobile facsimile and that the binary document images from such a camera have sufficient quality to be input into an OCR software.

References

1. Kevin C. Scott, System and Method for Bidirectional Adaptive Thresholding, US Patent, 5,313,533, 1994.
2. Yongchun Lee et al., Multi-windowing Technique for Thresholding an Image Using Local Image Properties, US Patent, 5,583,659, 1996.

Biography

Sadao Takahashi received his B.S. degree in Electrical Engineering and M.S. degree in Electronic Engineering from University of Tokyo in Japan in 1988 and 1991 respectively. Since 1991 he has worked in the Research and Development Group at Ricoh Co., Ltd. in Yokohama, Japan. His work had primarily focused on the document image processing for color copier such as text segmentation, filtering, and color correction. Since 1997 he has started the research of image processing for digital camera.