應用於文件分析的彩色表格二元量化技術
Binary Quantization of Color Table Images for Document Analysis

孫宏明
Suen Hong-Ming

成功大學資訊工程系
Department of Computer Science and Information Engineering, National Cheng Kung Univ.
sunhm@server2.iie.ncku.edu.tw

王駿發
Wang Jhing-Fa

成功大學資訊工程系
Department of Computer Science and Information Engineering, National Cheng Kung Univ.
wangjf@server2.iie.ncku.edu.tw

摘要

表格處理是屬於文件分析技術的一部份。早期的研究全部專注於黑底白字形態的表格，在本論文中我們提出了一個有效的方法將彩色表格轉換成為黑底白字形態的表格，以便後續處理。
關鍵字: 文件分析，表格處理，彩色影像處理

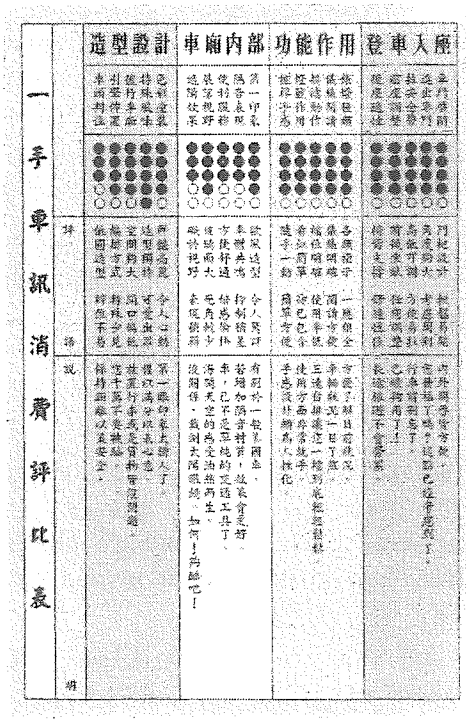
Abstract

Table processing is a subdomain of document analysis technology. Earlier researches are all based on the assumption that data is presented in the white-background/black-text (WB/BT) binary type. In this paper we propose a method for transforming color tables into binary format to facilitate subsequent processing.
Keywords: document analysis, table processing, color image processing

1. Introduction

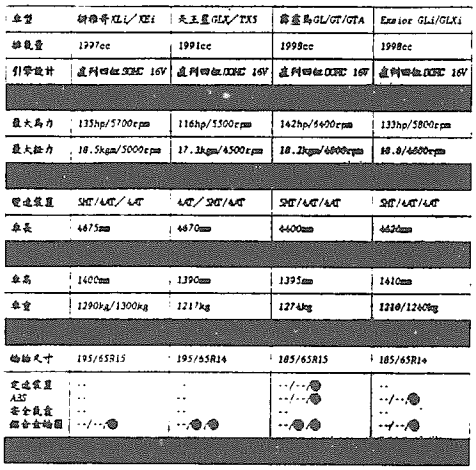
Many works concerning document image processing have been reported during the latest decade. This research field has involved a great number of studies because there is more and more urgent requisite to transform volumes of paper-based documents into a digitized format. A digitized document has many advantages over a paper-based one. For example, a digitized document needs much less space to store than a paper-based document; a digitized document can also be retrieved and reproduced in a more efficient and economical way. A more significant strength of digitized documents is that they can be conveyed by networks and this fact makes them gain more special appreciation in the modern applications. However, to give computers an ability to recognize paper-based documents is a challenging task. A number of prototype systems dedicated to some specific domain have been explored in recent years. A category of them concentrated on dealing with table of form images [1-10]. To simplify the task, all the

(a)

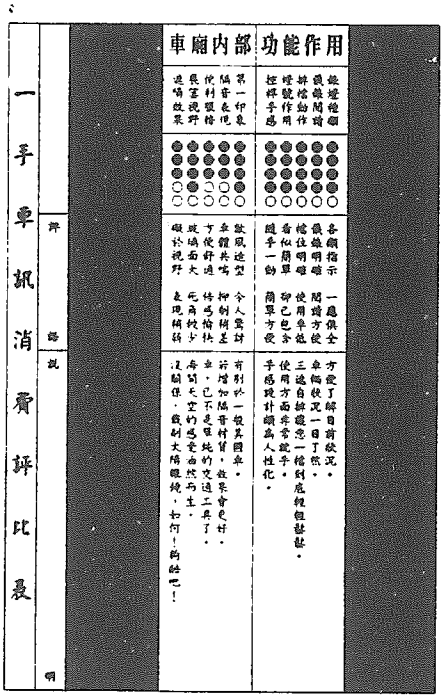


(b)

Fig. 1. Color tables presented in Chinese magazines.



(a)



(b)

Fig. 2. Table images obtained by scanning the color tables shown in Fig. 1 using the binary mode.

proposed techniques have made a prior assumption that the input table or form images are in the white-background/black-text (WB/BT) binary format. Although this condition is commonly held for monochrome tables, it is not always true for the color tables appearing frequently in today's magazines and newspapers (see Fig. 1 for example). Hence, if we want to employ these existing techniques to process color tables, the color table must be quantized into the WB/BT type first. A simple method of doing that is to directly scan the color table using the binary format. The table images shown in Fig. 2 are obtained in this way. It can be seen that the results are not satisfying because some text has been smeared seriously in these images and therefore a considerable piece of information has been lost. To avoid this problem, a more sophisticated method is required to achieve this task. We present in this paper an approach toward this end.

2. Proposed approach for binarizing color table images

In a color table, table cells can be of any color and so can the characters in the table cells. Hence, we can not make any *a priori* assumption on the colors of characters and the table cells. However, if we want to directly detect the colors of characters or table cells on the color image, the computational complexity will be relatively high because there can be 16 million colors in a true-color image (this is the image format used in our study). In this paper, we propose an efficient method for solving such a problem. We first use a color edge detection technique to find out edges in the color table image. For each pixel with coordinates (i, j), its gradients on the R, G, and B color bands are computed, respectively, using the Sobel operators [11]. For instance, the gradient at pixel p(i, j) on the R color band is computed by the following formula

$$\nabla f_R = \begin{bmatrix} G_{R_x} \\ G_{R_y} \end{bmatrix}$$

where

$$G_{R_x} = p_R(i+1,j-1) + 2p_R(i+1,j) + p_R(i+1,j-1) - p_R(i-1,j-1) - 2p_R(i-1,j) - p_R(i-1,j+1),$$

$$G_{R_y} = p_R(i+1,j+1) + 2p_R(i,j+1) + p_R(i-1,j+1) - p_R(i+1,j-1) - 2p_R(i,j-1) - p_R(i-1,j-1).$$

The gradients at pixel p(i, j) on G and B color bands (i.e., ∇f_G and ∇f_B) can also be computed in a similar manner. Since edges in color images appear at the points where R, G, and B color values have abrupt transitions, the edge strength of each pixel can be evaluated by

$$\text{edge strength} = |\nabla f_R| + |\nabla f_G| + |\nabla f_B| .$$

Then, the pixels with high edge strength will be regarded as the edge points. During processing, the result is written into a new binary image: edge points are represented by 1 (as white) and non-edge points are represented by 0 (as black). Fig. 3 shows the binary edge image of the table image shown in Fig. 1(a).

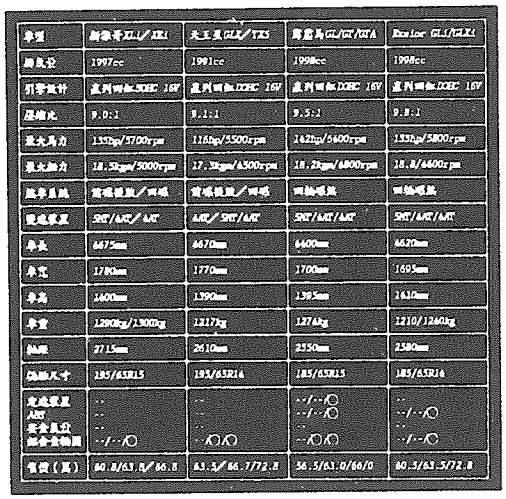


Fig. 3. Binary edge image of Fig. 1(a).

From the viewpoint of a reader, no matter how a color table is organized, the table background and data cells are both meaning just background parts in contrast to the foreground characters and line structures. Thus, if we can identify these background areas in the table image, then a WB/BT version of the table image can be obtained by resetting all the pixels belonging to the background areas to white and other pixels to black. Examining carefully the binary edge image shown in Fig. 3, we can see that the black regions in the image are just the background areas in addition to some small fragments within characters. To pick out the desired background areas, we first use a connected component labeling algorithm to locate all the black regions in the binary edge image. During the labeling procedure, the location, color, and pixel count of each black region are memorized. The location of a black region is represented by a bounding rectangle that encloses it. The color of a black region is represented by the color of the corresponding region in the original table image. The pixel count of a black region is the total number of pixels belonging to the region.

After the black regions on the binary edge image are extracted, we then make a pixel count histogram of them, as illustrated in Fig. 4. It can be seen that these black regions clustered distinctly into two groups in the pixel count histogram. Those regions with pixel count fewer than 1000 gathered as class one while the regions with pixel count more than 15000 gathered as class two. By tracing the locations of the black regions of class one, we found that they are just the small fragments within characters; on the other hand, the black regions of class two are the background areas. This is not just a special situation in this illustrative case. In fact, after examining over twenty different table images, we realized that the

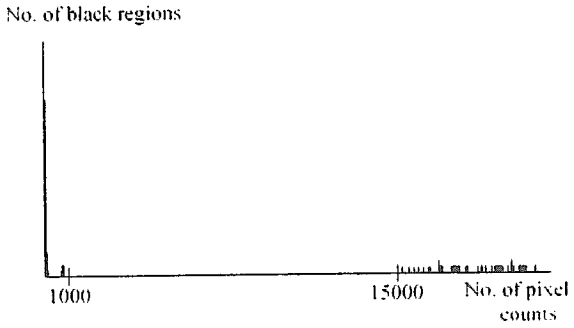


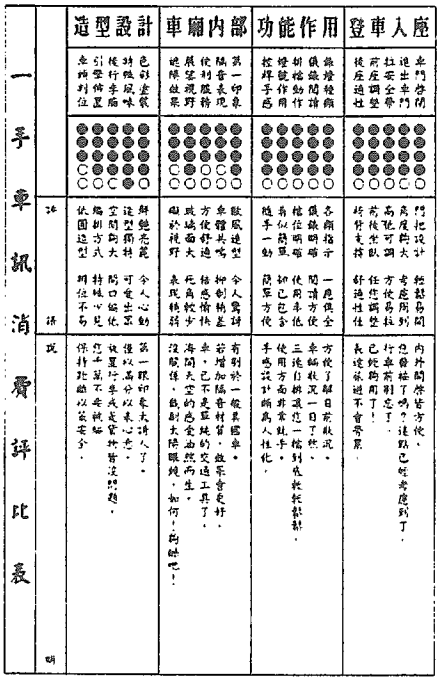
Fig. 4. Pixel count histogram of the black regions extracted from Fig. 3.

pixel counts of the small fragments within characters are always much fewer than the pixel counts of the background regions in a table image. According to this practice, the background areas can be distinguished from all the extracted regions by using only a filter for the pixel count. To determine a proper threshold value for filtering off the spurious fragments within characters, we initially scan the pixel count histogram within the range of 0 to 10000 to find the longest valley in this range. Suppose that the located valley starts from s and ends at t. Then the threshold value is chosen to be $(s+t)/2$. Once the background areas are identified, a WB/BT format of the color table image can be obtained by converting all the pixels belonging to the background areas to white and other pixels to black. As all the extracted background regions have been specified by respective bounding rectangles and their colors are also recognized, this pixel type conversion can be conveniently achieved by sequentially examining each pixel within the bounding rectangles to see if its color is close to the background color. Those pixels with nearly background color are considered as background parts and hence are reset to white while other pixels are reset to black as foreground components. When this conversion finishes, the resulting image is just the desired result.

3. Experimental results and discussion

The proposed approach was implemented and tested on a Pentium/200 PC. Fig. 5 illustrates the processing results of the table images shown in Fig. 1. As can be seen, these table images are successfully converted into the WB/BT binary format. The quality of the resulting images are also far better than those obtained through directly scanning the tables using the binary mode, as shown in Fig. 2. Fig. 6 is a table image scanned in a skew type, together with its processing result. This case demonstrates that our approach still work well even if the table image is severely skewed. To sum up, since the proposed approach does not

(a)



(b)

Fig. 5. Processing results of the table images shown in Fig. 1.

make any prior assumption on the color style and layout of the table images, it can be used to transform a wide fashion of color tables into the WB/BT binary format for further table recognition, and its practicability has been proved by test on over twenty different color table images.

References

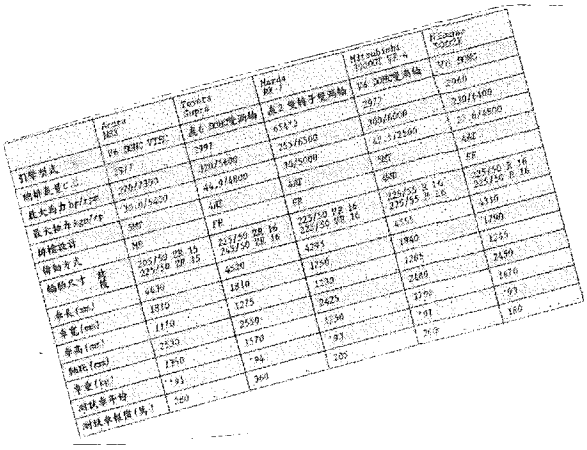
[1] Toyohide Watanabe, Qin Luo, and Noboru Sugie, "Layout recognition of multi-kinds of table-form documents." *IEEE Tans. PAMI*, Vol. 17, No. 4, 1995, pp. 432-445.

[2] Stephen W. Lam, Ladan Javanbakht, and Sargur N. Srihari, "Anatomy of a form reader," *Proc. 2nd International Conf. on Document Analysis and Recognition (ICDAR)*, 1993, pp. 506-509.

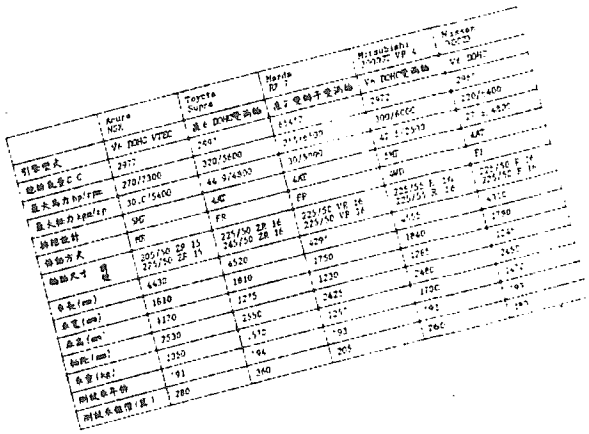
[3] Jenn-Yih Lin, Chi-Wei Lee, and Zen Chen, "Identification of business forms using relationships between adjacent frames," *Machine Vision and Applications*, Vol. 9, 1996, pp. 56-64.

[4] Suzanne Liebowitz Taylor, Richard Fritzson, and Jon A. Pastor, "Extraction of data from preprinted forms," *Machine Vision and Applications*, Vol. 5, 1992, pp. 211-222.

[5] Kuo-Chin Fan, Jeng-Ming Lu, and Liang-Sheng Wang, and Hong-Yuan Liao, "Extraction of characters from form documents by feature point clustering," *Pattern Recognition Letters*, Vol. 16, 1995, pp. 963-970.



(a)



(b)

Fig. 6. (a) skewed table image; (b) processing result.

[6] Richard Casey, David Rerguson, K. Mohiuddin, and Eugene Walach, "Intelligent forms processing system," *Machine Vision and Applications*, Vol. 5, 1992, pp. 143-155.

[7] Toyohide Watanabe, Hiroyuki Naruse, Qin Luo, and Noboru Sugie, "Structure analysis of table-form documents on the basis of the recognition of vertical and horizontal line segments," *Proc. 1st ICDAR*, 1991, pp. 638-646.

[8] Chang D. Yan, Yuan Y. Tang, and Ching Y. Suen, "Form understanding system based on form description language," *Proc. 1st ICDAR*, 1991, pp. 283-293.

[9] Dacheng Wang and Sargur N. Srihari, "Analysis of form images," *Proc. 1st ICDAR*, 1991, pp. 181-191.

[10] Jianxing Yuan, Lei Xu, and Ching Y. Suen, "Form items extraction by model matching," *Proc. 1st ICDAR*, 1991, pp. 210-218.

[11] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, 1992, Addison-Wesley.