The 13th IPPR Conference on
Computer Vision, Graphics and Image Processing

Book Content Digitization and Display for Digital Libraries by Document Image Analysis and Compression-By-Classification Techniques

Shih-Hao Chen (陳世豪) and Wen-Hsiang Tsai (蔡文祥)

Department of Computer and Information Science
National Chiao Tung University
1001 Ta Hsueh Rd., Hsinchu, Taiwan 300, R.O.C.
Tel: 886-3-5712121 Ext. 56650
Email:whtsai@cis.nctu.edu.tw

Abstract

In this study, an offline automatic book content digitization and display system is developed. First, we utilize an automatic document feeder (ADF) to scan multiple book pages into a computer. Then, we segment and classify page images into text blocks and picture blocks by an adopted bottom-up segmentation approach. In order to save book content storage space, we employ a compression-by-classification approach. In the approach, first we propose an image content classification method using a decision tree to classify picture blocks into various types, based on the properties of picture blocks as well as the use of a full-color hierarchical moment-preserving color reduction method. After classification of page contents, we propose a content-based compression scheme, which compresses different image blocks by appropriate compression algorithms according to their image attributes. A color reduction algorithm is adopted to eliminate distortion caused by printing or scanning and preserve the most important colors in image blocks, achieving a great deal of compression effect. Besides, we propose a repetitive-pattern recognition approach to detect common parts among different page images in order to improve compression effect further. Finally, we enhance page contents and provide a user-friendly interface for book contents display and reading. Experimental results show the feasibility and practicability of the proposed approaches.

(KEYWORD: Digital Libraries, Document Image Analysis, Compression-By-Classification)

1. Introduction

Today, with the widespread use of personal computers and the Internet, the demand for transforming existing printed documents into electronic form is very high nowadays, especially for those documents with enormous quantities, such as the collection of books in a library. It is well known that electronic documents are more suitable for archiving and retrieval than printed-paper documents. Therefore, book digitization is the most important task in a digital library.

The amount of data necessary for the image-based approach to represent document images is great, so data compression is essential for efficient transmission and archiving. The RightPages system [1] was designed for document image transmission over a local area network. Similar systems also have been developed recently [2, 3]. The most notable example is the DjVu system [4]. The DjVu system is designed for high-resolution, high-quality color document images, and for fast transmission of document images over low-speed connections.

A lot of works have been done for the purpose of image compression and most of them

concentrate on processing single document images. However, little attention has been paid to the digitization of an entire book, which includes multiple information-related pages instead of single-page documents. In this study, we try to design a system to deal with the digitization and display of book contents. By integration of document image analysis and compression techniques, it is hoped to achieve the objectives of compressing book contents more efficiently as well as displaying stored contents more friendly. Shown in Fig 1 is a diagram of the proposed book digitization system.

2. Analysis of Page Contents

2.1.Segmentation of Page Contents

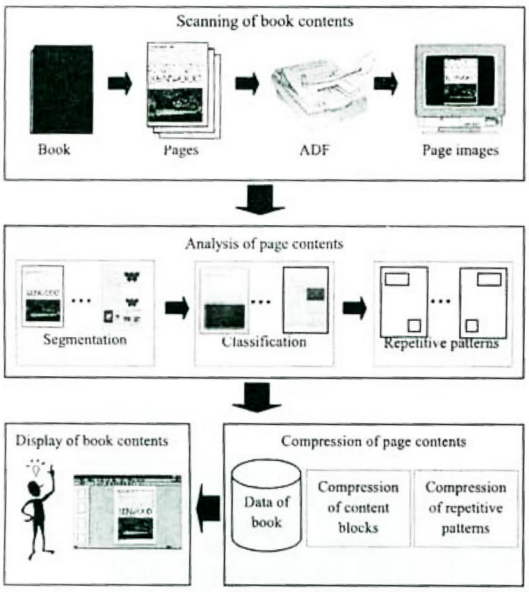


Fig.1. Proposed book digitization system with four modules : (I) Scanning of book contents. (II) Analysis of page contents. (III) Compression of page contents. (IV) Display of book contents.

After scanning book pages with an automatic document feeder, the resulting digital page images need be analyzed for subsequent utilization. In order to improve overall compression effect by applying appropriate compression algorithms to various images according to their types, we propose a new scheme shown in Fig 2 to distinguish page contents containing text and pictures. The scheme first segments page images into basic blocks and classifying these blocks into different image types.

In this study, we adopt the bottom-up approach proposed by Tsai and Chan [5] to segment page images into text and picture blocks. Color reduction is a useful technique in page document analysis and compression. In this study, we adopt the color reduction algorithm, which is based on the so-called hierarchical moment preserving principle developed by Tsai and Huang [6]

After the segmentation stage, the basic blocks are classified by a decision tree classification approach proposed in this study into different image types according to their content information.

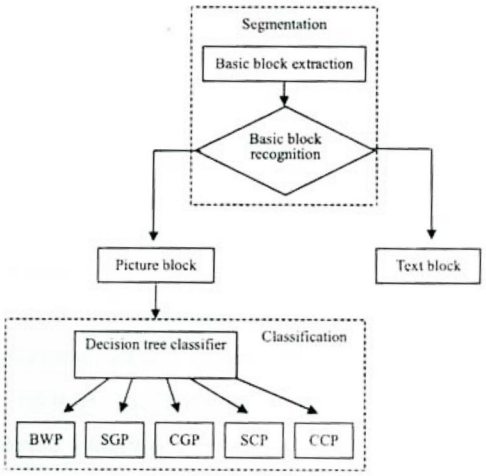


Figure 2. Flowchart of proposed page contents analysis system.

2.2. Classification of Page Contents with Proposed Decision Tree Approach

After page segmentation, the page image is classified into two basic types: text blocks and picture blocks. In order to improve the compressing result, we have developed a content-based decision tree classifier that analyzes and classifies picture blocks into more different types.

A decision tree, like the one shown in Figure 3, is proposed in this study for use to classify images by the color contents into several distinct types. The decision tree consists of four decision points, each of which uses a set of features extracted from the images. It classifies images into five different types:

1. B/w picture (BWP) —a picture with two levels, namely, black and white;
2. Simple gray picture (SGP) —a gray-level picture with only a few levels;
3. Complex gray picture (CGP) —a gray-level picture with a lot of levels;
4. Simple color picture (SCP) —a color picture with only a few colors, like a carton picture;
5. Complex color picture (CCP) —a color picture with a lot of colors.

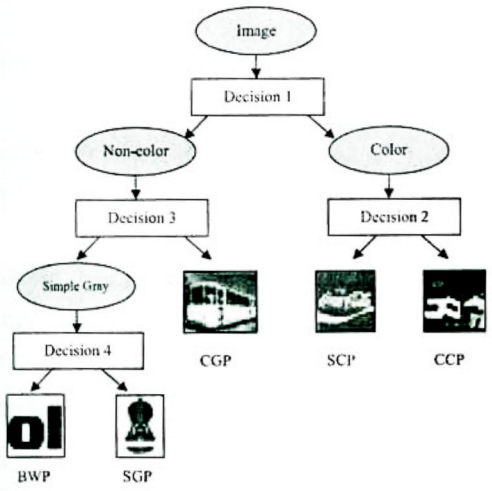


Fig 3. The proposed decision tree which consists of four decision points for classifying images into five different types

following.

Algorithm 2.1 Classification of color and non-color picture blocks (Decision point 1)

Step 1: Input the image *I* of a picture block with an unknown-type. Obtain the three histograms *Hr*, *Hg*, and *Hb* of the RGB channels of *I*.

Step 2: Compute three difference histograms *Hrg*, *Hgb*, and *Hbr* according to *Hr*, *Hg*, and *Hb*, respectively.

Step 3: Add up the values of these three difference histograms to obtain three sum values S*rg*, S*gb*, and S*br*, respectively.

Step 4: If the maximum of S*rg*, S*gb*, and S*br* is larger than a threshold *Ts*, classify *I* as the color image type. Otherwise, classify it as the non-color image type.

Algorithm 2.2 Classification of simple and complex color picture blocks (Decision point 2)

Step 1: Input the image *I* of a color picture block.

Step 2: Reduce the number of colors of *I* using the color reduction algorithm [6] to obtain an image *Ir* that contains only a few major colors.

Step 3: Calculate the three difference values of the RGB values of each pixel, respectively, between *Ir* and *I*. Add up these difference values for each color channel and compute the mean values to get three mean difference values *Mr*, *Mg*, and *Mb*.

Step 4: If the maximum of *Mr*, *Mg*, and *Mb* is larger than a threshold *Tc*, classify *I* as a CCP. Otherwise, classify *I* as an SCP.

Algorithm 2.3 Classification of simple and complex gray picture blocks (Decision point 3)

Step 1: Input an image *I* of a non-color picture block.

Step 2: Reduce the number of colors of *I* using the color reduction algorithm [6] to obtain an image *Ir* that contains only a few major colors.

Step 3: Calculate the three difference values of the RGB values of each pixel, respectively, between *Ir* and *I*. Add up these difference values for each color channel and compute the mean values to get three mean difference values *Mr*, *Mg*, and *Mb*.

Step 4: If the maximum of *Mr*, *Mg*, and *Mb* is larger than a threshold *Tg*, classify *I* as a CGP. Otherwise, classify it as an SGP.

Algorithm 2.4 Classification of b/w and simple gray picture blocks (Decision point 4)

Step 1: Input an image *I* of a simple gray picture block.

Step 2: Threshold *I* by the bi-level moment preserving thresholding algorithm [9] to obtain a b/w image *Ib*.

Step 3: Calculate the gray-value difference of each pair of corresponding pixels between *I* and *Ib*. Add up these difference values and compute the mean difference value *Mb*.

Step 4: If the mean difference value *Mb* is smaller than a threshold *Tb*, classify *I* as a BWP. Otherwise, classify it as an SGP.

3. Recognition of Repetitive Patterns in Pages

From a local viewpoint, we have already proposed a compression-by-classification approach to compress single-page images as described in the last section. In order to increase compression rates, we propose further a scheme that deals with repetitive patterns in pages from a global viewpoint.

Based on our observation, there is one property in book contents that other types of documents do not have. The property is that there often exist common parts, such as headers, footers, etc., as well as similar layout features among different pages in a book. If two individual pages have some common parts that represent identical contents, it is not necessary to save the contents twice. This data redundancy, called *inter-page redundancy*, can be exploited to reach more higher compression rates.

We propose a voting algorithm to detect candidate repetitive patterns in pages using the x-axis, y-axis positions of blocks and the area sizes of blocks as features. According to these three types of features, the segmented image blocks are then partitioned and put into a 3D cell space with each cell called a *bin*. Candidate repetitive patterns are detected from these bins finally. The detailed process is described as an algorithm in the following.

Algorithm 3.1 Detection of candidate repetitive patterns in pages.

Step 1: Compute the x-axis, y-axis positions and the areas of the blocks in the input page image.

Step 2: For each block, decide which bin in the cell space the block should fall in according to the three feature values of the block. The value of the bin then is incremented by one.

Step 3: Repeat Step 2 until all blocks of each page image are processed.

Step 4: Repeat Step 1 to Step 3 on each input page image until all input page images are

processed.

Step 5: If the value of a bin in the cell space is larger than a threshold *Tg*, then find the block corresponding to this bin. All blocks found in this way are collected as candidate repetitive patterns.

After detecting candidate repetitive patterns in pages, we can obtain multiple lists that contain different types of repetitive patterns. In each list, we obtain some repetitive patterns that contain similar geometric relationships and block sizes, but their contents might be different actually. So, we need further a method to verify the contents of image blocks to ensure that the contents of a list of candidate repetitive patterns are the same. We propose an algorithm for this purpose based on the template-matching concept, as described in the following.

Algorithm 3.2 Verification of repetitive patterns in pages based on template matching.

Step 1: Extract a list of candidate repetitive patterns and take the first candidate repetitive pattern in the list to be the base repetitive pattern *Rb*.

Step 2: Select a candidate repetitive pattern *Ri* from the list. Compare *Ri* with *Rb* by template matching. If they are similar enough, *Ri* is marked as a repetitive pattern and represented with an index number of the list. Otherwise, it is discarded.

Step 3: Repeat Step 2 until each candidate pattern in the list is compared with *Rb*.

Step 4: Take another candidate repetitive pattern in the list to be the base *Rb*, and repeat Steps 1 through 3, until all patterns in the list have been taken as the base.

Step 5: Repeat Steps 1 through 4 until each list is processed.

4. Compression of Page Contents

In this study, to reduce inherent information redundancy in text and picture images, a content-based compression scheme for document page images is proposed. We also propose a repetitive pattern compression approach especially for book contents. To deal effectively with page images that contain text blocks and different types of picture blocks, it is necessary to detect type information from page images, and employ different compression techniques for different image types. This is the main idea of the proposed content-based compression scheme. After analysis of page contents, page blocks are classified into six type: text block, black/white picture (BWP), simple gray picture (SGP), complex gray picture (CGP), simple color picture (SCP), and complex color picture (CCP). We utilize different compressing methods for different block types according to their content properties.

4.1 Compression of Textual Blocks

Various methods have been considered for textual image compression. Most textual images are stored using the generic bi-level image technique. However, it is often found that textual images generated in this way are hard to read and causes more eyestrain on a display. Based on our observation, most textual images have only a few major gray levels. For this reason, we adopt two bits per pixel (4 shades of gray) to keep a balance between the legibility and the compression rate.

We apply the color reduction algorithm [6] and the JBIG compression algorithm [7] to

compress textual images. The proposed method is described as follows.

Algorithm 4.1 Compression of textual block images.

Step 1: Reduce an input textual image to four colors by the color reduction method [6].

Step 2: After color reduction, represent each pixel of the resulting image with two bits (4 shades of gray).

Step 3: Separate the bit streams of the resulting image into two bit planes.

Step 4: Compress these two bit planes respectively by the JBIG compression algorithm.

4.2. Compression of Color and Non-color Blocks

Color image blocks are classified into two types in this study: complex color picture (CCP) and simple color picture (SCP). As implied in its name, a CCP such as a color photograph has complex color information in the image. For this kind of images, we simply apply the JPEG compressing algorithm [8] to them.

On the other hand, an SCP tends to have regions of constant colors and contains a few major colors. Therefore, we apply the color reduction algorithm [6] and the JBIG compression algorithm to compress an SCP. The steps of the proposed method are described as follows.

Algorithm 4.2 Compression of simple color pictures.

Step 1: Reduce the color numbers of an SCP to 16 colors by the color reduction method [6].

Step 2: After color reduction, use four bits to represent each pixel of the resulting image.

Step 3: Separate the bit streams of the resulting image into four bit planes.

Step 4: Compress these bit planes respectively by the JBIG compression algorithm.

Non-color image blocks are classified into three types in this study: b/w picture (BWP), simple gray picture (SGP), and complex gray picture (CGP). A BWP has only black and white in the image content. Obviously, it is suitable to apply the JBIG compressing algorithm to a BWP. An SGP has very similar properties as an SCP. Just like an SCP, an SGP also tends to have regions of constant gray levels and contains a few major ones of them. So, for SGP's we adopt the same compression algorithm as that we apply to SCP's.

4.3. Compression of Repetitive Patterns

Inter-page redundancy, which means data that represent the same information between any two pages in a book, can be eliminated to increase the overall compression rates. After detection and verification of candidate repetitive patterns, we obtain a sequence of repetitive patterns. The size of the sequence depends on the amount of identical parts among the pages in a book. We use a list data structure to store these repetitive patterns in a global structure of a book. These repetitive patterns are compressed according to their block types based on the proposed content-based compression scheme as described previously. We use an index number of the list to represent the same page content in the pages that contain repetitive patterns. That is, we only store a single "physical" copy of the repetitive patterns in a global structure and for the other pages that contain the repetitive patterns we store only an

index symbol for each page to represent the body of the corresponding repetitive pattern. It is obvious that the more repetitive patterns we obtain, the higher compression rates we can achieve.

5. Enhancement and Display of Page Contents

After book contents are compressed into electronic form, it is clear that a reduction of the page image size will also reduce the quality of the image. Hence, we need to investigate how to display the compressed book contents clearly on a computer screen. We also need to provide a convenient way to let users read the electronic book contents easily. To display the book contents properly on a computer screen, we must consider the legibility of book contents and the convenience of system operations. In this study, we utilize several techniques to enhance the page contents for display and to design a convenient user interface for book content reading.

In the compression methods for textual images that we described previously, we adopted two bits per pixel (4 shades of gray) to keep a tradeoff between the legibility and the compression rate. If textual images are generated on a display with gray-scaling capability, visual clarity can be improved by using more shades of gray. Additionally, it is also desired to improve the quality of textual images shown on the computer screen by digital signal processing techniques without increasing the storage. For this, we apply Gaussian blurring on the compressed textual image to maintain maximum readability of text in practice. Some experimental results will be shown later.

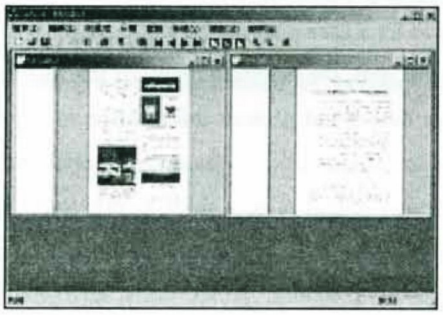
A book content digitization system should also provide a good interface for displaying the stored book contents on a screen and offer a convenient way to let a user read digital book contents easily. For this aim, we maintain many of the features of paper books and present information in a format as close as possible to the original book format. The page sequence of a book is preserved and a logical view that contains thumbnails of page images is provided to let users navigate book contents easily. In addition, we provide some necessary tools really usable and effective for reading digital book contents on a screen. These tools are classified into two categories as described in the following.

(1) Navigating tools – we provide "First Page", "Last Page", "Previous Page", "Next Page", and "Go to Page" to let users browse page contents easily and efficiently.
(2) Viewing tools – we provide "Actual Size", "Fit to Window", "Fit to Width", "Zoom In" and "Zoom Out" to let users read book contents entirely, partially or progressively.

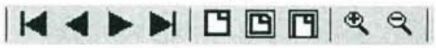
The interface for digital book display and some examples of book content reading are shown in Figure 4.

6. Experimental Results

In this section, some experimental results are shown. The experimental results of page content classification and are shown in Fig 5. The experimental results of color reduction of different text blocks using various numbers of gray levels are shown in Fig 6. Fig 7 shows the comparison of compression ratios for two kinds images using 4 compression methods.



(a)



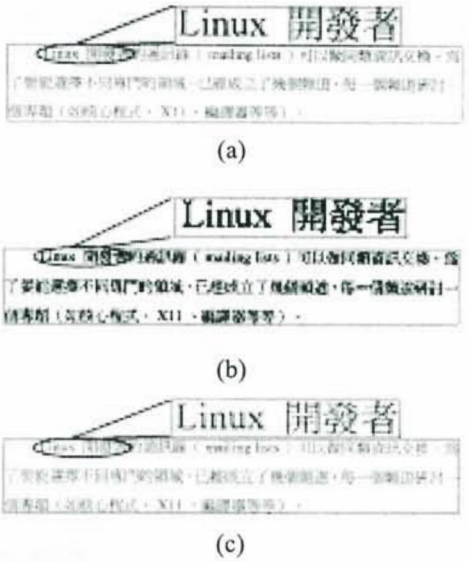
(b)

(c)

Fig 4. Illustration of proposed user interface for book reading and display. (a) User interface of the system. (b) Necessary tools for navigating and viewing book contents. (c) Some different situations of book content reading.

(a) (b)

Figure 5. Page content classification. (a) The original page image. (b) The resulting image after page content segmentation and classification.



(a)

(b)

(c)

Fig 6. Different text blocks using various color numbers. (a) The original image. (b) The bi-level image. (c) The 4-color image.

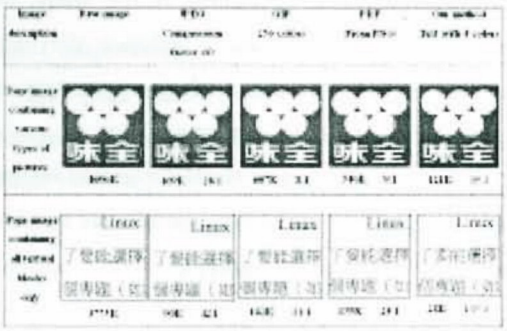


Fig 7. Comparison of compression ratios for two kinds images using 4 compression methods.

In order to obtain the overall compression rate, we tested different numbers of page images (5, 10, 20, 30, 40 pages). After digitization of these page images, Table 1 shows a comparison of the compression rates resulting from processing different numbers of page images.

7. Discussions

After observing the experimental results shown previously, some issues are discussed as follows.

Table 1. The comparison of compressing multiple pages.

Number of page images	Raw-image size (bmp)	Compressed-image size	Compression ratio
5	19,300 KB	141 KB	136:1
10	38,600 KB	247 KB	155:1
20	77,200 KB	504 KB	153:1
30	115,800 KB	714 KB	162:1
40	154,400 KB	962 KB	160:1

The compression rates depend on the contents of page images. The more textual parts in page images, the higher compression rates we can achieve. From an observation of Table 1, we can see that the compression rate of five pages is about 136:1. Following the increase of the number of pages, the compression rate gets higher. And the compression rate becomes stable when the number of pages achieves some level. The overall compression ratio is about 153:1 on average.

From the above discussions, we can see that the analysis of page contents is the key phase of the entire digitization work, especially page content segmentation. The results of segmentation will influence subsequent processing. In this study, it is assumed in our adopted segmentation method that page images are not skewed. But some scanned images are skewed. If these images are not de-skewed, the results of segmentation will not be meaningful.

In the proposed compression-by-classification approach, page contents are represented and compressed via the use of text blocks and picture blocks. If there are a large number of blocks in a page image, the proposed compression scheme will not have good compression results because header compression overhead will increase.

8. Conclusions

A book content digitization system has been successfully implemented. It contains four major parts, including analysis of page contents (segmentation and classification of page contents), recognition of repetitive patterns in pages, compression of page contents, and enhancement and display of page contents. Major achievements in different phases are summarized as follows.

(1) In the phase of analysis of page contents, a decision tree classification approach has been proposed. An algorithm has been designed to classify page contents into different types of blocks according to their color attributes. Also, a segmentation method based on a bottom-up approach and a moment-preserving color reduction algorithm [6] have been integrated successfully into the page contents analysis system. This phase is the basis of the following phases.

(2) In the phase of recognition of repetitive patterns in pages, algorithms for repetitive pattern detection and verification have been proposed. These algorithms can be used to extract common parts in different book pages to improve compression rates.

(3) In the phase of compression of page contents, a content-based compression scheme has been proposed. Based on the information of page-content classification result, appropriate compression algorithms are used to compress picture blocks effectively according to their image types. Besides, in order to keep a balance between the legibility and the compression rate, the strategy of display by two bits per pixel (4 shades of gray) is adopted in the compression of text blocks. The recognized repetitive patterns are also compressed in this phase.

(4) In the phase of display of page contents, page content enhancement and a user interface for reading have been proposed. Enhancement of page contents is used to improve the legibility of page contents to allow users to read the detail of book content clearly. Also, a user-friendly interface is provided to let users browse book contents in a convenient way.

In summary, the goal of designing an offline automatic book content digitization system has been achieved. The system has also provided a proper approach for compressing book contents efficiently. The experimental results have revealed the feasibility of the proposed system.

References

[1] G. Story, L. O'Gorman, D. Fox, L. Shaper, and H. Jagadish, "The RightPages image-based electronic library for altering and browsing," *IEEE Computer*, 25(9): 17-26, 1992.

[2] T. Phelpsand R. Wilensky, "Towards active, extensible, networked documents: Multi-valent architecture and applications," *Proc. 1st ACM International Conference on Digital Libraries*, p100-108, 1996.

[3] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*, Van Nostrand Reinhold, New York, 1994.

[4] Patrick Haffner, Leon Bottou, Paul G. Howard, Patrice Simard, Yoshua Bengio and Yann Le Cun, "Browsing through High Quality Document Images with DjVu," *Proceedings of IEEE International Forum on Research and Technology Advances in Digital Libraries*, 1998, pp. 309-318.

[5] C. Y. Chan and W. H. Tsai, "A Bottom-Up Approach to Color Image Document Analysis and Rearrangement," *Technical Report, Department of Computer and Information Science, National Chiao Tung University*, pp. 7-42, June 1999.

[6] S. S. Huang and W. H. Tsai, "Enhancement, Clipping, and Rearrangement of Color Document Images," *Technical Report, Department of Computer and Information Science, National Chiao Tung University*, pp 12-26. June 1998

[7] JBIG. Progressive bi-level image compression. ITU recommendation T.82, ISO /IEC International Standard 11544, 1993.

[8] JPEG. Digital compression and coding of continuous tone still images - requirements and guidelines. ITU recommendation T.81, ISO/IEC International Standard 10918-1,1993.

[9] W. H. Tsai (1985). "Moment-preserving thresholding: a new approach," *Computer Vision, Graphics, and Image Processing*, Vol. 29, No. 3, pp. 377-393.