DOI: http://dx.doi.org/10.26483/ijarcs.v9i2.5520

ISSN No. 0976-5697

Volume 9, No. 2, March-April 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

EFFICIENT COMPRESSION OF BINARIZED TAINTED DOCUMENTS

Ratnakumari Challa
Dept. of CSE
RGUKT, IIIT-AP, RK Valley,
AP, India

Kanusu Srinivasa Rao
Dept. of Computer Applications
YV University, Kadapa
AP, India

Abstract: Tainted documents are degraded or ruined documents of low quality and of worn out look. The taintations are like disparity variation, smear, ooze, uneven illumination. To enhance the visual quality of the tainted document binarization technique is applicable. Binarization can binarize all the tainted documents but performing binarization to faultily tainted documents is a complicated task, the complication is observed in the identification of variations between the document background and text foreground. The system uses OTSU binarization that can binarize any kind of taintations. The proposed technique addresses the variations between background and foreground text of the document and calculates the optimum threshold separating the two classes so that their combined spread is minimal or equivalent hence the vision quality increases. Enhancement of vision quality also results in the enhancement of document size. Compression is performed on the binarized tainted document to reduce the tainted document size. The compression technique projected to use in this paper is Run Length coding which helps to reduce the size of the tainted document. Run Length coding is lossless compression technique which is very successful in dealing with binary images.

Keywords: Binarization; Taintation; Compression; Run Length Coding;

I. INTRODUCTION

Image processing is the rapid developing technology in the recent technical era. Compared to the initial stage of image processing the current stage is extensively improved. Immediate result of this improvement is: it became a domain to reconstruct, reproduce and reinvent many applications. Image segmentation is a set of fragments that collectively cover the entire image, or a group of outlines extracted from the image. In the process of improving degraded documents binarization or segmentation is the suitable approach to enhance the visual quality of any image[1, 9, 10]. Generally, segmentation is process of separating an image into contours corresponding to objects. The objects or segmentation regions are usually separated by identifying the common properties. For example, the pixels in the same region may have the same intensity. So the natural way of separating fore ground objects or regions from the background is through Thresholding process, i.e. the separation of low intensity and high intensity regions. Thresholding process takes gray scale images and process the every pixel of them to create the binary images. It sets the all the pixels whose intensity value below some given threshold to zero and all pixels whose intensity value above that threshold to one [2].

The system uses Tainted document as its input. Taintations of the document deals with disparity variations, smear, bleed of ink, corrosion, aging and uneven illuminations. Hence image binarization is performed as the preprocessing step for analysis of the document that is tainted. It aims to segment or separate the foreground text from the document background.

There exists many systems that used numerous binarization techniques but there exists some difficulties within them. Namely analysis based binarization method this method uses global thresholding. The difficulty in using this global thresholding is it may cause some parts to be brighter and some parts to be darker. The number of misclassified pixels will be less so that we avoided this method. Another binarization is entropy based method, this method calculates the optimum threshold separating the two classes so that their combined spread is maximum but our system requires combined spread minimum. Other binarization approach is image variance but this method cannot binarize stained documents that is, not suitable for all kinds of documents. The binarization method called threshold based on image contrast only deals with the uneven illumination type documents by determining local thresholds.

Adaptive image contrasting is finer among the other techniques because it is the combination of local image contrast and local image gradient which overcomes the previous challenges. But for documents including big patterns and pictures this algorithm is not perfectly suitable. Hence OTSU Binarization is used in the scheme. OTSU binarization can binarize any kind of tainted document. OTSU binarization addresses the variations of document background and foreground text and calculates the optimum threshold separating the two classes so that their combined spread is minimal or equivalent[7]. This can binarize any kind of taintations. It is also called as automatic thresholding. By demonstrating superior performance against well-known binarization techniques, it is proposed to use OTSU Thresholding.

Binarized document is the resultant of binarization which is the collection of neighboring colors in larger areas. The colors include black and white. This similar pattern repetitions in larger observations motivated to compress the redundancy by using a technique called Run Length coding. The other motivation for compression is to reduce the size because after binarizing the document, size increases in most of the cases[8]. So compression using Run Length coding (RLC) is used to reduce the size. The reason behind using this RLC as it works well for binary images [11,12].

First find the size of the input document. Read the pixel values from the beginning. If the current pixel value is same as next pixel value then increment the count otherwise the value is stored in string array. Now the count value is read for the first row again this process until all the rows in the document are read completely. The result is stored string array to do this we used pdf writer from the java package com.itextpdf.text.pdf.PdfWriter. Final compressed image is stored in path specified in the class file. Compression effectiveness depends on input. To maximize the compression, document should contain consecutive runs of values. It should only be used in situations where we know for sure have repeating values. Compression technique used in the proposed scheme follows the coding strategy which involves iterations and storage mechanism.

II. RELATED WORKS

Analysis and comparison of algorithms for lossless data compression by Anmol Jyot Maan [3], gave the idea why to compress and what are the compression techniques. The work conveys that data compression is very important and is an art to reduce the size of data. The main goal of data compression is to reduce the size of the file by eliminating the redundant content from the file. Data compression is very useful to reduce the space required to store the data and band width and time needed to transmit the data. Data compression can either be lossless or lossy. In lossless data compression, the exact copy of the original data can be recreated from the compressed data. In lossy data compression, perfect original data cannot be regenerated from the compressed data. Using some inexact approximations and partial data eliminations, data can be recreated which is not exactly same as original data. This is mainly used for compression of multimedia files like audio, video or images.

To compress the oversized binarized tainted document it is recommended to use digital image compression techniques by Gomathi.K.V1, Lotus.R2[13] and many compression techniques are proposed. RUN LENGTH CODING (RLC) – compression technique is chosen as it suits to the requirement. RLC is an entropy encoding compression technique that works on inter pixel redundancy.

III. PRILIMINARY METHODS

A. OTSU Method

Scheme proposed here for binarization is based on OTSU method. Otsu's thresholding technique finds the threshold for the each image that minimizes the weighted within-class variance, i.e it can identify the pixels which can fall under the same region or intra-region. And it turns to be the same as maximizing the between-class variance, i.e it can separate the pixels of the different regions or inter regions (foreground and background) of the image.

The formula for weighted within class variance is:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

Where $q_1(t)$ and $q_2(t)$ are class probabilities. $q_1(t)$ and $q_2(t)$ are estimated as:

$$q_1(t) = \sum_{i=1}^{t} P(i) \qquad q_2(t) = \sum_{i=t+1}^{I} P(i)$$

The class means $\mu_1(t)$ and $\mu_2(t)$ are given by

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \qquad \mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)}$$

The individual class variances

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)}$$

and

$$\sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

Now, it is necessary to do and run through the gray scale range t from 0 to 255 and select t as threshold that minimizes the intra-class variance. But the relationship between inter-class and inter class variance can be explored through the recurrent relation which can be permit much quicker computations [6].

B. Compression Method : Run-length encoding (RLE)

Run-length encoding is the simplest compression techniques for compressing the data that made of any combination of symbols. The major idea of this method is to replace the consecutive large repetitions of a symbol (runs) by a pair that constitutes with the symbol and its count i.e number of its occurrences [4]. For example: the data stream constituted with symbols w, x, y and z and its compressed form using run length encoding is given in the figure 1.

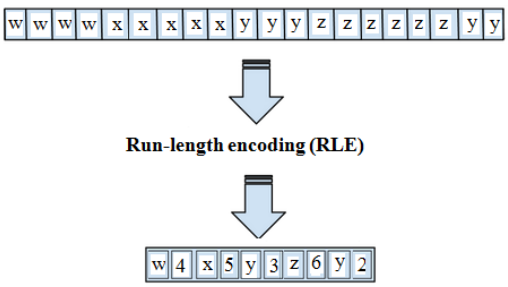


Figure 1. Example of Run Length Encoding

The method is even more efficient for the compression of the binary files which have the data with only two symbols (0 and 1) and if one symbol is more repeatedly occur than other symbol. An example for the compression of binary string is given in figure 2.

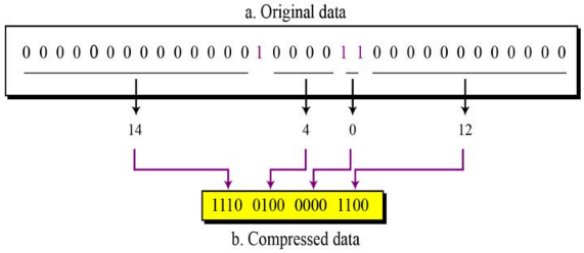


Figure 2. Run length encoding for two symbols

In the scheme, it is noted and proposed that run length encoding is more suitable and efficient to reduce the size of binarized tainted document.

IV. IDEAS AND SCHEMES

Here the system provides the detailed and simplest method of binarization and compression as shown in the figure 3. Given an input gray scale image of scanned tainted document, OTSU binarization algorithm will segment and separate the foreground text from the background. In OTSU method, for each potential threshold T, Splitting the pixels into two clusters according to the threshold, computing the mean of each cluster and squaring the difference between the means followed by multiplying with the number of pixels in one cluster times the number in the other cluster. The process continues to compute the optimal thresold that minimizes the intra-class variance and maximizes the inter-class variance.

Since it is huge work to compute for every likely threshold, but it turns out that calculations involved might dependent for one threshold value to another threshold value. First the class probabilities $q_1(t)$, $q_2(t)$ and relevant cluster means $\mu_1(t)$ *and* $\mu_2(t)$ computed and updated as pixels move from one region to another region as threshold *t* increases [5]. Using the following recurrence relations, inter-class variance can be computed and updated for each successively tested threshold:

$$q_1(t+1) = q_1(t) + q_t$$
$$q_2(t+1) = q_2(t) - q_t$$
$$\mu_1(t+1) = (\mu_1(t)q_1(t) + q_t t) / q_1(t+1)$$
$$\mu_2(t+1) = (\mu_2(t)q_2(t) - q_t t) / q_2(t+1)$$

Finally, the optimal thresold is computed which minimises intra-class variance and maximizes inter-class variance. Using optimal threshold, inputed tainted image can be binarized.

After the binarization process, compression is carried out to reduce the size of the image by eliminating redundant elements (runs) as shown in the figure 3. Elimination of runs can be done by replacing with set of two values. One value refers to the original pixel value i.e., repeated pixel value (0 or 1) in the image and second value refers to the count of the repeated value [4].

The implementation of the compression using RLE is done pixel by pixel of the input image of binarized tainted document. Read the pixel value of the source image and initialize the pair of (pixel value , count) and set its count to 1. Then proceed to the next all pixel by considering the next is pointed by index j. Now, increment its count by 1 if the pixel value is same as the previous pixel value (pointed by j-1). Otherwise, if next pixel pointed by j is not same as the pixel pointed by j-1 then create a new pair of pixel value and count with initial value 1. The process repeats for every pixel till the end of the last pixel of the image.

When the main focus is to consider the time complexities of the algorithm accordingly find the fastest implementation. But here in the proposed method, it is not much important to do quick data compression rather focus is more to reduce the size as much as possible without data loss [13].

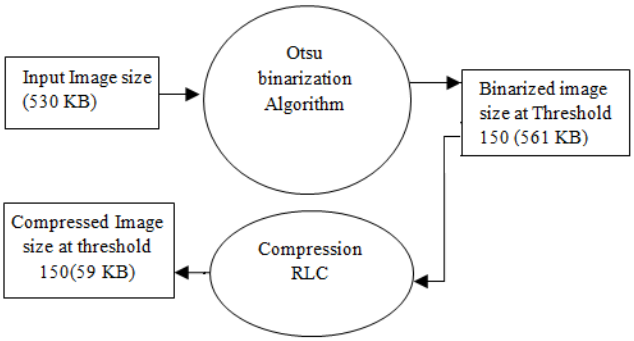
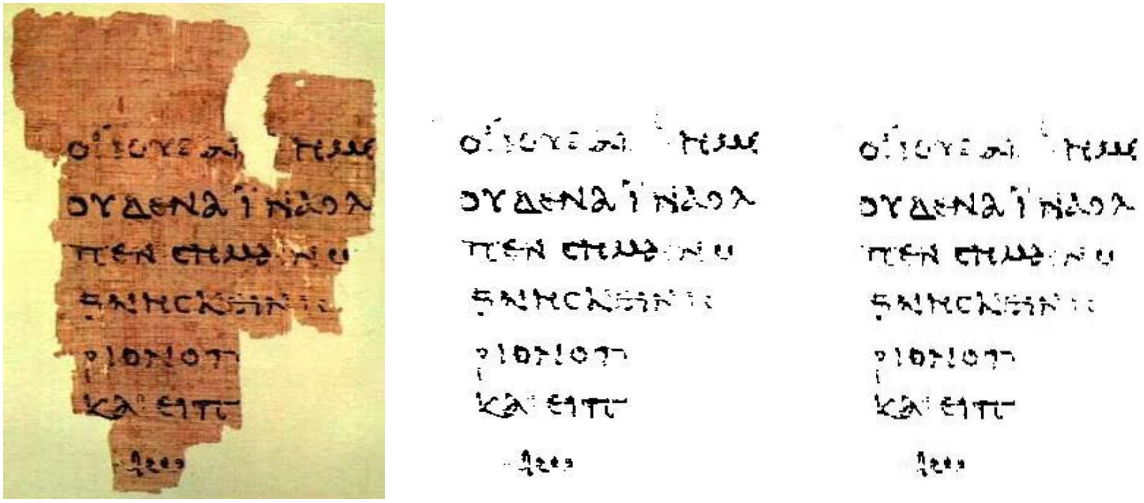


Figure 3. Scenario of the Scheme



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

Figure 4. (a) Input Tainted Document (b) Binarized document (c) Decompressed Binarized document

Table 1. Comparative study on binarization of tainted documents at different thresholds and its compression

Image name	Original Size	Size of Image in KB after binarization using Threshold (T)					Size of the corresponding Compressed Image (in KB)				
		T50	T100	T150	T200	T250	C50	C100	C150	C200	C250
5	434	195	227	471	186	279	62	73	130	59	76.3
6	530	146	170	561	204		55	135	59	63.4	
31	182	46.3	164	314	308		66	110	121	116	
35	88.3	3.13	21.2	28.3	29.4	29.5	28	63	84	85	87
37	339	137	156	167	237		87	99	104	136	
38	261	21.4	64.6	105	60.4	111	42	75	116	68	122
41	433	31.2	116	134	88.1	133	53	101	113	82	111
47	94.1	60.7					65.1				
49	172	62.6					90.6				
51	191	16.4					27.3				
53	128	135					58.3				
54	743	306					64.4				
56	151	51.1					109				
60	37.4	13.4					30.4				
62	180	74.6					69.4				
63	144	62.5					94.3				

V. DISCUSSION

Consider an image of size 530KB which contains 1415000 pixels this binarized at different threshold levels i.e., at Threshold 50, 100, 150 and 250 the image size after binarization is 146KB, 170KB, 561KB and 204KB respectively. The size enlargement is observed after binarization in some cases. So to compress the binarized image the system implemented Run Length Coding. The size of the image at threshold 150 is 561 KB after compression the size of the image is 59 KB as shown in the figure 4. The compression yields good results if and only if image contains less data. The comparative study of different tainted documents named by numbers 5 to 63 is presented in Table 1. These documents are binarized at different threshold levels 50, 100, 150 and 250. At the same time along with their same binarization the document is compressed at the same threshold level. The entire rows in the table specify positive test case since the aim of proposed system is satisfied. T stands for Threshold level and C stands for compression at each threshold level. The empty rows indicate that the content of the tainted document is not visible at the respective threshold levels so there is no use of compressing invisible tainted document.

VI. CONCLUSION

The system boldly conveys that it involves the binarization method that can enhance vision quality of any kind of taintations and the proposed technique requires less effort and overhead for developing and using. Compression technique is also simple and easy implementable. The compression and binarization does not require any external software. By using the proposed strategies, storage space required for storing the tainted images can be minimized and transmission rate can be maximized. From the experimental analysis of the proposed system, compression of all the tainted documents is not possible but it is successful in most of the cases like documents containing less data. Analysis of the results shown that the scheme is more suitable for the text files which contain lots of spaces for indenting and line-art images that contain large white or black areas. It is observed that the run-length algorithm is very useful on large sequences of repeating elements, no matter characters or array items. Hence the combination of binarization and compression gives rise to wide range of applicability.

VII. REFERENCES

[1] Bolan Su, Shijian Lu, and Chew Lim Tan, Senior Member of IEEE: Robust Document Image Binarization Technique for Degraded Document Images. IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 22, NO. 4, APRIL 2013.

[2] Hafizan Mat Som1, Jasni Mohamad Zain2 and Amzari Jihadi Ghazali3: Application of Threshold Techniques for Readability Improvement of Jawi Historical Manuscript Images by. Advanced Computing: An International Journal (ACIJ), Vol.2, No.2, March 2011.

[3] Anmol Jyot Maan: Analysis and Comparison of Algorithms for Lossless Data Compression. International Journal of Information and Computation Technology. ISSN 0974-2239 Volume 3, Number 3 (2013), pp. 139-146.

[4] S. Sarika, S. Srilali: Improved Run Length Encoding Scheme For Efficient Compression Data Rate. Journal of Engineering Research and Applications , ISSN : 2248-9622, Vol. 3, Issue 6, Nov-Dec 2013, pp 2017-2020.

[5] Automatic Thresholding document from http://www.math.tau.ac.il/~turkel/notes/otsu

[6] Otsu: A Threshold Selection Method from Gray-Level Histograms and Otsu Thresholding. IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, pp. 62-66, 1979.

[7] Boston Cigan's:Java image binarization using Otsu's algorithm. http://developer.bostjan-cigan.com/java-image-binarization

[8] Tarnjot Kaur Gill, Document Image Binarization Techniques- A Review. International Journal of Computer Applications 98(12):1-4, July 2014

[9] Aroop Mukherjee and Soumen Kanrar, Enhancement of Image Resolution by Binarization, International Journal of Computer Applications, Volume 10 – 10, 2010.

[10] Arwa Mahmoud AL-Khatatneh, Sakinah li Pitchay and Musab Kasim Al-qudah, Compound binarization of Degraded Documents, ARPN Journal of Engineering and Applied Sciences, Vol. 10, NO. 2, ISSN 1819-6608, 2015.

[11] Stoimen: Data Compression with Run-length Encoding Computer Algorithms http://www.stoimen.com/blog/.

[12] RLE Compression http://www.prepressure.com/library/compression-algorithm/rle

[13] Gomathi.K.V1, Lotus.R2: Digital Image Compression Techniques, IJRET: International Journal of Research in Engineering and Technology eISSN: 2319-1163, pISSN: 2321-7308 Volume: 03 Issue: 10, Oct-2014, pp.285-290.