

International Conference on Computational Modeling and Security (CMS 2016)

Visualizing CCITT Group 3 and Group 4 TIFF Documents and Transforming to Run-Length Compressed Format Enabling Direct Processing in Compressed Domain

Mohammed Javed ^{*a}, Krishnanand S.H. ^a, P. Nagabhushan ^a, B. B. Chaudhuri ^b

^a*Department of Studies in Computer Science, University of Mysore, Mysore, India*

^b*CVPR Unit, Indian Statistical Institute, Kolkata, India*

Abstract

Compression of data could be thought of as an avenue to overcome Big data problem to a large extent particularly to combat the storage and transmission issues. In this context, documents, images, audios and videos are preferred to be archived and communicated in the compressed form. However, any subsequent operation over the compressed data requires decompression which implies additional computing resources. Therefore developing novel techniques to operate and analyze directly the contents within the compressed data without involving the stage of decompression is a potential research issue. In this context, recently in the literature of Document Image Analysis (DIA) some works have been reported on direct processing of run-length compressed document data specifically targeted on CCITT Group 3 1-D documents. Since, run-length data is the backbone of other advanced compression schemes of CCITT such as CCITT Group 3 2-D (T.4) and CCITT Group 4 2-D (T.6) which are widely supported by TIFF and PDF formats, the proposal in this paper is to intelligently generate the run-length data from the compressed data of T.4 and T.6, and thus extend the idea of direct processing of documents in Run-Length Compressed Domain (RLCD). The generated run-length data from the proposed algorithm is experimentally validated and 100% correlation is reported with a data set of compressed documents. In the end, text segmentation and word spotting application in RLCD is also demonstrated.

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of the Organizing Committee of CMS 2016

Keywords: Run-length compressed domain processing, Run-length data, Modified Huffman(MH), Modified Read(MR), Modified Modified Read(MMR)

* Corresponding author: Mohammed Javed, Tel.: +919741161929;
E-mail address: javedsolutions@gmail.com

1. Introduction

In today's digital era, data compression is the technique generally employed to overcome the volume aspects of the Big data. In fact, on daily basis this results in large scale of compressed data being stored and transferred in the compressed formats. On the contrary, as generally witnessed, any operation or analytics over the compressed data is executed after decompression. If this reversing stage of decompression could be avoided and the analytics could be carried out directly in the compressed version, then it will be an additional breakthrough. Towards this, deeper understanding of the nature of the compression would provide some useful clues. Recently, this novel idea of operating directly over the compressed data has attracted many researchers and as a result latest books and research papers on compressed domain techniques^{1,2,3,4} on texts, images and videos have been published. The Document Image Analysis (DIA) community is yet to gain thrust in the area.

In the literature of DIA, there have been a few initial attempts to explore the possibility of operating directly over the compressed formats such as CCITT Group 3^{1,5,6,7} CCITT Group 4^{3,8,9} JPEG⁴ and JBIG¹⁰. However, the proposed methods and operations are limited to a particular compressed format. In the recent literature, lot of interesting and deeper works like feature extraction^{1,5,11}, page segmentation^{6,12}, text segmentation^{1,7}, font size detection¹¹, etc have been reported on the run-length compressed data of CCITT Group 3 1-D compressed documents in Run-Length Compressed Domain (RLCD). Incidentally, the other advanced compression schemes of CCITT are also based on Run-Length Encoding (RLE) technique. Based on the variations in the RLE encoding process, CCITT (International Telegraph Telephone Consultative Committee) has introduced a series of compression standards and transfer protocols for black and white images over telephone lines and data networks^{13,14}. They are popularly known as CCITT Group 3 1-D (MH-Modified Huffman), CCITT Group 3 2-D (MR-Modified Read) and Group 4 2-D (MMR-Modified Modified Read). These compression algorithms are widely supported by TIFF and PDF formats for handling printed and handwritten text documents. CCITT Group 3 contains synchronization codes and hence was developed for network communications, whereas CCITT Group 4 was designed for archival purpose, applicable in large databases because of its high compression ratio. Overall, it can be observed that the run-length compressed data is the backbone of CCITT compression schemes. Therefore, the proposal in the research paper is to extend the idea of directly operating on compressed documents in RLCD to advanced compression schemes like MR and MMR by intelligently generating run-length code. Towards this purpose, a novel algorithm is proposed in this paper.

In this backdrop, the proposed research paper aims at (i) getting deeper understanding of the compressed data of RLE flavored advanced compression schemes like MH, MR and MMR of CCITT, (ii) transforming the compressed data of MR and MMR to Run-length data, and (iii) demonstrating direct operations and analytics on the generated run-length compressed data.

Rest of the paper is organized as follows. Section 2 is dedicated for discussing background information related to this research work such as TIFF data format, MH, MR, MMR encoding schemes from the perspective of compressed domain processing. Section 3 demonstrates visualization of TIFF compressed data, section 4 introduces the novel algorithm of transforming MR and MMR compressed data to run-length data and subsequently discusses the Run-Length Compressed Domain processing. Section 5 reports experimental results and section 6 summarizes the research work.

2. Background

2.1. Structure

TIFF¹⁵ is a graphical format which stands for Tagged Image File Format and a typical TIFF file organization is shown in Fig-1. In the figure IFH stands for Image File Header, Bitmap data actually contains the black and white pixels data either in raw or compressed form, IFD stands for Image File Directory, and EoB indicates End of Byte.



Fig. 1. File organization of TIFF

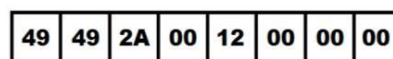


Fig. 2. Image File Header

A TIFF file always begins with an 8-byte IFH that points to an IFD which is shown in Fig-2. In the figure, the first two bytes indicate the byte order, where 4949H in hexadecimal notation represents little-endian and 4D4DH

indicates big-endian order. The next two bytes 002AH identifies the file type which in this case is a TIFF file. The last four bytes 00000012H indicate the offset value of the first IFD in bytes. An IFD inside a TIFF file gives information about the specific tag associated with the image. This specific tag information will be used during the decoding process. A general structure of an IFD is shown in Fig-3, which is made up of different fields totally constituting of 12 byte data. There are nearly 14 IFD's¹⁵ associated with a TIFF image. Image Width, ImageLength, Compression, X Resolution, Y Resolution are few examples of TIFF IFD's. A sample IFD for the tag ImageWidth is shown in Fig-4.



Fig. 3. General Image File Directory

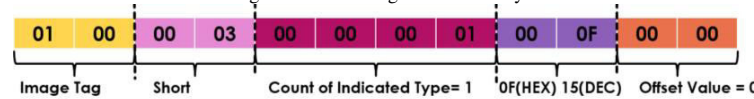


Fig. 4. IFD for the tag Image-Width

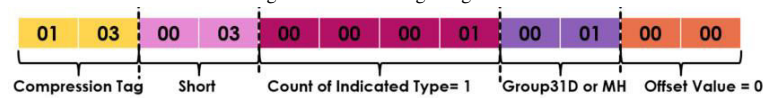


Fig. 3. IFD for an image without compression

Further, the IFD tag Compression in a TIFF file internally supports different compression algorithms, and based on the type of data the compression algorithms are selected. The type of compression employed to the image data is identified by a special tag number 0103H. A general IFD structure for the Compression tag is given in Fig-3. In the figure, the byte numbers 9 and 10 indicate the type of compression algorithm used. In Fig-3, the value 0001H indicates 'No Compression', whereas the presence of values such as 0002H, 0003H and 0004H respectively indicate MH, MR and MMR compression schemes. Therefore based on the compression scheme indicated in the Compression tag, the compressed data is utilized for the proposed compressed domain processing.

Text contents are very common in documents such as research articles, newspapers and magazines. Moreover, text carries important information of the document and can be losslessly represented and reproduced using a black and white image. To compress the contents of black and white images, the popular image compression formats like TIFF and PDF widely support three compression schemes namely MH, MR and MMR which represent different flavors of RLE. Therefore the upcoming subsections are dedicated to discuss the working model of MH, MR and MMR from the perspective of compressed data processing. The study presented here is to project the presence of RLE backbone in these compressed formats and hence to get an avenue to transform these codes into RLCD. However, a detailed discussion regarding the compression schemes is available in the works^{13,14,15}.

Modified Huffman (MH) or CCITT Group 3 1-D encoding is a variation of the Huffman compression algorithm¹³. A binary image is made up of a series of black and white pixel runs of variable lengths. The MH encoder scans the black(0) and white(1) pixel runs line by line and outputs a variable-length binary code word representing the run-length and run-color from a standard predefined Huffman table. This standard table representing runs of black and white pixels is part of the T.4¹³ specification. The table is used for encoding and decoding all CCITT Group 3 data. The output code word is normally shorter than the input pixel data, and hence the compression is achieved.

2.2. Modified Huffman (MH)

In MH encoding each run length is encoded using two codes, namely Makeup and Terminating codes. Every encoded pixel run is a combination of zero or more Makeup code words and subsequently followed by a Terminating code word. The shorter runs are represented by Terminating code words and the longer runs by Makeup code words. There exist separate predefined tables for Terminating and Makeup code words for both black and white runs. Pixel runs with a length varying from 0 to 63 are represented using a single Terminating code word. Runs with a length between 64 to 2623 pixels are encoded using a single makeup code and a terminating code. Runs

with a length greater than 2623 pixels will be encoded using one or more makeup codes and a terminating code. The overall run length is the sum of the run-length values encoded by each code word.

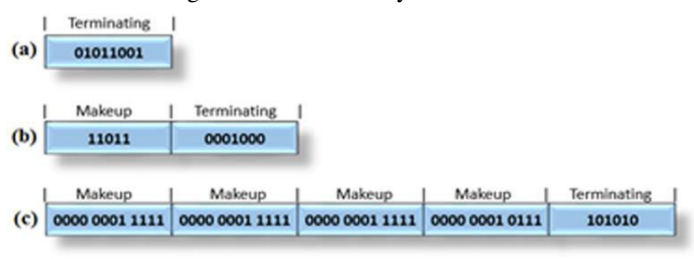


Fig. 4. Modified Huffman encoding

Consider the examples shown in Fig-4. A run of 22 black pixels will be encoded by the terminating code for a black run length of 22 (code word 01011001 obtained from standard table¹⁵). This reduces a 22-bit run to the size of an 8-bit code word, a compression ratio of 3 : 1. This is illustrated in Fig-6a. Further a white run of 84 pixels will be represented using the makeup code for a white run length of 64 pixels followed by the terminating code for a white run length of 20 pixels ($64 + 20 = 84$). This encoding reduces 84 bits to 12 bits, or a compression ratio of 7 : 1. This is illustrated in Fig-6b. A run of 10000 white pixels would be encoded as three makeup codes of 2560 white pixels (7680 pixels), a makeup code of 2304 white pixels, followed by the terminating code for 16 white pixels ($2560 + 2560 + 2560 + 2304 + 16 = 10000$). In this case 8800 run-length bits are encoded into five code words with a total length of 54 bits, for an approximate compression ratio of 185 : 1. This is illustrated in Fig-4c.

In MH encoding process, all scan lines are conventionally designed to always begin with a white run-length code word (because in most of the document images scan lines begin with a white space or run). In case a scan line begins with a black run, a white run-length code word of zero length will be added at the beginning of the actual scan line code. An EOL stands for End Of Line code, which is a 12-bit code word that begins every line in a Group 3 transmission. This code word is used to identify the start or end of a scan line during the transmission stage. The decoder uses EOL codes to detect the width of a decoded scan line, and also to keep track of the number of scan lines in an image. This is because if any short image is detected, it pads the remaining length with scan lines of all white pixels.

Further, RTC (Return To Control) code is used to terminate Group 3 message transmissions and is added to the end of every Group 3 data stream. The RTC code signal consists of simply six consecutive EOL's and this indicates the end of message transmission. The RTC signal is not actually the part of the encoded message data but actually part of the facsimile protocol. A FILL code word is a run of one or more zero bits that appear between the encoded scan line data and the EOL code. The FILL bits help to pad out the length of an encoded scan line to compensate the transmission time of the line to a required length.

2.3. Modified Read (MR)

The MR or CCITT Group 3 2D coding scheme is a line-by-line coding method. The important definitions associated with MR coding scheme are reproduced below¹³.

- Changing element : In a scan line, an element whose color is different from that of the previous element
- Reference element : An element whose position determines a coding mode
- Coding mode : A method to code the position of each changing element along the coding line
- Coding line : The current scan line
- Reference line : The previous scan line

In MR compression scheme¹³, the position of each changing element on the coding line is encoded with respect to the position of a corresponding reference element. The reference element may be located either on the coding line or on the reference line. After the coding process is over, the current coding line becomes the reference line for the next coding line. In MR, to limit the facsimile transmission error, a MH coded line is generally sent at regular intervals which are referred to as K factor. For a standard facsimile, the value of K is equal to 2, and at the higher

resolution K is equal to 4. The value of K for a digital image can be of any positive non-zero integer. MR compression scheme implements Group 3 encoding without using the EOL or RTC code words. Also while writing Group 3 data to an image file, the initial 12-bit EOL, the 12 EOL bits per scan line, and the 72 RTC bits affixed onto the end of each image are not used. Overall, for every K lines, the CCITT Group 3 2-D scheme encodes first line in 1-D MH coding and the other $K - 1$ lines in 2-D coding.

In the CCITT Group 3 2-D¹³ coding there are 5 changing elements defined which are given below,

a0: the reference element on the coding line

a1: the next changing element to the right of a0 on the coding line.

a2: the next changing element to the right of a1 on the coding line.

b1: the next changing element on the reference line to the right of a0 and of inverse color of a0.

b2: the next changing element to the right of b1 on the reference line.

At the beginning of coding process, the changing element a0 is first set on imaginary white changing element located just before the first element on the coding line. During the encoding process, the position of a0 is determined by the previous coding mode. The changing elements for a sample reference and coding line is shown in Fig-7.

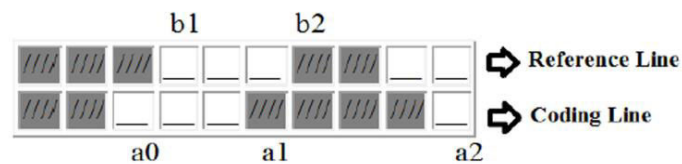


Fig. 5. Changing elements in MR/MMR encoding

In CCITT T.4 standard, there are 3 coding modes: Pass Mode (P), Vertical Mode (V), and Horizontal Mode (H). Based on the location of a changing element along the coding line, the appropriate coding mode is selected. Pass Mode: when the position of b2 lies to the left of a2. Vertical Mode: when the relative distance between a1 and b1 is less than or equal to 3. Horizontal Mode: when neither pass mode nor vertical mode occur. In the Vertical Mode, depending on the relative distance between a1 and b1, there are seven possible cases. The V(0) implies a1 just under b1, VR(1) indicates that a1 is one pixel to the right of b1. Similarly other cases are shown in Table-1. In the table M(ai-aj) represents the code words of 1-D compression standard for the run ai-aj.

2.4. Modified Modified Read (MMR)

The CCITT Group 4 2-D¹⁴ coding scheme is known as the Modified Modified Relative element address designate code (MMR). The coding process in MMR is similar to MR, where the position of changing element along the coding line is encoded with reference to the position of a corresponding reference element which may be on coding line or the reference line. The reference line is present immediately above the coding line. The current coding line becomes the reference line after the coding process of the current coding line. The reference line for the first scan line of a page is an imaginary white line. Overall, the coding scheme is very much similar to MR, except that the MMR does coding of the first line differently and unlike MR it avoids coding of every K^{th} line ($K = 2$ or 4) of image data in MH mode.

Table 1: Standard Reference table for Vertical coding mode¹³

Mode	Coding Elements		Code Word
Pass P	4	4	3
Horizontal H	4	4	3
Vertical: V(0)	a1 just under b1	a1b1 = 0	1
VR(1)		a1b1 = 1	011
VR(2)		a1b1 = 2	000011
VR(3)		a1b1 = 3	0000011
VL(1)		a1b1 = 1	010
VL(2)		a1b1 = 2	000010
VL(3)		a1b1 = 3	0000010
Extension	2-D(Extensions)		0000001xxx
	1-D(Extensions)		00000000xxx

3. Visualizing TIFF Compressed Data

Data visualization¹⁶ is the presentation of data in a pictorial or graphical format, so that the nature of data is easily understood, analyzed and interpreted. In this section, the visualization of TIFF compressed data is demonstrated. A sample binary image pattern consisting of 5 rows and 16 columns is shown in Fig-8. It has five scan lines which are made up of black and white pixels.

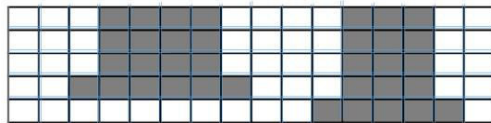


Fig. 6. Sample black and white image

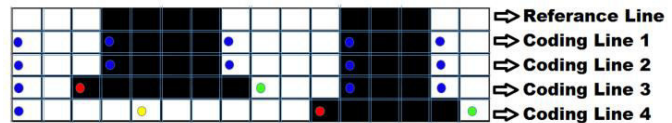
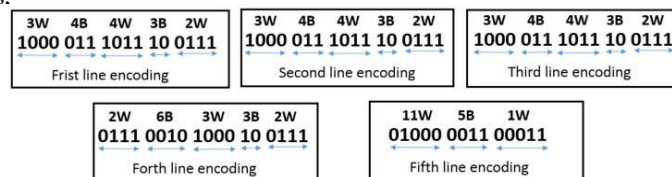
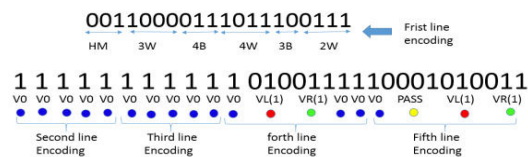


Fig. 7. Changing elements in the sample binary image with Blue, Red, Green and Yellow colors respectively indicate MR/MMR coding modes V(0), VL(1), VR(1) and Pass mode

The MH coding technique reads line by line the black and white pixel runs present in the image and encodes them using the standard Huffman table¹³. The MH compressed data generated line by line for the sample image shown in Fig-6 is given as follows.



For the sake of illustrating MR/MMR coding technique, all the changing elements with respect to the reference line (first scan line) in Fig-6 are marked and shown in Fig-7. From the figure, it can be observed that every changing element in the coding lines (in scan lines 2, 3 and partially in 4) are underneath or at least very close to that of the reference line. Therefore these positions will be encoded with a Vertical Mode. In Vertical Mode, the relative positions are within the proximity of three pixels with each other, and hence this type of encoding is most commonly occurring in a document image. Specifically, the case where the positions of the changing pixels are identical, known as V(0) is encoded using a single bit (the other options being VL(3) to VR(3) in Table-1). It can be observed that a vertical line of any thickness can be coded very efficiently. This interesting pattern is observed in bar-codes and has been used for automatic detection of bar-codes by¹⁷. On the other hand, when the bottom of the image is detected Pass Mode is encountered. This is because the position of the changing pixel is very different to that on the line above. This implies skipping of two changing pixels, to black and back to white on the line above. These interesting features were explored by Lu and Tan^{3,8} for word searching and document retrieval purpose, for simulating an OCR by⁹, for skew detection by¹⁸, for document similarity and equivalence by^{19,20}. The positions of changing pixels which are not in close proximity to those above are encoded in pairs using the Horizontal Mode of Group 3 encoding mechanism. The MR compressed data generated for the sample image in Fig-7 using MR(K=5) code is given below,



The equivalent Hexadecimal code for the above MR(K=5)/MMR compressed data is given as 87 73 80 87 73 80 87 73 80 72 89 C0 39 8E

3.1. Binary Viewer

When the sample image shown in Fig-6 is compressed with TIFF format, the compressed data is generated and it can be visualized as shown in Fig-8, using a Binary Viewer Software²¹. The file header, compressed data and the tags in the compressed format are clearly marked in Fig-8. The compressed data is shown within a Red colored Manhattan layout. The other layouts in Blue, Yellow and Pink colors indicate, the file header, the tags associated with the file and end of file.

Hexadecimal (1 Byte)

49	49	2A	00	16	00	00	00	87	73	80	87	73	80	87	73
80	72	89	C0	39	8E	0E	00	00	01	03	00	01	00	00	00
10	00	00	00	01	01	03	00	01	00	00	05	00	00	00	00
02	01	03	00	01	00	00	00	01	00	00	03	01	03	00	00
01	00	00	00	02	00	00	00	06	01	03	00	01	00	00	00
00	00	00	00	11	01	04	00	01	00	00	08	00	00	00	00
12	01	03	00	01	00	00	00	01	00	00	15	01	03	00	00
01	00	00	00	01	00	00	00	16	01	03	00	01	00	00	00
00	02	00	00	17	01	04	00	01	00	00	0E	00	00	00	00
1A	01	05	00	01	00	00	00	C4	00	00	1B	01	05	00	00
01	00	00	00	CC	00	00	00	1C	01	03	00	01	00	00	00
01	00	00	00	28	01	03	00	01	00	00	02	00	00	00	00
00	00	00	00	48	00	00	00	01	00	00	00	48	00	00	00
01	00	00	00												

Fig. 8. Binary viewer visualization of TIFF compressed data of a sample image in Fig-8 using MR/MMR codes

3	4	4	3	2	0	0	0	0	0	0	0	0	0	0	0
3	4	4	3	2	0	0	0	0	0	0	0	0	0	0	0
3	4	4	3	2	0	0	0	0	0	0	0	0	0	0	0
2	6	3	3	2	0	0	0	0	0	0	0	0	0	0	0
10	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0




Fig. 9. (a) Run-length compressed data and (b) its equivalent image view

3.2. Image View

The other way to visualize the compressed data is to generate line by line the run-length data from the MH/MR/MMR codes and then view it as an image. Both run-length compressed data and its equivalent image view for the sample image in Fig-6 is shown in Fig-9

4. Run-Length Compressed Domain

In this section, we propose a new compressed domain model called as Run-Length Compressed Domain (RLCD) for processing the compressed document data of all the three related coding modes (MH/MR/MMR) of CCITT compression. The RLCD model is shown in Fig-10, which generates run-length data intelligently with the help of the proposed run-length data extraction algorithm (see Fig-11) and defines compressed domain operations and analytics over the generated data. In the proposed model, reiterating from the work of ^{1,5,7,11}, it can be observed in Fig-10 that carrying out decompression is avoided.

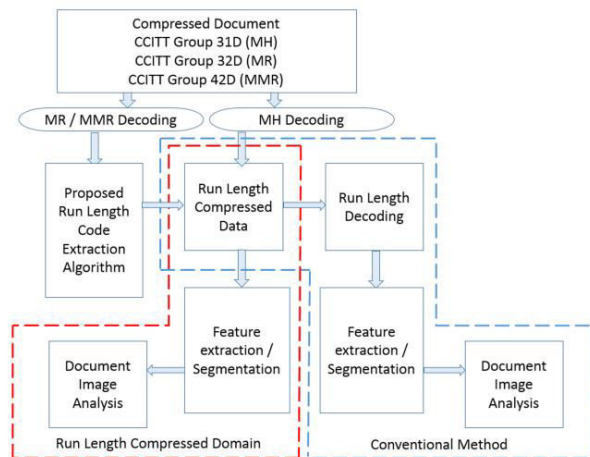


Fig. 10. Proposed RLCD model

Extracting run-length compressed data from MH codes is quite straight forward. This can be achieved by using the standard Huffman table proposed by CCITT in reverse. However, generating run-length data from MR/MMR is little tricky and is intelligently accomplished with an algorithm proposed in this paper. The proposed model of run-length data extraction from MR/MMR compressed data is shown in Fig-11. Using the method, the run length code generated stage by stage from the sample image shown in Fig-6 is tabulated in Table-2. As discussed earlier, the run-length code generation procedure for the first three rows are very straight forward. They belong to vertical mode V(0), and hence the run-length data from the reference line is directly copied to the run-length data of the current coding line. The run extraction in row number four and five are as follows, where RR is Reference Run V(0).

Row 4: RR-VL(1), RR+VL(1)+VR(1), RR-VR(1), RR, RR

Row 5: RR+Pass Mode-VL(1), RR+VL(1)+VR(1), RR-VR(1)

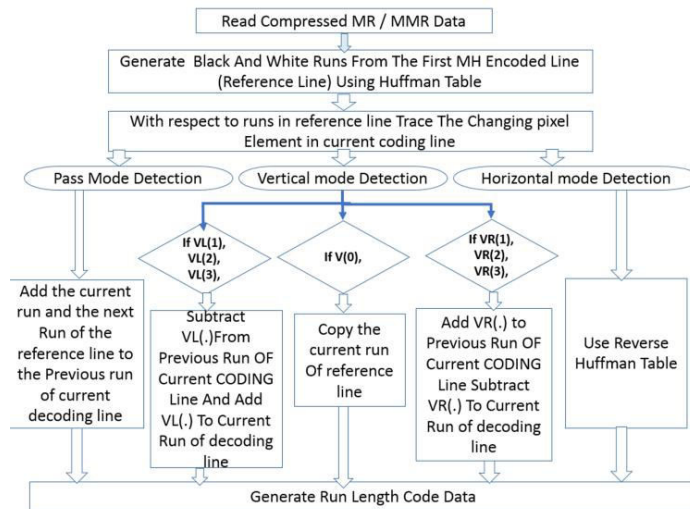


Fig. 11. Proposed run-length code extraction algorithm.

The run-length data extracted for each row in the sample image in Fig-6 is given in Table-2.

Table 2: Illustration of run-length data extraction using the proposed algorithm (In the table, W is white run and B is black run)

W	B	W	B	W
3	4	4	3	2
3	4	4	3	2
3	4	4	3	2
3-1 (VL)= 2	4+1(VL)+1(VR)=6	4-1(VR)=3	3	2
3+Pass(4+4)-1(VL)=10	3+1(VL)+1(VR)=5	2-1(VR)=1		

5. Experimental Results

In this section, we conduct experiment on MH, MR and MMR compressed documents to validate the run-length code generated using the proposed run-length code extraction algorithm. The ground truth data for the experiment is generated by directly decompressing the MH/MR/MMR compressed data to pixel data. On the other hand, the test data is generated by decompressing the run-length data extracted the proposed algorithm. The decompressed results from both test and ground truth is measured using correlation measure given below,

$$Correlation(\%) = \left[1 - \frac{\sum_{i=1}^m \sum_{j=1}^n |A_{i,j} - B_{i,j}|}{m \times n} \right] \times 100 \quad (1)$$

where m and n are the corresponding rows and columns in the test image(A)after decompression, and ground truth image(B).

Table 3. Experimental results.

TIFF Compression Algorithms	#Document	Correlation
CCITT Group 3 1-D (MH)	20	100%
CCITT Group 3 2-D (MR)	20	100%
CCITT Group 4 2-D (MMR)	20	100%

In the literature, lot of work on direct processing of run-length compressed document data have been proposed. The document image operations and analytics such as feature extraction^{1,5,11}, page segmentation^{6,12}, text segmentation^{1,7}, font size detection¹¹ using Run-Length data have been attempted. Therefore, based on the experimental results in Table-3, all these operations and analytics can be extended to work with the advanced CCITT compression schemes underscored in this research paper. One such extension is illustrated taking the application of text segmentation and subsequently word spotting⁷. The experimental results of text segmentation and word spotting for a subset of documents from the dataset of ⁷ is tabulated in Table-4 and Table-5.

Table 4. The Accuracy of Word and Character segmentation for 60 (20 documents each using MH, MR and MMR compression) compressed text documents

Segmentation	Precision(%)	Recall(%)	F-Measure
Words	97.30	96.83	97.06
Characters	96.80	87.03	91.65

Table 5. The Accuracy of Word and Character segmentation for 60 (20 documents each using MH, MR and MMR compression) compressed text documents

#Documents	Font Style	#Keywords	Precision(%)	Recall(%)	F-Measure
60	Times New Roman	30	100	59.10	74.29
60	Arial	30	100	67.61	80.67
60	Calibri	30	100	68.15	81.05

6. Conclusion

This research paper a novel idea of carrying out document image analysis directly in Run-Length Compressed Domain (RLCD) which is capable of handling the compressed data of CCITT Group 3 1-D and 2-D, and CCITT Group 4 2-D is proposed. This is accomplished by an algorithm that intelligently extracts the run-length data from T.4 and T.6 compressed TIFF documents and subsequently extends the proposed model to advanced compression schemes of CCITT. The compressed data generated from the proposed algorithm were experimentally validated using correlation measure.

References

1. M. Javed, P. Nagabhushan, and B. B. Chaudhuri, "Extraction of projection profile, run-histogram and entropy features straight from run-length compressed documents," ACPR, pp. 813–817, November 2013.
2. J. Lu and D. Jiang, "Survey on the technology of image processing based on dct compressed domain," ICMT, pp. 786–789, 2011.
3. Y. Lu and C. L. Tan, "Word searching in ccitt group 4 compressed document images," ICDAR, pp. 467–471, 2003.
4. J. Mukhopadhyay, Image and Video Processing in Compressed Domain. Chapman and Hall/CRC, 2011.
5. M. Javed, P. Nagabhushan, and B. B. Chaudhuri, "Automatic extraction of correlation-entropy features for text document analysis directly in run-length compressed domain", In the IEEE Proceedings of ICDAR 2015.
6. M. Javed, P. Nagabhushan, and B. B. Chaudhuri, "Automatic page segmentation without decompressing the run-length compressed printed text documents," International Journal of Information Processing Systems (JIPS) (Accepted for Publication), 2015.
7. M. Javed, P. Nagabhushan, and B. B. Chaudhuri, "A direct approach for word and character segmentation in run-length compressed documents and its application to word spotting.", In the IEEE Proceedings of ICDAR 2015.
8. Y. Lu and C. L. Tan, "Document retrieval from compressed images," Pattern Recognition, vol. 36, pp. 987–996, 2003.
9. U. V. Marti, D. Wymann, and H. Bunke, "Ocr on compressed images using pass modes and hidden markov models," Proceedings of IAPR Workshop on Document Analysis Systems, pp. 77–86, 2000.
10. E. Regentova, S. Latifi, D. Chen, K. Taghva, and D. Yao, "Document analysis by processing jbig-encoded images," IJDAR, vol. 7, pp. 260–272, 2005.
11. M. Javed, P. Nagabhushan, and B. B. Chaudhuri, "Automatic detection of font size straight from run length compressed text documents," IJCSIT, vol. 5, pp. 818–825, February 2014.
12. M. Javed, P. Nagabhushan, and B. B. Chaudhuri, "Direct processing of run-length compressed document image for segmentation and characterization of a specified block," IJCA, vol. 83(15), pp. 1–6, December 2013.
13. CCITT-Recommendation(T.4), "Standardization of group 3 facsimile apparatus for document transmission, terminal equipments and protocols for telematic services, vol. vii, fascicle, vii.3, geneva," tech. rep., 1985.
14. CCITT-Recommendation(T.6), "Standardization of group 4 facsimile apparatus for document transmission, terminal equipments and protocols for telematic services, vol. vii, fascicle, vii.3, geneva," tech. rep., 1985.
15. TIFF, "(tagged image file format) revision 6.0 specification," tech. rep., 1992.
16. "Data visualization (www.sas.com/en-us/insights/big-data/data-visualization.html)."
17. C. Maa, "Identifying the existence of bar codes in compressed images," CVGIP: Graphical Models and Image Processing, vol. 56, pp. 352–356, July 1994.
18. A. L. Spitz, "Analysis of compressed document images for dominant skew, multiple skew, and logotype detection," Computer vision and Image Understanding, vol. 70, pp. 321–334, June 1998.
19. J. J. Hull and J. Cullen, "Document image similarity and equivalence detection," Proceedings of the Fourth International Conference on Document Analysis and Recognition (ICDAR'97), vol. 1, pp. 308 – 312, 1997.
20. J. J. Hull, "Document image similarity and equivalence detection," International Journal on Document Analysis and Recognition (IJDA'98), vol. 1, pp. 37–42, 1998.
21. "Binary viewer software (<http://www.proxoft.com/binaryviewer.aspx>)."