International Journal of Computer Science, Communication & Information Technology (CSCIT)
Vol.3 – issue2 - 2016
pp.33-38

Arabic Textual Images Compression

Jassem MTIMET
Signal, Image and Technology of Information Laboratory
National Engineering School of Tunis
Tunis el Manar University,
BP 37 Belvedere, 1002, Tunis, Tunisia
mtimat.jassem@yahoo.fr

Hamid AMIRI
Signal, Image and Technology of Information Laboratory
National Engineering School of Tunis
Tunis el Manar University,
BP 37 Belvedere, 1002, Tunis, Tunisia
hamidlamiri@yahoo.com

***Abstract*— In this paper, a novel compression approach for large Arabic textual images is proposed. Initially, input texts are segmented into patterns which represent sub-words. Then, patterns-matching procedure is used in order to find similar patterns within the image. Finally, to optimize the performances of this approach, an adaptive arithmetic coding is used to encode the resulting data streams. Experimental results show that the average compression ratio of our approach is better than other existing algorithms such as LZW and CCITT Group 4.**

Keywords*— *Arabic textual image; compression; pattern matching; neighborhood coding; code reduction.

I. INTRODUCTION (*Heading 1*)

Nowadays, document image compression is an important step in digital archiv-ing. The document images can be regarded as manuscript that was handwritten or printed and consist of semantically meaningful characters. The compression must be efficient to reduce the amount of data required to present a document image. Moreover the compressed file should be including a content-based searching oper-ation. Textual images contain two types of redundancies: bit-level and symbol-level. In order to retrieve information in documents image, these documents should be compressed by symbols. Then, textual images can be considered as bi-nary images that mainly compose of symbols. Existing document image compression approaches include several standards such as G3, G4, JBIG and JBIG2 [1] which achieve various compression ratios. The JBIG1 uses an adaptive context-based arithmetic coding known as Q-coder and has results better than those of CCITT Group4. Moreover, this standard supports multiple bit-plane encoding and progressive mode of transmission.

In the last decade, JBIG2 [2], [1] and JB2 [3] (used in DjVu file format) was introduced as binary image encoding methods which incorporate a compression approach based on pattern matching techniques. The aim of these techniques is to reduce the redundancy at the symbol level among the text region of an image. First all patterns in the document image are extracted. Each pattern corresponds to a symbol, a word or a punctuation mark. Then, all similar patterns are assigned to the corresponding prototype. Since the document image contain similar patterns can occur many times, a high compression ratios can be achieved by reducing these redundancies. In the encoding step, prototype symbols will be encoded by an efficient coding method then their corresponding index and residual pattern are used to encode the repeated symbols. The residual pattern, which represents the difference image between a symbol and its corresponding prototype, can influence on the quality of a decompressed image. In the lossless mode, the error maps are encoded into the compressed file. Otherwise, the compression will be lossy. In our experiments, the compression will be visually lossless because there is no visible error in the reconstructed image.

II. SYSTEM OVERVIEW

In this paper, we propose a compression method for large Arabic text in document images. One of the test images is shown as Fig. 1. We used three steps to accomplish the desired purpose. In the first step, which involves segmentation module, the input image is partitioned into individual symbols. In the second step, a pattern matching technique used to determine similar patterns. Finally, a neighborhood coding is performed to encode each symbol. The three steps in which the algorithm compress the large text are described below.

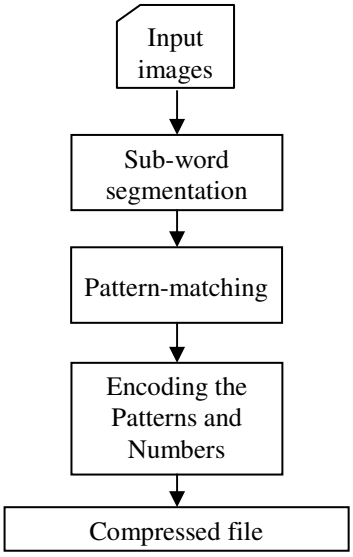
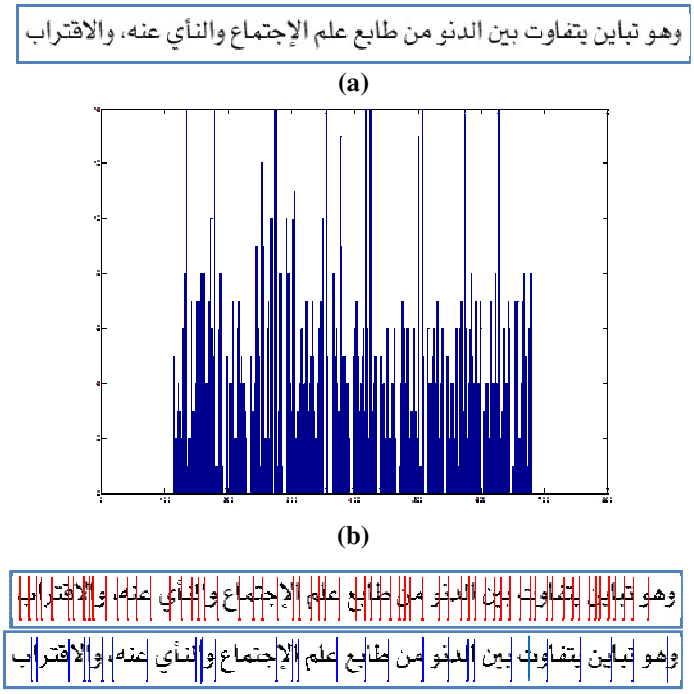


Fig.1. The block diagram of the proposed compression approach

A. *Segmentation module*

Word segmentation is a critical stage towards our method. Arabic word segmentation algorithms can be categorized into three types. The first type segment a whole Arabic word into characters [4], [5], [6]. Otherwise, the second category uses the whole words without segmentation [7], [8]. Recently, Mona et al. [9] proposed sub-words segmentation algorithm based on conditional labeling using horizontal and vertical projection. Moreover, Jawad et al [10] use both vertical histogram and connected component analysis to segment the words into sub-words. In this work we use the segmentation between connected characters of each word to obtain naturally segmented sub-words. In order to obtain sub-words that corresponding to one or multiple characters we apply a vertical projection on the input image as shown in Fig. 2. The bounding box of each sub-word is then determined (Fig. 3).



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

Fig. 2 (a) example of Arabic text (b) Vertical projection of a sub-word (c) segmented sub-words

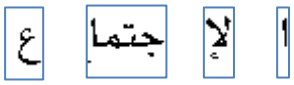


Fig. 3. Examples of segmented sub-words.

B. *Pattern matching module*

The pattern matching module uses the corresponding bounding box as binary features for the sub-word images. We compared these features using the correlation measure to obtain a similarity score between 0 and 1, which represents the extent of match between two sub-words. We used a similarity measure which is defined for two sub-words X and Y, as in equation (1).

$$d(X,Y)=\frac{1}{2}\left(1-\frac{s_{11}s_{00}-s_{10}s_{01}}{\sqrt{(s_{10}+s_{11})(s_{01}+s_{00})(s_{11}+s_{01})(s_{00}+s_{10})}}\right) \quad (1)$$

Where sij represent the number of corresponding bits of X and Y that have values i and j.

The smaller the similarity measure, the better is the match. Then, every detected sub-word in the input image is compared with every prototype and a distribution of similarity measure is obtained. Finally, sub-word is affected in rank in accordance with this score. Fig.4 shows an example of this operation.

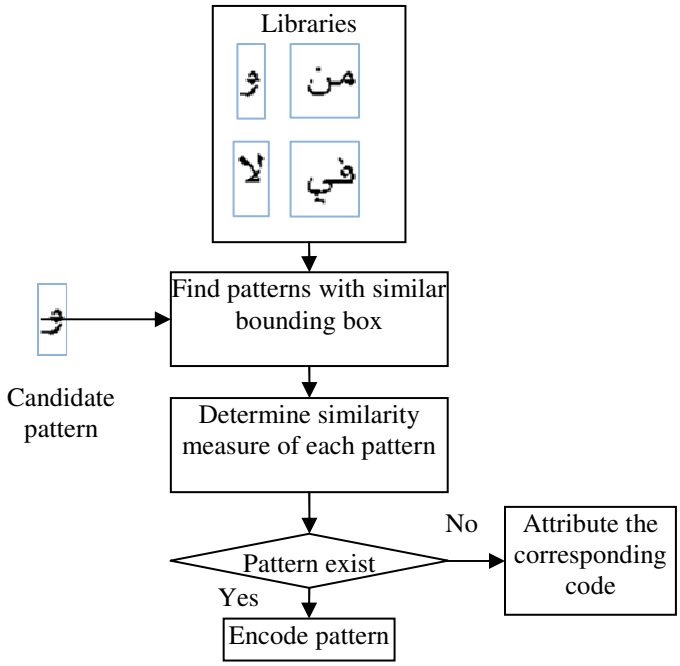


Fig.4. Example of pattern matching procedure

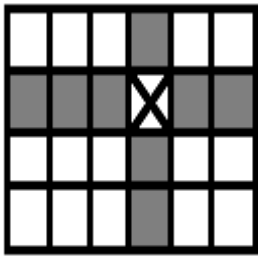
C. *Patterns coding*

- The neighbourhood coding

In this step, we transform the sub-word into a set of neighbourhood codes that represent each pixel contained within that pattern. Based on Tsang et al [11] algorithm, we characterize the sub-word by converting every foreground pixel into a vector V= (north, east, south, west). Each component presents the number of neighbouring pixels in the associated direction (see Fig. 5). Whereas this representation presents several drawbacks such as the large number of possible codes and redundancies since each code contains information that can be coded in another vector. In order to overcome this problem, it's necessary to reduce these redundancies either by re-scaling the vector's components or by giving another representation.

We scan the sub-word image sequentially from top-left to down-right and each time we locate a foreground's pixel I (i.e. pixels that have values equal to 1), we calculate their neighbourhood vector Vi. The set of all vectors produce a Φ matrix which represents the whole sub-word, where Φi= (xi, yi, ni, ei, si, wi) (see Fig.6). As we see this method transforms an n × m pixel image into p × 6 elements.

In order to reduce the redundancies of this representation we applied a special code reduction process.



Vi= (1, 2, 2, 3)

Fig. 5. Neighbourhood vector.

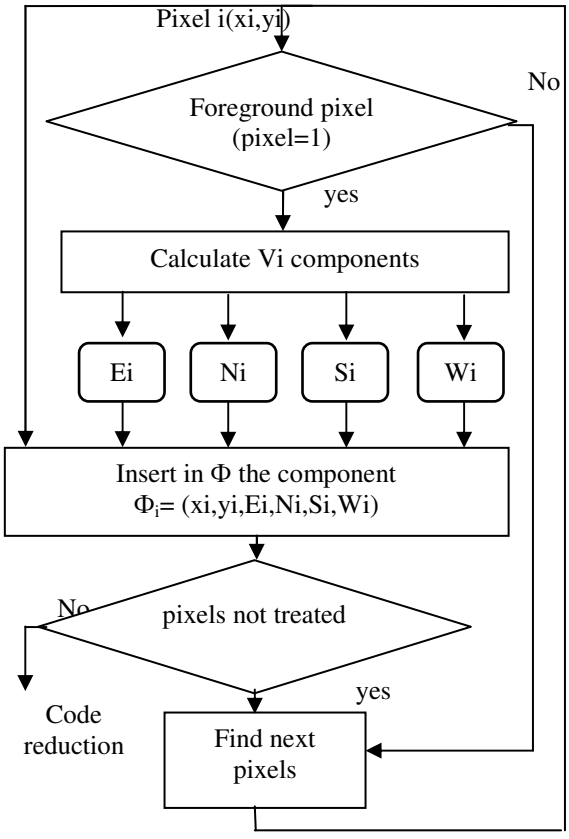


Fig.6. Neighbourhood coding principle.

- Code reduction

In this step we aim to obtain a compact representation that contain least number of neighborhood vectors to represent the sub-word without loss of information. A vector defined by equation 2, is used to identify the coding capacity CC of each neighborhood vector. Then the code with maximum capacity and their corresponding pixels is identified. Finally, this capacity is updated by subtracting the number of pixels will not be coded anymore (Fig.7).

$$CC_i = \sum Vi \text{ components} \quad (2)$$

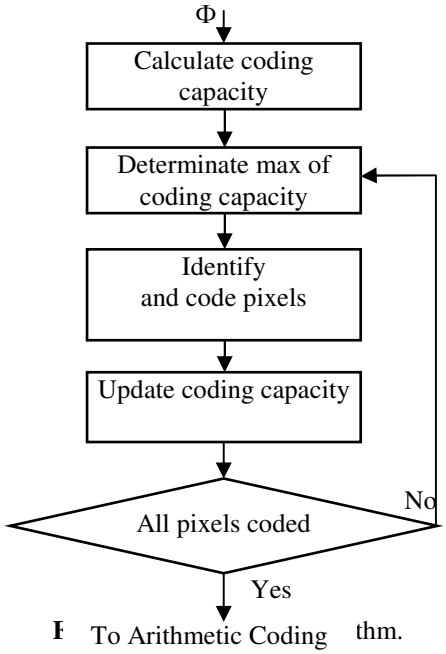
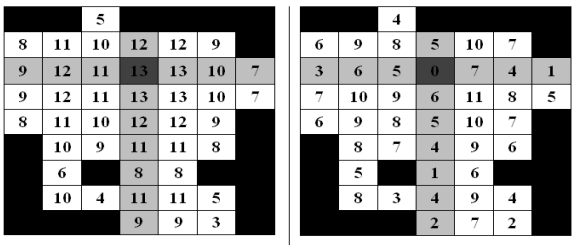


Fig.8 illustrates an example of updating coding capacity procedure. Each number represents the coding capacity of the corresponding pixel. As can be seen, the maximum capacity code is located on pixel (xi=4, yi=3) (Fig.8 (a)), so in the next step it will be marked as coded (capacity code=0). Furthermore, the pixel (xi=5, yi=2) has two neighbourhood pixels (xi=4, yi=2) and (xi=5, yi=3) that should be coded by pixel (xi=4, yi=3). So, in order to update their capacity code we will subtract 2 from this value in the next step (Fig.8 (b)).



(a) First step (b)Second step

Fig.8. Update coding capacity example.

- The arithmetic coding

Arithmetic coding is the stepgenerating the binary bitstream. This algorithm is a form of entropy coding which overcomes the drawback of the Huffman coding [12].

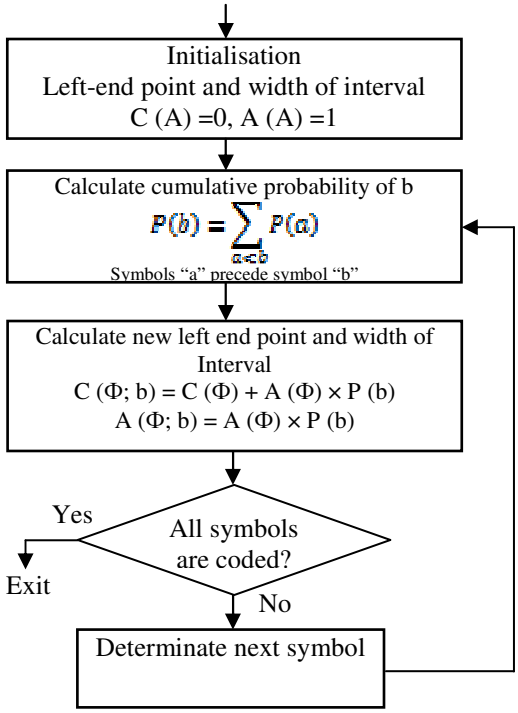


Fig.9. Arithmetic Coding algorithm.

The basis of this algorithm is to represent a sequence of n symbols as a number between 0 and 1. It is based on iterative division of an interval to the sub-intervals proportional to the symbol's probability distribution (Fig.9**Erreur ! Source du renvoi introuvable.**).

III. EXPERIMENTAL RESULTS

In this section, we will present compression results of our approach on specific data sets. Comparison with the existing compression algorithms is given.

A. Data set for experiment

In the literature, few data sets are built for Arabic textual image compression. To validate the effectiveness of our algorithm, we collect a variety of large textual images. It consist 40 heading and titles extracted from various Arabic newspapers and books (see **Erreur ! Source du renvoi introuvable.**). The image resolutions of this data set are all 300 dpi.

B. Results and discussions

First, let us show the results of code reduction step. As shown in Fig.11, the number of representative pixels was reduced approximately to 23%.

Then, after getting the reduction rate, let us look into the results of compression file sizes. We compress the data set with LZW used in the GIF file format, CCITT group 4 used in the TIFF file format and our approach. Compression results are shown in **Erreur ! Source du renvoi introuvable.** and Fig. 12. We can see, compared with LZW and CCITT group 4, compression file sizes of our approach on the data set are all the smallest.

Fig.10.Textual image portions

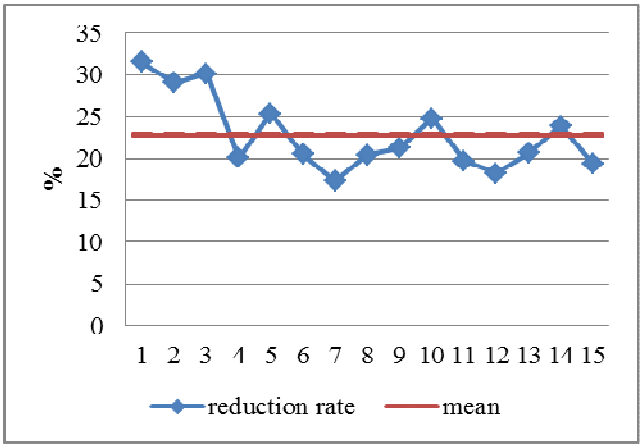


Fig.11. Reduction rate

TABLE I. COMPRESSION FILE SIZES

Data set	Original size (MB)	LZW	CCITT group 4	Our
1	8,060	1,360	0,750	0,298
2	8,060	1,310	0,478	0,057
3	8,060	1,340	0,540	0,151
4	8,060	1,390	0,572	0,186
5	8,060	1,310	0,508	0,158
6	8,060	1,430	0,612	0,196
7	8,060	1,450	0,616	0,273
8	8,060	1,370	0,552	0,207
9	8,060	1,290	0,456	0,109

10	8,060	1,340	0,514	0,238
11	2,060	1,080	0,436	0,189
12	2,060	1,060	0,402	0,113
13	8,060	1,370	0,568	0,270
14	8,060	1,310	0,486	0,146
15	8,060	1,440	0,612	0,257

IV. CONCLUSION

In this work, we developed an algorithm for large Arabic textual image compression. The main idea of this approach is Neighbourhood coding with reduction. Based on the extracted patterns, we can improve the pattern-matching procedure, and this lead to further compression by Neighbourhood coding.

Experimental results show our approach can improve compression ratio on large Arabic textual images. The compression results satisfy the lossless context, namely no error happens during compression. Our future work will concentrate on integrating on our approach in compound image compression framework.

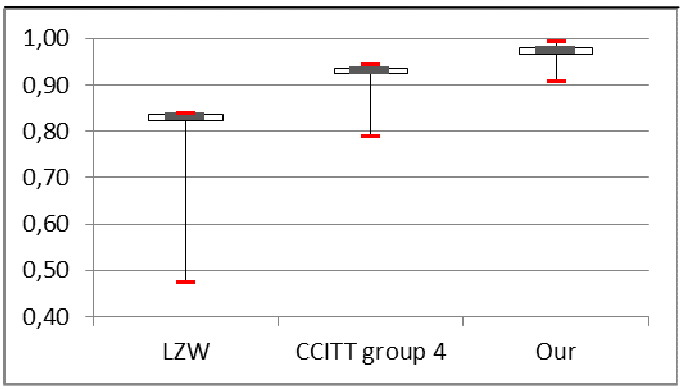


Fig.12. The compression Gain

REFERENCES

[1] P. G. Howard, "Text Image Compression using Soft Pattern .Matching," Computer Journal, 40:2-3, pp. 146-156, 1997.

[2] ISO/IEC 14492 and ITU-T recommendation T.88, "Information Technology—Lossy/Lossless Coding of Bi-level Images,"2000.

[3] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio and Y. LeCun, "High Quality Document Image Compression with DjVu," Journal of Electronic Imaging, 7(3):410-425, 1998.

[4] Abdulaziz, E.Alsaif, K.I., "Radon Transformation for Arabic Character Recognition", International conference on Computer and Communication Engineering, Kuala Lumpur, Malaysia, 13-15 May 2008, pages 433-438, 2008.

[5] Zidouri, "A PCA-based Arabic Character feature extraction", 9th International Symposium on Signal Processing and Its Applications, Sharjah, UAE, 12-15 Feb. 2007, pp. 1 – 4, 2007.

[6] Mahmoud, S.A. Mahmoud, A.S., "Arabic Character Recognition using Modified Fourier Spectrum (MFS)", Cybernetics and Systems, Vol. 40, No. 3, April 2009, pp. 189 – 210.

[7] Gregory R Ball, Sargur N Srihari, Harish Srinivasan, "Segmentation-Based And Segmentation-Free Methods for Spotting Handwritten Arabic Words", Tenth International Workshop on Frontiers in Handwriting Recognition, CEDAR 2006.

[8] Sargur N. Srihari, Gregory R. Ball and Harish Srinivasan. "Versatile Search of scanned Arabic Handwriting", Arabic and Chinese Handwriting Recognition, LNCS 4768, Springer 2008, pp. 57-69, 2008.

[9] Mona Omidyeganeh, Reza Azmi, Kambiz Nayebi and Abbas Javadtalab, "A New Method to improve Multi Font Farsi/Arabic Character Segmentation Results: Using Extra Classes of Some Character Combinations", Multimedia Modeling Conference, Nanyang Technological University, Singapore MMM2007, pages 670-679, 2007.

[10] Jawad H AlKhateeb, Jianmin Jiang, Jinchang Ren, and Stan S Ipson, "Component-based Segmentation of Words from Handwritten Arabic Text", World Academy of Science, Engineering and Technology, vol. 41, pp.344-348, 2008.

[11] I.J. Tsang, I.R. Tsang, and D. Van Dyck, "Image coding using neighbourhood relations," Pattern Recognition Letters, vol. 20, no. 3, pp. 1279--1286, 1999.

[12] Witten, Ian H., Radford M. Neal, and John G. Cleary "arithmetic coding for Data Compression," in Communications of the ACM, 1987, pp. 30(6):520–540.