Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

1-D chaincode pattern matching for compression of Bi-level printed farsi and arabic textual images ☆

Hadi Grailu [a], Mojtaba Lotfizad [a,*], Hadi Sadoghi-Yazdi [b]

[a] Department of Electrical Engineering, Tarbiat Modares University, Tehran, Iran
[b] Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

ARTICLE INFO

Article history:
Received 26 October 2008
Received in revised form 19 February 2009
Accepted 13 April 2009

Keywords:
Pattern matching
Bi-level text image
Image compression
Chain code
Farsi and Arabic text images

ABSTRACT

In some scripts, especially the Farsi/Arabic script, letters normally attach together and produce many different patterns, some of which are fully or partially similar. Detecting such patterns and exploiting them to reduce the library size, has a rather great effect on the compression ratio.

In this paper, a lossy/lossless compression method is proposed for bi-level printed text images in archiving applications. For this, we propose a new 1-D pattern matching technique in the chain coding domain that uses the proposed technique of detecting the repetitive sub-signals in order to detect the fully or partially similar patterns.

Experimental results show that the compression performance of the proposed method is considerably better than those of the existing bi-level printed text image compression methods as high as 1.8–4.2 times in the lossy case and 1.6–3.8 times in the lossless case at 300 dpi.

1. Introduction

There are many methods for compression of natural images [21–24], the most important of which include transform-based methods [23], vector quantization [7] and fractals [5]. Such methods reduce the redundancy at the pixel level; however, the text images have most of their redundancy at the symbol level. Hence, such methods have a moderate or poor efficiency on text images.

Most existing text image compression methods work on the basis of pattern matching (PM), because it can reduce the redundancy at the symbol level. In this method, first all patterns (marks) in the input image are extracted. Each pattern is a set of connected (neighboring) black pixels. Then, all similar patterns are grouped together and assigned a prototype which is usually the most similar pattern to each one of them. Of course, each pattern is not necessarily fully similar to its corresponding prototype, and is usually somewhat different. The difference image is called the “residual pattern” or the “error map”, and its pixels are called the “residual pixels” or the “error pixels” [17]. The set of all prototypes is called the library. Each prototype in the library has an index number. The image patterns are extracted successively and compared with the prototypes to find a match. If a match occurs, the corresponding index of the matched prototype, the relative position of the current pattern with respect to the previously processed pattern, and the corresponding residual pattern are computed and saved; otherwise the current pattern is added to the library as a new prototype and a new index number is assigned to it. Finally, all prototypes, number sequences, and probably the whole or parts of the residual patterns should be encoded. In the lossy or limited bit rate compression, all or some of the residual patterns may be omitted and hence need not be encoded. Since the residual patterns (and even the residual pixels) have not equal importance [17,15], it is reasonable to omit the less important ones and encode the rest. This technique preserves the reconstructed image quality as much as possible.

In the Farsi/Arabic script, contrary to the printed Latin script, letters usually attach together and produce various patterns. Therefore, some patterns are fully or partially subsets of some others. Detecting such situations and exploiting them for reducing the number of library prototypes have a great effect on the compression efficiency. This is because such situations frequently occur in Farsi and Arabic text images. Fig. 1a shows an example of the full similarity for two sample Farsi sub-words (or prototypes), where the whole smaller prototype is repeated in the other. Fig. 1b shows two kinds of the partial similarity. In the first (left) one, some parts of a prototype are repeated in different parts of another prototype. In the second (right) one, some parts of a prototype are repeated in different parts of several prototypes.

In this paper, a PM-based method of bi-level printed text image lossy/lossless compression is proposed for archiving purposes

☆ This work was supported by the Iranian Telecommunication Research Center (ITRC).

* Corresponding author. Tel.: +98 9128024969; fax: +98 2182883381.
E-mail addresses: grailu@modares.ac.ir, grailuhadi@yahoo.com (H. Grailu), lotfizad@modares.ac.ir (M. Lotfizad), sadoghi@sttu.ac.ir (H. Sadoghi-Yazdi).

0262-8856/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2009.04.006

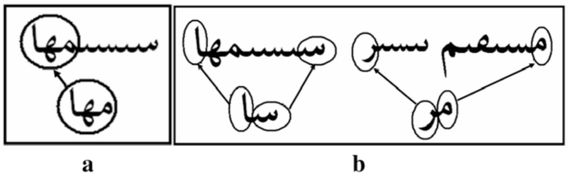


Fig. 1. (a) An example of a full similarity, (b) two examples of partial similarity.

which demand for high compression ratios, a relatively low reconstruction time and a reasonable image quality. The proposed PM technique operates in the 1-D domain of chain code sequences (or signals) and can detect almost any kind of repetitive patterns. Consequently, it can considerably reduce the library size and yield a much higher compression ratio than those of the existing text image compression methods for Farsi and Arabic text images as well as those types of printed Latin text images which contain touching characters or some fancy font faces. Nonetheless, it still has a higher compression ratio for other scripts as well. In this paper, we have modified and employed the multi-symbol QM coder [10] to encode the sequences of numbers. The proposed compression method uses a particular template matching technique which can compare patterns independently of their font sizes.

2. Review of the existing pm-based compression methods

Pattern Matching is the most widely used tool for compression of bi-level text images and almost all existing efficient text image compression methods or standards work on the basis of this idea. These methods are grouped into lossy and lossless ones. The PM idea, originally introduced in [1], did not encode the residual patterns, and thus is a lossy compression method. Also it did not introduce any compression method for prototypes. Pratt et al. [20] introduced a method named the "combined symbol matching" (CSM) for improving the performance of the mentioned method in which it left the infrequently used patterns and encoded the residual patterns using a two-dimensional run-length coding as in ITU Group3 (*G*3) and Group4 (*G*4) facsimile standards. The proposed method of [4] uses a preloaded library with alphabet characters with some different fonts. The pattern matching and substitution (PMS) method [14] can also handle graphics in the text images. In this method, large bodies of image graphics are divided into some smaller parts, each of which is considered as a pattern. The weighted AND–NOT (WAN) [8] was introduced as an improvement to the CSM method which had a better pattern matcher. One of the authors later stated elsewhere [9] that the performance of the CSM, PMS and WAN methods is decreased when the symbol size is decreased or the quantization noise is increased. Thus, he proposed the combined size independent strategy (CSIS) [9], which was similar to the PMS method but tried to operate independently of the symbol size by performing symbol size normalization. In [27], a multistage lossy method was proposed, where after some preprocessing based on a histogram analysis and skew correction, a PM with a rigorous comparison of patterns was applied. Some of similar patterns may be considered as dissimilar due to quantization noise. Thus, at the next step, such patterns were found and merged by a multi-stage structural clustering. This method has used the Q-coder for encoding the prototypes, and arithmetic coding for encoding the number sequences. The *mgtic* method ([26,25]) was based on pattern matching and used an enhanced encoder in which the reconstructed image, using the prototypes, was used for conditional context-based arithmetic coding of the original image. The proposed method in [10] had a similar encoder but had a difference where a combination of the library prototype and current pattern was used as the context for encoding the current pattern. This technique was named *soft pattern matching* or *SPM* [10]. The lossy *SPM* method performed some further processing for lossy compression. The most important procedure was the *selective pixel reversal*, in which the color of poorly predicted pixels was reversed if certain conditions were satisfied. The *SPM* method had a rather better performance than that of the *mgtic* method [10].

Almost all PM-based methods of bi-level text image compression use the weighted error map [25,26]. In the CSM method, first the error map is computed by XORing the two patterns. Then, a weighted sum of the map elements are computed and compared against a threshold. The larger the cluster size of the error map pixels, the larger would be the weight [20]. The *WAN* method distinguishes between the black-to-white errors and the white-to-black ones. It also uses the perimeter of patterns to determine if a match should be attempted. The PMS method rejects a match, if any position in the error map is found to have four or more neighbors set to 1. The *SPM* and *mgtic* methods use the Hamming distance which is the count of the number of mismatched pixels between the patterns when they are aligned according to the geometric centers of their bounding boxes [25,10]. In the lossy compression case, all or some residual pixels are omitted and need not be encoded. In [17,16] a method of prioritization of the residual pixels for encoding has been proposed based on the distance of these pixels from their nearest edges. This prioritization helps to have an effective progressive compression since the most important information has already been encoded.

The early standards such as ITU-T *G*3 and *G*4 standards have not so far used the PM technique and operate on the basis of various run length coding techniques and Huffman codes [26,25,10]. In these standards the run time is more important than the compression ratio. The ISO/IEC *JBIG1* standard [12] for lossless bi-level image compression uses the adaptive context-based arithmetic coding using a special 10-pixel template for defining the context [12] and has results better than those of the mentioned standards ([26,10] but still does not employ the PM technique [10]. The *JBIG2* standard is the best existing compression standard, introduced in the last decade and has better compression performance than that of all previously mentioned standards [11,29,28]. This standard has not specified any specific encoder but the best existing *JBIG2*-based methods, such as *jb2*, work on the basis of the soft pattern matching technique and therefore, are similar to the *SPM* method in this respect. The *jb2* method is a lossy text bi-level image compression which is used in the *DjVu* document image compression method [3].

Pattern encoding is an important part of any PM-based compression method. The *mgtic* method uses a specific two-level coding scheme [18] for encoding the prototypes. With this technique, a larger template without a high learning cost can be used for context-based arithmetic coding. For encoding the residual image, as mentioned before, the original image is conditionally encoded using the reconstructed image. The numbers are encoded using a specific technique [2] based on arithmetic coding. The *SPM* method uses a *JBIG1*-like method for encoding the prototypes and the soft pattern matching for encoding the residual patterns. It also uses the multi-symbol QM-Coder [10] for encoding the numbers. QM-Coder is a specific binary arithmetic coding which provides both an approximate arithmetic coder and a table-driven technique for updating the probability estimate for each event's context after the event is coded; probability estimation is linked to coding both to increase coding speed and to use the coder as a source of pseudo-random bits.

Boundary description methods such as the chain code (CC) description convert the 2-D text image signals into a 1-D one, and in fact, can be considered as a compression method in its own right. Chain code is a description method which has been mostly used for bi-level images. It was first introduced in [6]. Differential chain code is another variant of the chain code which was introduced in [19]. Some advantages of the (differential) chain code are as follows. First, the (differential) chain code is a compact representation of bi-level image [13]. Secondly, the chain code is a translation invariant representation of a bi-level image or object. This property makes it easier to compare objects. Thirdly, the chain code is a complete representation of an object or curve. Therefore, we can compute any shape feature from the chain code [13]. Fourthly, the computation of the (differential) chain code needs a relatively low time. However, the chain code is neither rotation nor scale invariant. This is a significant disadvantage for object recognition, although the chain code can still be used to extract rotation invariant parameters, such as the area of the object [13]. To our best knowledge [23,26], the chain code technique has not so far been used for bi-level text image compression.

In this paper, a lossy/lossless bi-level printed Farsi and Arabic text images compression method is introduced which first converts the 2-D text image signal to 1-D chain code and coordinates signals and then employs pattern matching in the new 1-D domain of chain code signal using the proposed technique of detecting the repetitive sub-signals, in order to find the best library prototypes which are a set of 1-D signals. The chain code signal is then described by these prototypes as well as the corresponding residual chain code signal, both of which are finally encoded using the proposed modified multi-symbol QM coder. The proposed compression method has both the lossy and lossless modes. In the lossy case, it uses some chain code-based procedures which increase the compression ratio, and at the same time, improve the image quality and/or readability as much as possible. These procedures include the spot noise removal, boundary smoothing, filling the inner holes of characters, and quantizing the coordinates signal.

3. Chain code, differential chain code and some applications

In this section, some techniques based on chain code are presented some of which including boundary smoothing and spot noise removal is used only in the lossy mode of the proposed compression method. The last technique for detection of repetitive sub-signals is always used (in both lossy and lossless modes). The proposed technique for template matching has not any application in the proposed bi-level image compression method of this paper but it can be used as a better substitute for template matching in existing 2-D PM-based image compression methods.

Chain code [13] is a boundary description method which converts the 2-D image signal into the corresponding 1-D one. We can perform this conversion by concatenating chain code sequences of image patterns (and probably the corresponding holes) and producing the final "chain code (CC) signal". Also, all the corresponding start pixel coordinates are put together to form the final "coordinates signal". The coordinates of the start pixels are computed relative to the previous processed ones, except the first start pixel whose coordinates are computed relative to the original image dimensions.

In a bi-level text image, all the information exists at edges; thus, the boundary description methods, such as the chain code are, in fact, compression methods too, because they do not assign any extra code number for the inner regions of pattern body. Also as they are suitable for shape description, they can describe both the text and graphical parts of a text image. Thus, contrary to some existing compression methods, there is no need to process the text and graphical parts separately.

3.1. Boundary smoothing

The differential chain code (DCC), contrary to the chain code, is almost rotation-invariant. It can be used to design nonlinear smoothing filters that can remove specific outgrowths and indentations, and thus smooth out the edges of patterns.

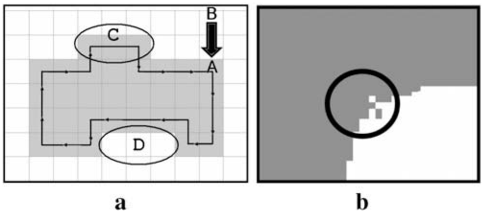


Fig. 2. (a) A sample bi-level pattern, (b) a sample of temporary deviation from a point of main route and returning again to the same point.

The bi-level pattern of Fig. 2a has an outgrowth and an indentation which are marked by C and D, respectively. By inspection of DCC, we can detect such specific shapes and remove or fill them. Each outgrowth or indentation is, in fact, a temporary deviation from the main route. Thus by inspection of DCC signal in a sliding window of length *W*, we can detect any desired shape and remove or change it; this operation is in fact a nonlinear filtering. The general prototype of some outgrowths, such as C, is $\{-1, Z(N_1), 1, Z(N_2), 1, Z(N_1)\}$ and for indentations, such as D, is $\{1, Z(N_1), -1, Z(N_2), -1, Z(N_1)\}$ in which $Z(N)$ means a sequence of zeros with the length of *N*. Total length of any shape prototype should be less than the filter length, i.e., *W*. In order to smooth out such shapes, we replace their corresponding DCC with a sequence of zeros with the length of $Z(N_2)$+2. The larger the value of *W*, the longer is the length of prototypes which we can detect and smooth out. Another type of smoothing is to detect the temporary deviation from a point of the main route and returning to the same point, a sample which is shown in Fig. 2b. By designing or determining prototypes of various desired shapes, we can employ a variety of filters on text images.

3.2. Spot noise removal

In noisy bi-level images which contain spot noise, like the noisy bi-level pattern of Fig. 3a, there exist either small holes in body of patterns or undesired black dots in the background of the text image, both of which lower the text image quality. The chain code description can help remove or reduce this type of noise using a proper nonlinear filter similar to the one mentioned in the previous section. For this we analyze the chain code signal to detect all such temporary deviations from a point of the main route length of which is smaller than a predetermined threshold value T_{ih}. Thus, we can remove the undesired holes or dot patterns except some of the ones which are attached to the boundaries of patterns.

Fig. 3b shows the result of employing such a nonlinear filter to the noisy pattern of Fig. 3a. As can be seen the CC-based nonlinear filter have removed most of the noisy dot patterns or holes, and

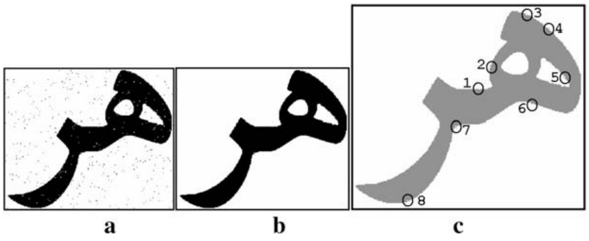


Fig. 3. (a) A sample noisy bi-level image, (b) the result of noise removal with $T_{\text{ih}} = 10$, (c) the same result with which is shown in larger scale.

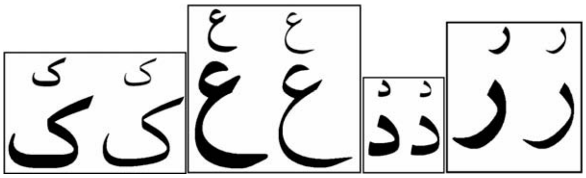


Fig. 4. Some Farsi alphabet letters with the font face *B-Lotus* and different font sizes and manners.

thus the quality of the resulting image is much better than that of the original. Fig. 3c shows this result in a larger scale. As mentioned before, the nonlinear filter may not remove some of noisy dot patterns or holes which are attached to the boundaries of the desired pattern. Some of these situations are shown and marked by some numbers in Fig. 3c. As can be seen, these situations sharpen or unsmooth the pattern boundaries. Most of these situations can be removed or smoothed out by employing the CC-based nonlinear smoothing filter of the previous section. For example, the situation marked by number 1 is in fact an indentation, the situation marked by number 3 is an outgrowth and the situation marked by number 7 is a temporary deviation from the main route.

3.3. Template matching

CC or DCC signals can be used to compare two patterns and determine the degree of their similarity. To this end, we can compute their corresponding CC or DCC signals and stretch them such that both of them have the same length. Then we use a similarity measure like the cross correlation, to determine the degree of their similarity. The weighted cross correlation of two discrete signals $x(n)$ and $y(n)$ of the length, N is always between -1 and +1, and can be defined as

$$CC_{xy} = \frac{\sum_{n=1}^{N} x(n)y(n)}{\sqrt{(\sum_{n=1}^{N} x(n)^2)(\sum_{n=1}^{N} y(n)^2)}}. \tag{1}$$

In this paper, we introduce another similarity measure which has a lower computational complexity and more efficiency. This measure whose value is always between 0 and 1, is based on the difference of the two signals and is defined as

$$DC_{xy} = 1 - \frac{\sum_{n=1}^{N} |x(n) - y(n)|^m}{N \times MAD^m}, \tag{2}$$

where MAD is the maximum absolute difference of signals $x(n)$ and $y(n)$ and m is a predetermined constant value.

In each of Figs. 4–6, the images of four Farsi alphabet letters are shown. In each figure a specific font face has been used. In Fig. 4 the font *B-Lotus*, in Fig. 5 the font *Mitra*, and in Fig. 6 the font *2-Elham* have been used. For each letter in each figure, four cases are shown. The two letters in the upper row have the same font size, but are smaller than the two letters in the bottom row. In each row, the right letter is normal and the left one is bold.

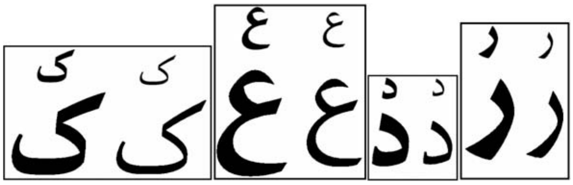


Fig. 5. Some Farsi alphabet letters with the font face *Mitra* and different font sizes and manners.

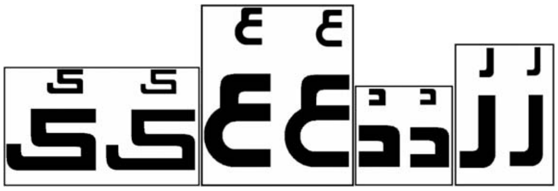


Fig. 6. Some Farsi alphabet letters with the font face *2-Elham* and different font sizes and manners.

For inspection of the CC-based comparison performance on comparing the patterns of Figs. 4 and 5, we can compare the CC signal of an arbitrary letter of them with all the 32 patterns of these two figures. For this, we first compute and normalize the corresponding CC signals, and then, compare them using one of the two similarity measures of the relations (1) or (2). To compare the letters of Figs. 4 and 5, for ease of notation, we have assigned a name and an index to each of the patterns, as shown in Fig. 7a. Now, for example, if we compare the pattern A1 with all the patterns shown in Fig. 7a using one of the two similarity measures of relations (1) or (2), we obtain the graph of the degree of similarity shown in Fig. 7b. In this figure, the solid and dashed curves correspond to the relations (1) and (2), respectively; and m is set to 1.3 in the latter. The vertical axis represents the degree of similarity and the horizontal axis represents the index of patterns of Fig. 7a to which the pattern A1 has been compared.

The pattern A1 is similar to the patterns A1–A8, although it may have a different font size, face or manner. As can be seen from the

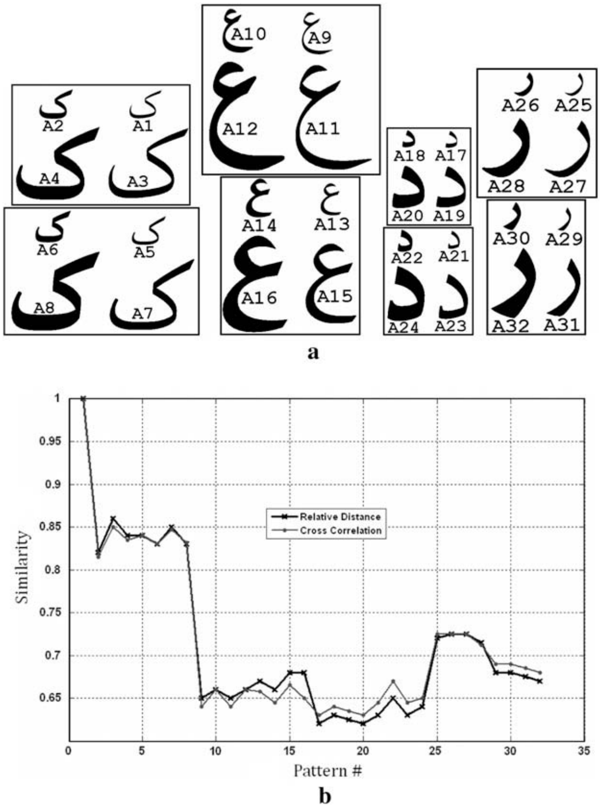


Fig. 7. (a) The letters of Figs. 4 and 5 with the font faces *B-Lotus* and *Mitra* which are named and indexed. (b) The similarity graph resulting from comparison of pattern A1 to all letters of part (a).

graph of Fig. 7b, the similarity measure for indices 1–8 is higher than that of the other indices, which shows that the pattern A1 is similar to patterns A1–A8 in spite of the differences between their font sizes or faces or manners. We can see that the CC signal comparison of patterns using one of the similarity measures of relations (1) and (2) can discriminate between different letters and can group similar patterns with different font size, face or manner. Thus, we can use the CC-based template matching in the pattern matching based text image compression in order to further reduction in the library size and consequently, increase the compression ratio.

If a character shape is considerably changed from one font face to another, such as the font *2-Elham* relative to the font *B-Lotus*, then the degree of similarity and consequently, the discrimination property is decreased. This can be understood from the curve shown in Fig. 8 which is the similarity graph of comparison of the pattern A1 with all the patterns of Fig. 7a but for the two font faces of *B-Lotus* and *2-Elham*.

For comparison of patterns, we could also use DCC instead of CC, but we should take two points into consideration. First, DCC signal is sensitive to the relative changes of directions not the directions themselves. On the other hand, usually Farsi/Arabic patterns have straight or smooth boundaries; thus, DCC signal contains a relatively high number of zeros. The cross correlation of the relation (1), contrary to the similarity measure of (2), is sensitive to the value of two signals and not to their differences; thus, if we desire the cross correlation to work well, we should bias the DCC signals by a proper value. This bias value should be large enough such that number of zeros in the biased DCC signal decreases considerably and on the other hand, the bias value should be small enough, such that the difference of two signals is comparable to the average absolute values of each of them. Since the DCC signal mostly contains zeros, −1s, and +1s, the best value of the bias seems to be a value of 2 which has been approved experimentally.

The second point regarding to the use of DCC signals for comparison of patterns is that the degree of similarity of two patterns is generally higher than that of the case of CC signals. The reason can be understood by considering that the DCC signals are sensitive only to the "changes" of directions and not to the directions themselves. For example all straight directions, including up, down, left, and right, have the same DCC value of zero, although their CC codes differ from each other; thus, more similarity will occur in the case of comparison of the DCC signals. This property makes the DCC signal rather undesirable for the template matching but on the other hand, it will be suitable in compression applications because as stated above, the DCC signal has more uniformity and lower variance in comparison to the corresponding CC signal and thus, is more suitable for compression by techniques such as run-length coding and multi-symbol QM coder [10]. Because of this suitable property, in this paper, we will convert the CC residual and prototype signals to their corresponding DCC ones and compress them using the run-length coding and a modified multi-symbol QM coder.

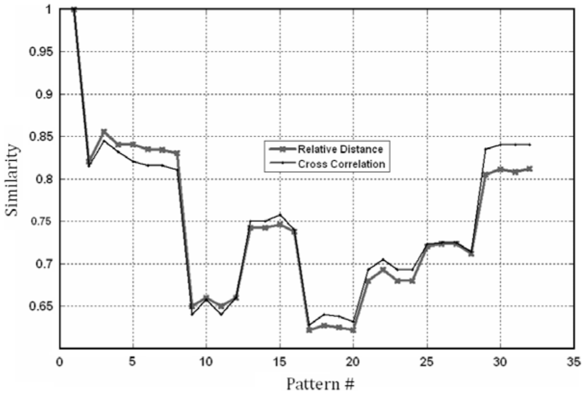
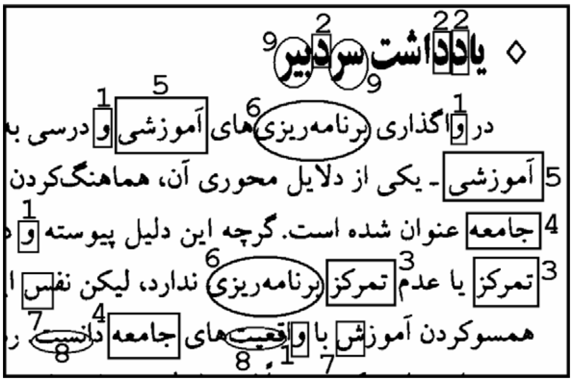


Fig. 8. The similarity graph of comparison of pattern A1 with the patterns of Fig. 9a but for the two fonts *B-Lotus* and *2-Elham*.



یادداشت سردبیر

در واگذاری برنامه‌ریزی‌های آموزشی و درسی به

آموزشی ـ یکی از دلایل محوری آن، هماهنگ‌کردن

جامعه عنوان شده است. گرچه این دلیل پیوسته و

تمرکز یا عدم تمرکز برنامه‌ریزی ندارد، لیکن نفس ا

همسوکردن آموزش با واقعیت‌های جامعه دانسته ر

9
2
22
9
1
5
6
1
5
1
4
6
3
3
7
4
8
8
1
7

Fig. 9. Types of pattern repetition in a text image.

3.4. Detection of repetitive sub-signals

The technique presented in this sub-section contributes substantially to the main theme of the paper. It helps to find 1-D library prototypes in the proposed 1-D PM-based image compression method.

There are some types of repetitive patterns in typical Farsi/Arabic text images. Fig. 9 shows a part of a sample Farsi text image in which some samples of these repetitive patterns are marked by numerical marks. All similar repetitive patterns are marked by a same and distinct number. The first type of repetitive patterns is full repetition. In this type, a pattern or a group of patterns which may corresponds to a word, phrase and even a sentence, repeats in the text image. In Fig. 9, the cases marked by numbers 1 and 2 are samples of repetition of a pattern, and the samples marked by numbers 3–6 are samples of repetition of a group of patterns.

The second type of the repetitive patterns is partial repetition in which some part of a pattern is repeated in the text image. A sample of this type is marked in Fig. 9 by number 7. The third type of pattern repetition is boundary repetition in which some part of CC or DCC description of boundary of a pattern repeats inside the CC or DCC signal of the corresponding text image. Two samples of this type of repetition are marked by numbers 8 and 9 in Fig. 9 and their more clear illustrations are shown in Fig. 10.

It is obvious that the third type of repetition patterns always includes the first type, but not necessarily the second type, because if the partial repetitive pattern lies in the middle of its corresponding pattern, the corresponding CC (or DCC) sub-signals of upper and lower boundaries lie apart in the CC (or DCC signal) of the text image. But in this case, we can consider the situation as two separate partially repetitions; consequently, if we find all repetitive sub-sig-

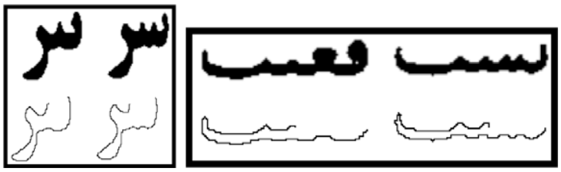


Fig. 10. Some samples of the third type of pattern repetition corresponding to the samples marked by numbers 8 and 9 in Fig. 9.

nals of CC (or DCC) signal of a text image, we would detect all existing repetitive patterns. Indeed, if we find larger repetitive sub-signals, we would detect larger repetitive patterns and the library size would be decreased more.

It should be noted that in library size reduction, we aim to compromise between three factors including the length of a group of repetitive sub-signals, the corresponding number of repetitions, and the corresponding average degree of similarity between repeated sub-signals. Usually, by increasing the length of a repetitive sub-signal, the corresponding number of repetitions and the average degree of similarity are decreased; the first factor causes a further reduction in the library size and consequently increases the overall compression ratio, but, the last two factors cause an increment in the variance of the corresponding residual CC signal, and thus, decrease the overall compression ratio. Thus, we should compromise between these factors in order to increase the overall compression ratio. To this end, we can define a proper objective function of these factors and optimize it.

In the proposed repetitive sub-signals detection, the aim is to find all groups of repeated sub-signals which simultaneously maximize the mentioned three factors as much as possible. This technique converts the input text image to the corresponding CC signal and performs all subsequent procedures in the new 1-D domain. We define $W_{opt}(X = X_0)$ as the optimal length of a group of repetitive sub-signals, one of which starts at the point (index) $X = X_0$ of the CC signal. The proposed algorithm, for each point X of the CC signal, considers a "candidate sub-signal" which is a sub-signal that starts from this point and has the length W as a parameter. Then, it compares the candidate sub-signal with all possible co-length sub-signals of the CC signal, using one of the similarity measure (2) with $m = 1.3$. Finally, by varying W and optimizing an objective function, it computes a score function and compares it with a threshold in order to determine whether the current sub-signal can be a library prototype or not; if so, then its proper length, $W_{opt}(X = X_0)$, is computed as well.

Corresponding to each specific value of W, we obtain a similarity curve, whose value at a point n represents the degree of similarity between current candidate sub-signal (starting at point $X = X_0$) and that sub-signal which starts at the point n of the CC signal. The algorithm considers W as a parameter which varies in a predetermined interval by a predetermined step. By computing all similarity curves corresponding to all possible values of W, we obtain a 3-D similarity surface versus the axes W and n. This similarity surface corresponds to the point $X = X_0$.

As W is increased, the number of repetitive sub-signals and the average degree of their similarity generally is decreased. Thus, we should define and optimize a proper objective function in order to compromise this trade-off and determine the proper length of the current candidate sub-signal, $W_{opt}(X = X_0)$. We define this objective function as follows to find the length of the current candidate sub-signal

$$W_{opt}(X_0) = \arg\max_{SSL(i)} \left\{ \frac{K_1.NP(i).SSL(i)}{L} + K_2.SV(i) \right\} \text{ for all } i, \tag{3}$$

where L is the length of the CC signal, and K_1 and K_2 are weights of the mentioned factors, i.e., *NP*, *SV*, and *SSL* which correspond to the number of peaks, similarity value, and sub-signal length, respectively. These weights are experimentally set to 1. The index i is related to the i'th allowable value of W. After finding the best value of W at the point $X = X_0$, we compute the following score function

$$i_{opt}(X = X_0) = \arg\max_{i} \left\{ \frac{K_1.NP(i).SSL(i)}{L} + K_2.SV(i) \right\} \text{ for all } i, \tag{4}$$

$$Score(X = X_0) = \frac{K_1.NP(i_{opt}).SSL(i_{opt})}{L} + K_2.SV(i_{opt}). \tag{5}$$

We repeat the above procedure to compute the values of $W_{opt}(X)$ and *Score(X)* for all possible values of X. Then we find all peaks of the score function, *Score(X)*, and select those ones which are above a predetermined threshold, T_{opt}. Finally, we use the corresponding indices of the selected peaks and the values of $W_{opt}(X)$ at these indices, to compute the best repeated sub-signals which are the library prototypes.

4. The proposed compression method

The block diagram of the proposed bi-level printed text image compression method which is the main theme of this paper is shown in Fig. 11. As shown, at the first stage, the corresponding CC and coordinates signals of the input bi-level image are computed. In the lossy mode, we employ the second stage; otherwise, we skip it. At this stage, some previously described CC-based procedures are performed. These procedures include boundary smoothing, spot noise removal, filling some inner holes of characters, and quantizing the coordinates signal by steps of 2.

At the third stage of Fig. 11, the best repetitive sub-signals are detected using the proposed algorithm for detecting the repetitive sub-signals. For each group of similar sub-signals, a prototype is determined, for example, by selecting the first-encountered one as the prototype signal. Finally, the library is formed. Each prototype in the library has an index number. Then, each prototype is subtracted from the corresponding matched parts in the CC signal. We call the resulting CC signal as the "residual chain code" (RCC) signal. The start index of each matched sub-signal (in the CC signal) is saved as an index vector of $[Idx, N_1, N_2, \ldots]$ for every prototype. This index vector denotes that the prototype signal with the index *Idx*, is repeated in those parts of the CC signal which start at the points N_1, N_2, etc.

Next, the RCC and prototype signals are converted to their corresponding DCC signal; this conversion is because of the useful property of DCC signals in coding applications. Therefore, at the end of the third stage of Fig. 11, we have a residual DCC (RDCC) signal and some prototype DCC signals as well as the index vectors. These sequences are encoded at the next (fourth) stage of Fig. 11, using a modified multi-symbol QM coder. We have modified the multi-symbol QM coder [10] and used it for encoding the sequences of numbers. In the proposed modification, for encoding any new arrived number, we first subtract it from the rounded average of all previously encoded numbers in the current sequence and then encode the result by the multi-symbol QM coder. This subtraction is meant for producing smaller numbers and a higher compression efficiency; because the smaller the numbers, the smaller is the length of the output bit-stream of the multi-symbol QM-coder [10].

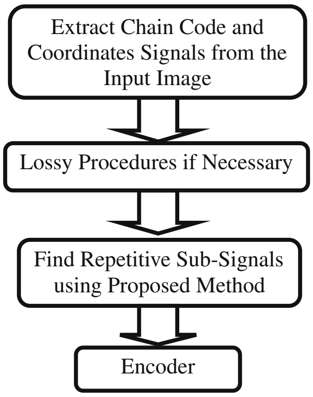

Extract Chain Code and Coordinates Signals from the Input Image
Lossy Procedures if Necessary
Find Repetitive Sub-Signals using Proposed Method
Encoder

Fig. 11. Block diagram of the proposed compression algorithm.

5. Experimental results

The text image database used in this work contains approximately 800 text images which are mostly bi-level printed Farsi and Arabic text images with different spatial resolutions of 100, 200, 300 and 600 dpi and a relatively good quality. By "good quality", we mean that those degradations which change the shape of patterns in a non-uniform manner (such as skew) and thus produce more different prototypes do not exist; thus, common degradations such as spot noise and jaggy artifacts there exist in the images. Although the compression rate of the proposed method is generally decreased in such situations, we expect that the proposed method is not very sensitive to such degradations, because of employing the proposed technique of repetitive sub-signals detection.

We have classified the images into six classes for a better discussion on the compression performance. These classes include mainly graphics, mixed Farsi-graphics, mainly Farsi, mixed Farsi–Arabic, mainly Arabic, and English. It should be noted that the Arabic and Farsi scripts are the same but some Arabic text images, like the one shown in Fig. 12, contain some guide marks for better pronunciation. The mainly Arabic class of images contains such images. Other Arabic text images are included in the mainly Farsi class. For discussion and illustration of the compression performance of the proposed method, a sample text image is chosen for each class such that its compression performance is close to the average compression performance of the images of the corresponding class. These sample images are shown in Fig. 13.

In implementing the proposed method, selecting the interval size of variation of *W* in the second stage has a rather considerable effect on the compression efficiency, because by adjusting this parameter, we determine the permissible length of set of repetitive patterns including sub-patterns, patterns and group of patterns such as words, phrases, and even sentences. Indeed, the larger the value of *W*, the longer is the set of repetitive patterns that could be detected and, usually, the larger is the compression rate, but also the longer is the computation time.

The proposed text image compression method has both lossy and lossless modes. The performance of the lossless mode is tabulated in Table 1 versus the lossless *JBIG2* standard. Although the *JBIG2* standard has not specified any specific encoder [29,11], we have used the *SPM* encoder [10] in the implementation.

Table 1 shows the compression performance (in bit per pixel) of the two methods for the six images of Fig. 13 and the spatial resolutions of 100, 200, 300 and 600 dpi. Also in this table, the relative run time of these methods is compared where the lossless *JBIG2* run time is chosen as the reference. In this table, for each combination of the image number, dpi, and compression method, two numbers are shown. The first (upper) number shows the compression performance in bit per pixel (*bpp*) and the second (lower) number shows the relative run time in our implementation. As can be seen from this table, the run time of the proposed compression method is relatively higher than that of the lossless *JBIG2*.

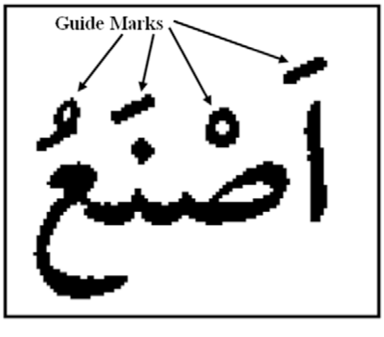


Fig. 12. Some guide marks which exist in some Arabic text images.

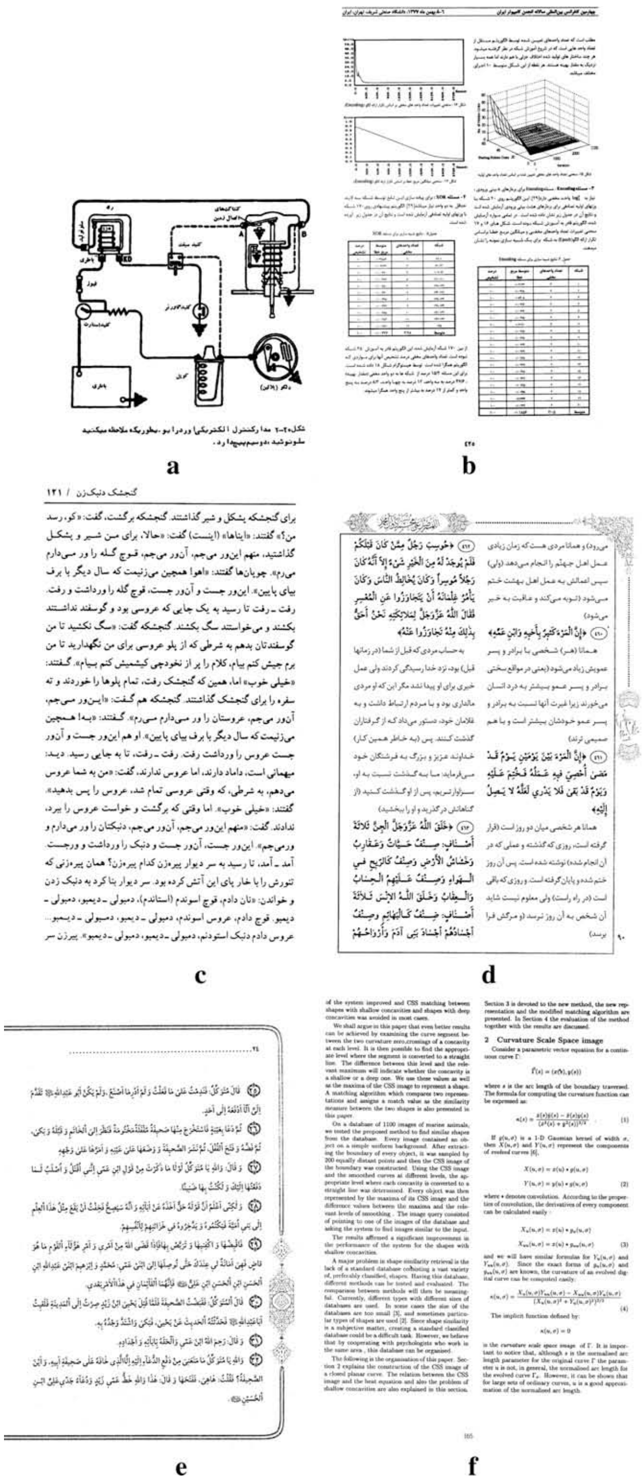


Fig. 13. Some text image samples of the six classes, the image dimensions are at 300 dpi, (a) mainly graphics, 1500 by 1537, (b) mixed Farsi-graphics, 2100 by 3264, (c) mainly Farsi, 1300 by 2059, (d) mixed Farsi-Arabic, 1842 by 2626 (e) mainly Arabic, 1837 by 2667, (f) English, 2178 by 2957.

The compression performance of the mentioned methods for each of the six images is also shown in Fig. 14 versus the four different spatial resolutions of 100, 200, 300 and 600 dpi, respectively. The relative compression rate curves of the proposed

Table 1
The compression performance and the relative run time of the proposed lossless compression method to the lossless *JBIG2* for the six text images.

No	100 dpi		200 dpi		300 dpi		600 dpi	
	Proposed	JBIG2	Proposed	JBIG2	Proposed	JBIG2	Proposed	JBIG2
1	0.0525	0.1102	0.0223	0.0524	0.0124	0.0310	0.0048	0.0133
	0.7	1	0.9	1	1.2	1	2.3	1
2	0.0864	0.1642	0.0381	0.0782	0.011	0.0242	0.0047	0.0114
	1.2	1	1.5	1	1.8	1	3.2	1
3	0.0628	0.2104	0.0212	0.0743	0.0136	0.0491	0.0491	0.0301
	0.9	1	1.2	1	1.4	1	3.7	1
4	0.0171	0.0463	0.0151	0.0431	0.0064	0.0186	0.0045	0.0136
	0.8	1	1.3	1	1.5	1	3.9	1
5	0.0273	0.0943	0.0118	0.0420	0.0071	0.0264	0.0048	0.0191
	1	1	1.7	1	1.9	1	3.5	1
6	0.1552	0.2250	0.0308	0.0493	0.0128	0.0217	0.0068	0.0126
	0.7	1	1.1	1	1.3	1	3.1	1

lossless compression method to the lossless *JBIG2* versus the six image numbers for the four spatial resolutions are shown in Fig. 15.

As can be seen, the highest compression ratio of the proposed method to lossless *JPEG2* occurs for the image classes of mainly Farsi or Arabic. For the English text image, the compression rate of the proposed compression method is still better than that of the lossless *JBIG2*. Also, this ratio for the image class of mixed Farsi–Arabic is not as much as the image classes of mainly Farsi or mainly Arabic, because the text language variation causes to produce more different prototypes with smaller average number of occurrence of prototypes. For the images of the first and second classes, although there are not much repetitive patterns, the compression ratio of the proposed compression method to that of the lossless *JBIG2* is as high as that of the mixed Farsi–Arabic image due to the existence of line-drawings. As can be seen, we have the highest lossless compression ratio for the mainly Farsi or Arabic text images as high as 3.7 times and the lowest one for the English text image as high as 1.7 times, at 300 dpi.

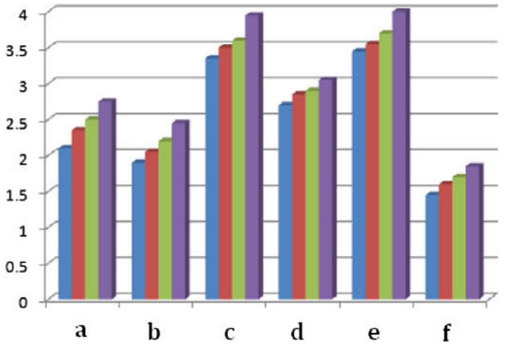


Fig. 15. The bar chart of the relative compression ratio of the proposed lossless compression method to the lossless *JBIG2* versus the six images (a–f) of Fig 13. For each image, four bars are shown which correspond to the four spatial resolutions of 100, 200, 300 and 600 dpi, respectively (from left to right).

For comparison of the lossy version of the proposed compression method, we have applied all previously mentioned lossy pro-

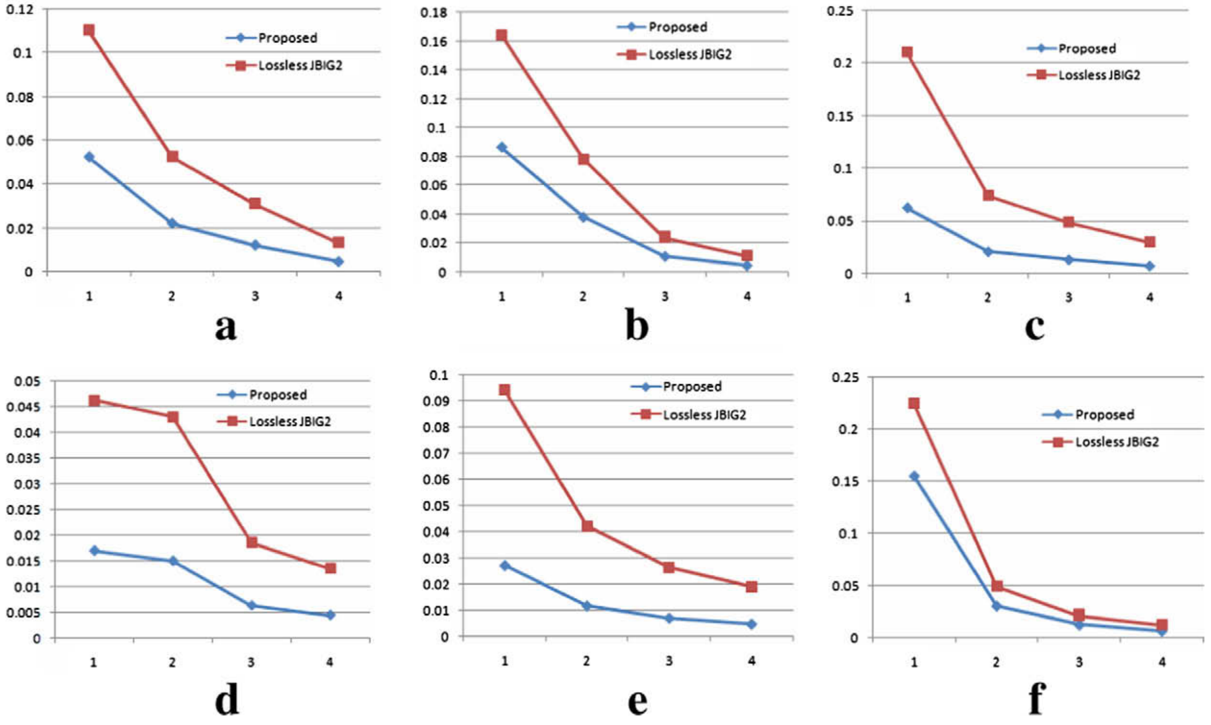


Fig. 14. (a–f) The compression performance curves (in *bpp*) of the proposed lossless compression method and the lossless *JBIG2* versus *dpi* for each of the six images. The vertical axis is in *bpp* and the horizontal axis represents the four spatial resolutions 100, 200, 300 and 600 dpi.

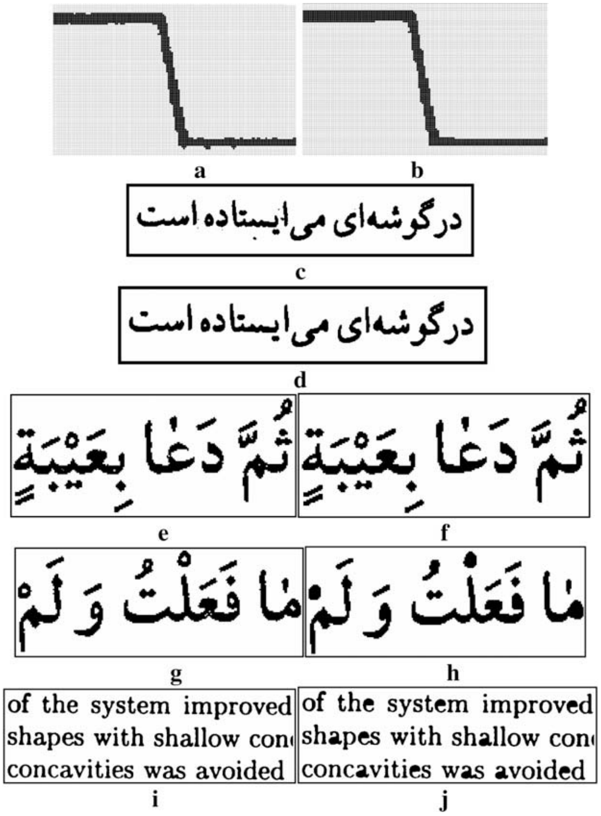

a
b
درگوشه‌ای می‌ایستاده است
c
درگوشه‌ای می‌ایستاده است
d
ثُمَّ دَعَا بِعَيْبَةٍ
e
f
مَا فَعَلْتُ وَلَمْ
g
h
of the system improved
shapes with shallow con
concavities was avoided
i
j

Fig. 16. Some results of the proposed lossy procedures, (a and b) the original and the resulting boundary smoothed sample, (c and d) the original Farsi sample and the result of removing the small patterns, (e and f) the original Arabic sample and the result of removing the small holes and smoothing the boundaries, (g and h) the original Arabic sample and the result of smoothing the boundaries and removing those inner holes whose perimeter is less than a threshold, as can be seen the quality is somewhat increased and the readability is rather well preserved, (i and j) the original Latin sample and the result of removing those inner holes whose perimeter is less than a threshold, as can be seen, contrary to the Arabic sample, the quality and readability is somewhat decreased.

cedures. These procedures include boundary smoothing, spot noise removal, filling some inner holes of characters, and quantizing the coordinate signal values by dividing them by 2 and rounding the results. The results of applying some of these lossy procedures are shown in Fig. 16. An advantage of the proposed lossy procedures is that they usually improve or preserve the image quality and/or readability especially for Farsi and Arabic text images. For example, removing (or equivalently filling) the inner holes of Farsi/Arabic sub-words preserves their readability and quality rather well, as shown in Fig. 16h, but this is not true so much for English sub-words, as shown in Fig. 16j; because the ratio of the average area of the inner holes to the average area of the corresponding sub-words for Farsi and Arabic text images is smaller than that of English text images. We only omit/fill those holes whose perimeters are less than a predetermined threshold, T_{ih} (which is set to 65 at 300 dpi in our experiments). Thus, as shown in Fig. 16j, only some of the holes have been filled.

The performance of the lossy version of the proposed compression method is tabulated in Table 2 versus a lossy version of the *JBIG2*, named *jb2*. The *jb2* method is part of the *DjVu* compound document image compression method. In computing the entries of Table 2, we tried to tune the proposed compression method so that the resulting image quality was near to that of the *jb2* method as much as possible. As can be seen, the relative run times of these methods are not included in this table because their platforms were different. For implementing the *jb2* method, we used the *DjVu Solo*® from the *LizardTech* company.

The lossy compression performance of the mentioned methods for each image of the six classes is shown in Fig. 17 versus the four spatial resolutions 100, 200, 300 and 600 dpi, respectively. Also the relative compression ratio curves of the proposed lossy compression method to the lossy *jb2* versus the six image numbers are shown in Fig. 18 for the four spatial resolutions.

It can be seen that approximately similar results are again obtained. For example, the highest compression ratio of the proposed method to that of the *jb2* occurs for the mainly Farsi or Arabic text images. In addition, the compression ratio of the proposed lossy compression method still is higher than that of the lossy *jb2* method for the English text image. The boundary smoothing technique has a considerable effect on increasing the compression ratio, especially, for the mainly graphics, mainly Farsi and mainly Arabic classes. Also as can be seen, we have again the highest lossy compression performance for mainly Farsi or Arabic text images as high as 4.3 times and the lowest one for the English text image as high as 2 times at 300 dpi.

6. Conclusions

In this paper, a lossy/lossless compression method for bi-level printed text images was proposed for archiving purposes. For this, we proposed a new 1-D pattern matching technique in the chain code domain, where the library prototypes are 1-D chain code signals. The proposed method used some chain code-based procedures, in the lossy mode, to simultaneously increase the compression ratio and image quality and/or readability.

In the Farsi/Arabic script, contrary to the printed Latin script, letters normally attach together and produce many different patterns. A relatively high number of these patterns are fully/partially similar to each other. In addition, some printed Latin text images contain undesired touching characters due to factors such as resampling. The touching/attaching characters are more important for Farsi and Arabic text images, because they occur very frequently. The compression performance of the conventional PM technique is decreased in all of the above situations, because more different prototypes with smaller average number of occurrence are produced.

Table 2
The compression performance of the proposed lossy compression method and the lossy *jb2* method for the six text images.

No	100 dpi		200 dpi		300 dpi		600 dpi	
	Proposed	jb2	Proposed	jb2	Proposed	jb2	Proposed	jb2
1	0.0273	0.0752	0.0128	0.0378	0.0076	0.0235	0.0028	0.0093
2	0.0546	0.1228	0.0226	0.0542	0.0065	0.0162	0.003	0.0084
3	0.0364	0.1438	0.0132	0.0543	0.0091	0.0381	0.0049	0.0215
4	0.0091	0.0313	0.008	0.0287	0.0036	0.0134	0.0024	0.0091
5	0.0166	0.0663	0.0075	0.0313	0.0043	0.0187	0.0029	0.0131
6	0.0887	0.1553	0.0194	0.0368	0.0078	0.0159	0.0041	0.0090

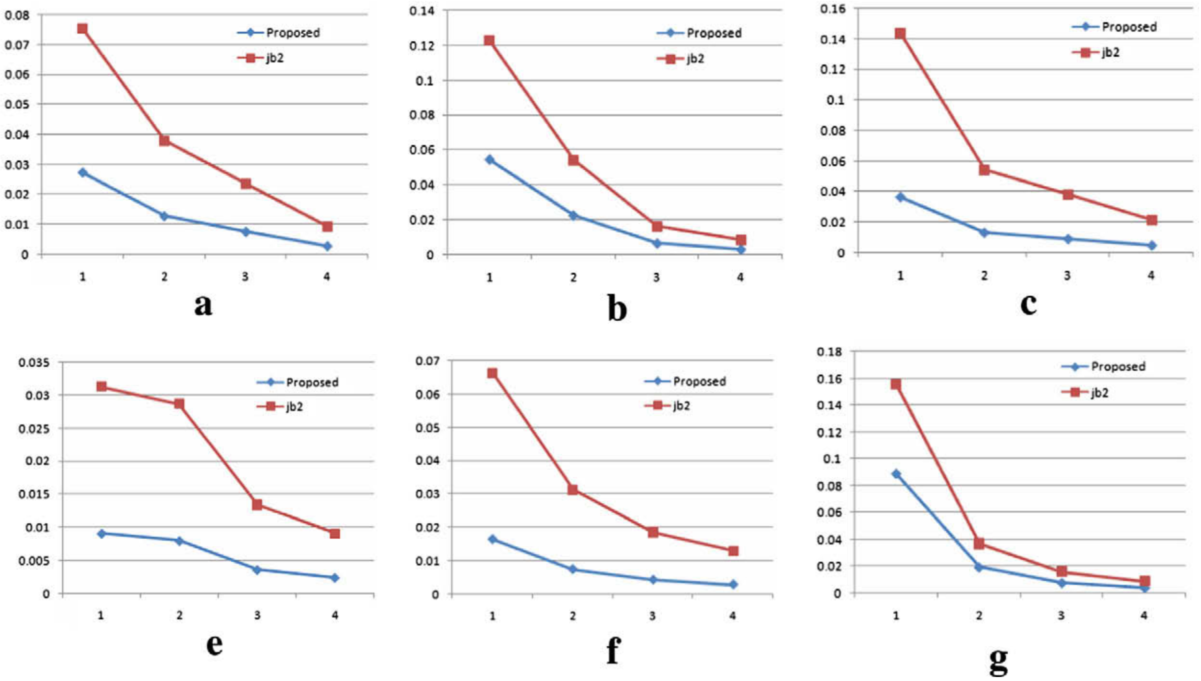


Fig. 17. (a–f) The compression performance curves (in *bpp*) of proposed lossy compression method and *jb2* versus *dpi* for each of the six images. The vertical axis is in *bpp* and the horizontal axis represents the four spatial resolutions 100, 200, 300, and 600 dpi.

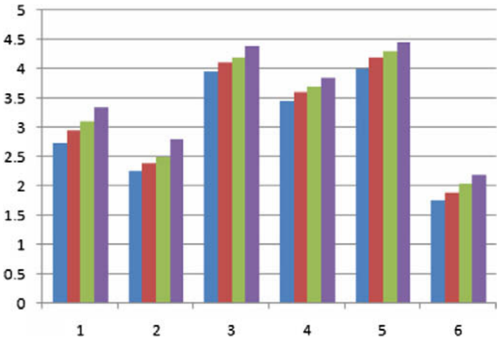


Fig. 18. The bar chart of the relative compression ratio of the proposed lossy compression method to the lossy *jb2* versus the six images (a–f) of Fig 13. For each image, four bars are shown which correspond to the four spatial resolutions 100, 200, 300 and 600 dpi, respectively (from left to right).

The proposed compression method can detect such similar patterns and exploit them to reduce the library size and consequently increase the compression ratio. It uses the proposed technique of repetitive sub-signal detection in order to detect the mentioned fully/partially similar patterns.

The compression performance of the proposed method was compared to that of the state-of-the-art compression methods, for Farsi, Arabic and English text images. Experimental results showed that the compression performance of the proposed method was considerably better than that of the best existing bi-level text image compression methods as high as 1.8–4.2 times in the lossy case and 1.6–3.8 times in the lossless case at 300 dpi. The highest compression ratios were achieved for Farsi and Arabic text images.

References

[1] A.C. Ascher, G. Nagy, A means for achieving a high degree of compaction on scan-digitized printed text, IEEE Trans. Comput. 23 (11) (1974) 1174–1179.

[2] T.C. Bell, J.G. Cleary, I.H. Witten, Text Compression, Prentice Hall, Englewood Cliffs, NJ, 1990.

[3] L. Bottou, P. Haffner, P.G. Howard, P. Simard, Y. Bengio, Y. LeCun, High quality document image compression with 'DjVu', Journal of Electronic Imaging 7 (1998) 410–425.

[4] N.F. Brickman, W.S. Rosenbaum, Word autocorrelation redundancy match (WARM) technology, IBM J. Res. Dev. 26 (1982) 681–686.

[5] Y. Fisher (Ed.), Fractal Image Compression, Springer-Verlag, New York, 1995.

[6] H. Freeman, Techniques for the digital computer analysis of chain-encoded arbitrary plane curves, Proc. Nat. Electron. Conf. (1961) 421–432.

[7] A. Gersho, R. Gray, Vector quantization and signal compression, Kluwer, Norwell, MA, 1992.

[8] M.J. Holt, C.S. Xydeas, Recent developments in image data compression for digital facsimile, ICL Tech. J. (1986) 123–146.

[9] M.J. Holt, A fast bi-level template matching algorithm for document image data compression, in: J. Kittler (Ed.), Pattern Recognition, Springer-Verlag, Berlin, Germany, 1988, pp. 230–239.

[10] P.G. Howard, Text image compression using soft pattern matching, The Computer Journal 40 (2/3) (1997) 146–156.

[11] P.G. Howard, F. Kossentini, F. Forchhammer, W.J. Ruchlidge, The Emerging JBIG2 Standard, IEEE Trans. Circuits Syst. Video Tech. 8 (7) (1998) 838–848.

[12] ISO/IEC International Standard 11544, Progressive Bi-level Image Compression, JBIG, ITU-Recommendation T.82, 1993.

[13] B. Jahne, Digital Image Processing, sixth ed., Springer-Verlag, Berlin, Heidelberg, 2005.

[14] O. Johnsen, J. Segen, G.L. Cash, Coding of two-level pictures by pattern matching and substitution, Bell Syst. Tech. J 62 (8) (1983) 2513–2545.

[15] T. Kanungo, R.M. Haralick, I.T. Phillips, Global and local document degradation models, in: Proceedings of the International Conference on Document Analysis and Recognition, 1993, pp. 730–734.

[16] O.E. Kia, D.S. Doermann, A. Rosenfeld, R. Chellappa, Symbolic compression and processing of document images, Computer Vision and Image Understanding 70 (3) (1998) 335–349.

[17] O.E. Kia, D.S. Doermann, Residual coding in document image compression, IEEE Trans. Image Process. 9 (6) (2000) 961–969.

[18] A. Moffat, Two level context based compression of bi-level images. in: Proceedings of IEEE Data Compression Conference, 1991, pp. 382–391.

[19] T. Pavlidis, Algorithms for Graphics and Image Processing, Computer Science Press, Maryland, 1982.

[20] W.K. Pratt, P.J. Capitant, W.H. Chen, E.R. Hamilton, R.H. Wallis, Combined symbol matching facsimile data compression system, Proc. IEEE 68 (7) (1980) 786–796.

[21] I.M. Pu, Fundamental Data Compression, Butterworth-Heinemann, 2006.

[22] D. Salomon, Data Compression, The Complete Reference, fourth ed., Springer-Verlag, London, 2007.

[23] D. Salomon, A Concise Introduction to Data Compression, Springer-Verlag, London, 2008.

[24] K. Sayood (Ed.), Lossless Compression Handbook, Academic Press, 2003.

[25] I.H. Witten, T.C. Bell, H. Emberson, S. Inglis, A. Moffat, Textual image compression: two-stage lossy/lossless encoding of textual images, Proceedings of the IEEE 82 (6) (1994) 878–888.

[26] I. Witten, A. Moffat, T.C. Bell, Managing Gigabytes: Compressing and Indexing Documents and Images, Second ed., Academic Press, 1999.
[27] Y. Yang, H. Yan, D. Yu, Content-lossless document image compression based on structural analysis and pattern matching, Pattern Recognition 33 (2000) 1277–1293.
[28] Y. Ye, P. Cosman, Dictionary design for text image compression with JBIG2, IEEE Trans. Image Process. 10 (6) (2001) 818–828.
[29] Y. Ye, P. Cosman, Fast and memory efficient text image compression with JBIG2, IEEE Trans. Image Process. 12 (8) (2003) 944–956.