Journal of King Saud University – Computer and Information Sciences (2017) **29**, 29–38

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

A new algorithm for skew correction and baseline detection based on the randomized Hough Transform

Abdelhak Boukharouba *

Faculté des Sciences et de la Technologie, Département d'Electronique et de Télécommunications, Université 8 Mai 1945 Guelma, BP 401 Guelma 24000, Algeria

Received 15 July 2015; revised 6 February 2016; accepted 11 February 2016
Available online 29 March 2016

KEYWORDS

Document lower edge extraction;
Labeling algorithm;
Randomized Hough transform;
Skew correction;
Baseline detection

Abstract The proposed technique is based on the detection of the lower baselines of the text lines of Arabic documents. As the lower baseline pixels belong to the lower edge of the word images, we first locate vertically the black–white transitions at the black pixels where the resulting image would emphasize the baselines of the text. Once the skew angle is determined using a randomized Hough transform, the baselines are extracted using *y*-intercept histogram. This algorithm can also contribute significantly for text line extraction from skewed document images for many languages.

1. Introduction

Document image processing has become a very important technique for different fields of computer applications in recent years. The conversion of the paper-based documents into electronic versions for storage, retrieval, automatic processing, transmission, etc... is very important (Yin, 2001). The document processing can be divided into two phases: document analysis and document understanding (Tang et al., 1996). Document analysis consists essentially of the structural layout analysis and the information type of the documents, whereas document understanding is defined as the process of recognizing the contents and meanings of each block and constructing an index structure for efficient retrieval of the document. Document analysis consists of three phases: the preprocessing phase, the block segmentation phase, and the block classification phase (Tang et al., 1996; O'Gorman and Kasturi, 1995). The preprocessing phase consists of digitization, noise removal, and skew correction of the documents where different methods are discussed and compared (Saba et al., 2011; Rehman and Saba, 2011).

In our work, we concentrate on the skew correction and baseline detection of Arabic documents. In Arabic manuscript, most characters connect to each other on the lower baseline and consequently our approach is based on the detection of lower baselines of document words. The baseline represents the main orientation of a text line and it is often a preliminary condition for subsequent algorithms, such as segmentation and feature extraction. Various techniques of baseline estimation are used and reported. For efficient skew detection of printed

* Tel.: +213 777 08 32 32; fax: +213 37 20 72 68.
E-mail address: boukharouba_abdelhak@hotmail.com.
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2016.02.002
1319-1578

documents, Papandreou et al. (2014) combined reinforced vertical and enhanced horizontal projection profiles with the use of the minimum bounding box area criterion, whereas in Shafii and Sid-Ahmed (2015) the skew is detected by minimizing the area of the axis-parallel bounding box where this algorithm is script and content independent.

The traditional projection method is a simple solution for baseline detection (El-Hajj et al., 2005; Parhami and Taraghi, 1981), which counts the black pixels line by line and the position of the baseline is indicated by the maximum number of pixels. But there are some cases where this approach does not work well, mainly for skewed script. Al-Rashaideh (2006) proposed an algorithm that depends on the iteration with angle. To determine the skew angle, the projection profile is applied to rotated image with a number of angles. The angle corresponding to the higher peak is considered as the skew angle. Pechwitz and Maergner (2002) proposed a skeleton-based technique to detect the baseline of Arabic handwriting where the skeleton is approximated by piecewise linear curve and the baseline is the line that best fits the edges. Another approach to detect Arabic handwriting baseline according to the word contour representation has been proposed (Farooq et al., 2005). It is based on a two-step linear regression applied to the local minima points of word contour, whereas Burrow (2004) presented an approach based on the Principle Components Analysis for detecting Arabic handwriting baseline. A new two-stage method for estimating and correcting the baseline of handwritten subwords in Arabic and Farsi text lines has been presented (Ziaratban and Faez, 2008). Finally, Boubaker et al. (2009) proposed an algorithm to detect straight or curved baselines for short Arabic handwritten writing.

When a document is fed to a scanner, a small skew is inevitable. This affects the accuracy of subsequent algorithms for recognition systems. To deal with this problem we have employed skew correction procedure, which determines a skew angle from a page of text lines.

For Arabic script, most of the characters have a lot of pixels on the lower baselines of the document image. Consequently, we propose to calculate the skew angle as the slope of lower baselines of the text lines. To improve line detection accuracy and reduce the data amount we apply a randomized Hough transform only to the relevant connected pixels of the lower edges of document images.

This paper is organized as follows. Section 2 describes the Arabic script characteristics. Section 3 details the steps of skew correction and baseline detection algorithm. In Section 4, experimental results are presented, and some analyses are also given. Some conclusions are given in Section 5.

2. Arabic script characteristics

Arabic is written by more than 250 million people, and by nature, Arabic text is inherently cursive both in handwritten and printed forms and is written horizontally from right to left. The alphabet contains 28 different characters in which 16 of them have diacritics. The diacritics can be above, inside, or below a character and accordingly form different semantics and pronunciation. Different Arabic characters may have exactly the same shape, and are distinguished from each other only by the addition of one out of five diacritics. These are normally one, two or three dots, "hamza" or "medda". For example, the three different characters (ب, ت, ث) have the same main shape but different diacritics. In contrast to Latin, Arabic characters are not divided into upper and lower case categories. Instead, an Arabic character might have several shapes depending on its relative position in a word (beginning, middle, end or alone), for example (ع, عـ, ـعـ, ـع). Table 1 shows a complete set of Arabic characters in all their forms depending on their position in a word. There are six characters which cannot be connected to the left. These characters will be called last characters of sub-words (ا, د, ذ, ر, ز, و). Arabic writing is cursive and words are separated by spaces. However, a word can be divided into smaller units called sub-words (a portion of a word including one or more connected characters). Most characters connect to each other on the writing line, which we call baseline. Baseline detection is a very important stage of Arabic character recognition systems. Fig. 1 describes the main characteristics of Arabic script.

Table 1 Arabic alphabet in all its forms (end form EF, middle form MF, beginning form BF, and isolated form IF).

EF	MF	BF	IF	EF	MF	BF	IF
ض	ضـ	ضـ	ض	ـا		أ	
ط	طـ	طـ	ط	ب	بـ	بـ	ب
ظ	ظـ	ظـ	ظ	ت	تـ	تـ	ت
ع	عـ	عـ	ع	ث	ثـ	ثـ	ث
غ	غـ	غـ	غ	ج	جـ	جـ	ج
ف	فـ	فـ	ف	ح	حـ	حـ	ح
ق	قـ	قـ	ق	خ	خـ	خـ	خ
ك	كـ	كـ	ك	د			د
ل	لـ	لـ	ل	ذ			ذ
م	مـ	مـ	م	ر			ر
ن	نـ	نـ	ن	ز			ز
ه	هـ	هـ	ه	س	سـ	سـ	س
و			و	ش	شـ	شـ	ش
ي	يـ	يـ	ي	ص	صـ	صـ	ص

3. Skew correction and baseline detection algorithm

If the original document is grayscale or color, the document image must be binarized. Next, the binary image is smoothed by removing isolated pixels and filling the gaps in the image.

The approach is based on the detection of lower baselines of document words. As the lower baseline pixels belong to the lower edge of the word images, we first determine the lower edge pixels. To do this, we scan the binarized image vertically column by column and each black–white transition is located at the black pixel and the other black pixels will be transformed into white pixels. The resulting image would highlight the baselines of the text. Fig. 2b shows the resulting image by locating the black–white transition vertically on Fig. 2a.

At the same time of the lower edge detection step, we form a segmentation of this edge image into continuous curves of foreground pixels using labeling algorithm. The main idea of segmentation algorithm is that pixels of different black–white transitions in vertical direction should be separated from each other. To extract the connected pixels we use a simple sequential procedure which compares successive pixels of an edge image to determine whether black pixels in any edge are

Figure 1 Main characteristics of Arabic script.

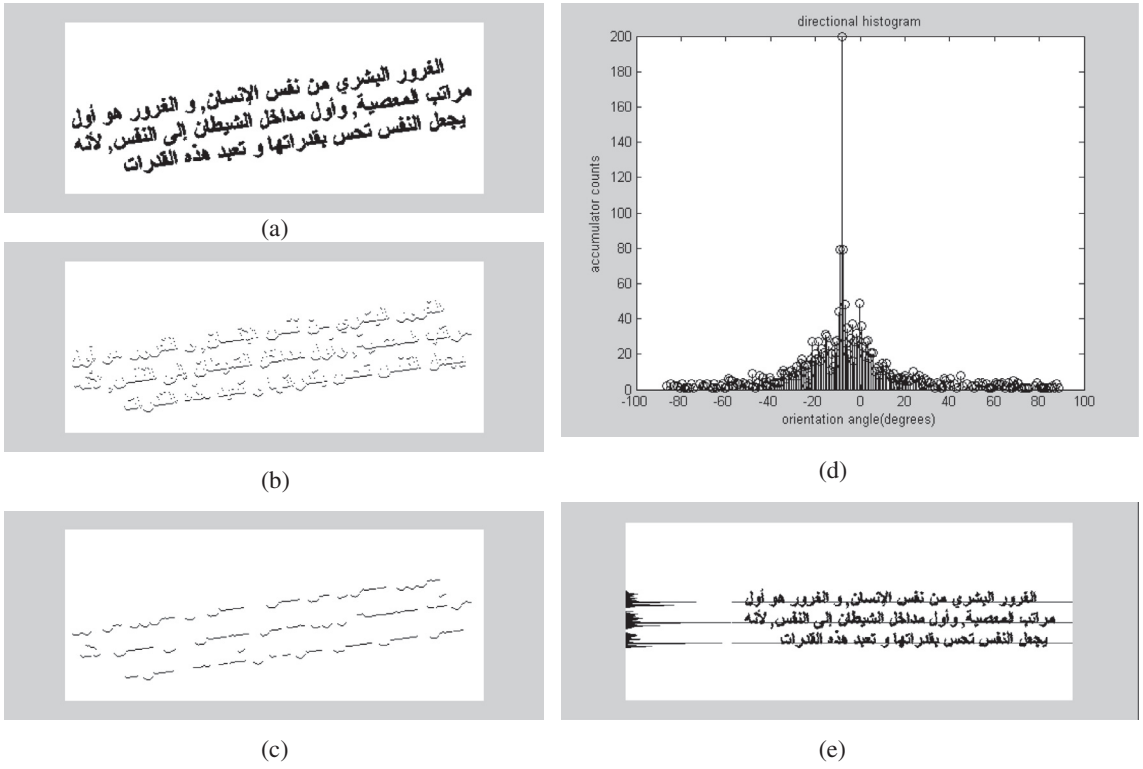


Figure 2 Example illustrating the skew correction and baseline detection algorithms: (a) skewed text image; (b) black–white transition location; (c) resulting filtered edge image; (d) directional histogram of image (c); (e) lower baselines of corrected text image.

connected together. Consequently using this algorithm, an edge image is extracted and segmented into a number of separated curves.

After the steps of edge segmentation, isolated pixels are removed and the mean length b of the resulting curves is also computed. Next, curves having length greater than or equal to b, are retained. By thresholding at b, small edges of components like dots, punctuation marks and characters without lower baseline are mostly filtered out. In this way the small segments were filtered out before the skew estimation step, see Fig. 2c. Consequently, we have considerably reduced the data amount that will be used in the skew and baseline detection. Next, the randomized Hough transform is applied to the pixels of resulting curves to detect the straight edges.

Traditional Hough transform is a voting procedure that needs to cumulate the counts of different parameters for the set of all straight lines going through a given black pixel in the image. This transformation needs a large storage requirement and expensive computation cost depending on the number of black pixels and the number of the parameters to be cumulated. In the standard Hough transform, two parameters are needed to describe a straight line as $\rho = x\cos\theta + y\sin\theta$. However, only one line can pass through two pixels and one parameter needs to be cumulated if we choose two pixels at a time. Thus, we can just cumulate the votes for the angle θ to detect straight lines, which reduce considerably the number of votes in the parameter space.

In order to limit the number of pixel pair combinations and ensure rapid computation of the Hough transform, we choose to use a small random samples set rather than the full data set (Xu et al., 1990).

In the randomized Hough transform of our algorithm, the accumulator space is constructed as a data structure, built dynamically during accumulations, instead of a predefined accumulator array. The details are presented as follows. Two edge pixels are randomly selected and the angle between the line joining them and X-axis $\theta = arctg\left(\frac{y_k - y_i}{x_k - x_i}\right)$ is calculated. The resulting angle is put into a vector defining the angle θ and an integer accumulator. The process of pixel selection and angle deduction is repeated, and for each new solution if

the required vector already exists (within some specified tolerance) then its accumulator value is increased by one, otherwise a new vector is created and its count set to unity. A line is detected when the count in an accumulator reaches some predefined threshold.

Fig. 2d shows the directional histogram of edge image of Fig. 2c. In this paper, the randomized Hough transform tolerance for opening a new accumulator cell is set to 0.2° in θ and the line detection threshold to 200 counts. Once the skew angle is determined, the image is then rotated by the same angle in the opposite direction so that the text lines become horizontal. Fig. 2a and e shows an Arabic text before and after skew correction respectively. The baseline is then obtained after skew correction using a horizontal projection of the lower edge image without segmentation process mentioned above. The rows which have the maximum number of pixels are taken as the lower baselines of the text. Fig. 2e shows the horizontal projection of the lower edge image and the lower baselines of the corrected text image. If the skew angle equals zero then we make no correction and the baselines are determined using the horizontal projection of the same resulting filtered edges.

Moreover, this algorithm can be also employed for baseline detection without skew correction to decrease the consuming time and avoid degradation due to image rotation especially for slightly skewed documents. To do this, we apply the same algorithm to find the baseline direction θ. Next we can determine the y-intercept $c = y - mx$ using voting-based method where $m = tg(\theta)$ is the slope of baseline/baselines and (x, y) are the coordinates of curves' pixels. The resulting intercept is put into a vector, built dynamically during accumulations, defining the y-intercept and an integer accumulator. For each new y-intercept, if the required vector already exists (within some specified tolerance) then its accumulator value is increased by one, otherwise a new vector is created and its count set to unity. The process is repeated until no more foreground pixels are found. Finally, the highest peak defines directly the y-intercept of the most dominant baseline and so on. Note that the lower edge image does not contain vertical segments, thus we have no problem to compute the y-intercept. The tolerance for opening a new accumulator cell is set to 2 pixels.

Fig. 3 illustrates the baseline detection algorithm without skew correction. Here Fig. 3c shows the directional histogram of resulting filtered edge image of Fig. 3b where the highest peak corresponds to the angle $\theta = -13.17°$. Fig. 3d shows the y-intercept histogram corresponding to $\theta = -13.17°$ and the highest peak of each cluster corresponds to one y-intercept. From this histogram we distinguish four y-intercepts 364.70, 445.80, 526.50, and 606.53 where their corresponding baselines are shown in Fig. 3e.

To summarize, the main steps of this algorithm are:

Step 1. *Lower edge determination and segmentation*Scan the binarized image vertically column by column and locate each black–white transition at the black pixel. At the same time, form a segmentation of this edge image into continuous curves using labeling algorithm as described above.

Step 2. *Small edge filtering*Remove the isolated pixels and compute the mean length b of the resulting curves. Retain only the curves whose lengths are greater than or equal to b.

Step 3. *Skew angle detection*Select randomly two edge pixels and calculate the skew angle: $\theta = arctg\left(\frac{y_k - y_l}{x_k - x_l}\right)$.Put the resulting angle into a vector defining the skew angle and an integer accumulator.Repeat the pixel selection and angle deduction process until the accumulator reaches a predefined threshold.The highest peak defines directly the skew angle.

Step 4. *Baseline estimation with skew correction*Rotate the image by the same angle in the opposite direction so that the text lines become horizontal.Determine the baselines using a horizontal projection of the lower edge image.The rows which have the maximum number of pixels are taken as the baselines of the text.

Step 5. *Baseline estimation without skew correction*Determine the y-intercept $c = y - mx$ using voting method where $m = tg(\theta)$ is the slope of the baseline and (x, y) are the coordinates of curves' pixels.Put the resulting intercept into a vector, built dynamically during accumulations, defining the y-intercept and an integer accumulator.Repeat the process until no more foreground pixels are found.The highest peak defines directly the y-intercept of the most dominant baseline and so on.

4. Results and discussion

The performance of the present algorithm has been tested separately with samples of handwritten and printed texts. We have collected a variety of printed Arabic documents consisting of magazines, books and newspaper. The handwritten texts are collected from students' copybooks. The images are digitized by a scanner at a resolution of 300 dpi. The proposed method has been also tested on IFN/ENIT (Pechwitz et al., 2002) database consisting of 26,459 Arabic words handwritten by 411 different writers.

In order to estimate the skew angle of the text image, we compute the difference between the first two biggest values of accumulator bins. If the difference exceeds certain predetermined threshold T_1, the highest peak defines directly the skew (orientation) of the text as displayed in Fig. 1. Otherwise, the baseline is not accurate and cluster modes provided by the direction histogram constitute a set of baselines.

If there is only one evident cluster then there is only one baseline where the pixels lying on it are not exactly collinear. The skew angle is to be chosen as the average of the directions whose accumulator bins exceed a threshold T_2. In the following experiments T_2 is set to 100 counts. If there is more than one cluster mode in the direction histogram, each cluster mode may correspond to a local subword baseline. The local skew angle is the average value of the bins exceeding a threshold T_2. As the local skew angles are different from each other, the skew of estimated global baseline is approximated by the average of the local skews.

In the experiments, 861 pages from different books and many samples of Arabic document images taken from Algerian newspapers are considered. Samples of skewed images taken from Algerian newspapers are shown in Fig. 4 (a)–(b) and other samples taken from books in Arabic are shown in Fig. 4c. Fig. 4(d)–(f) shows their skew corrected documents.

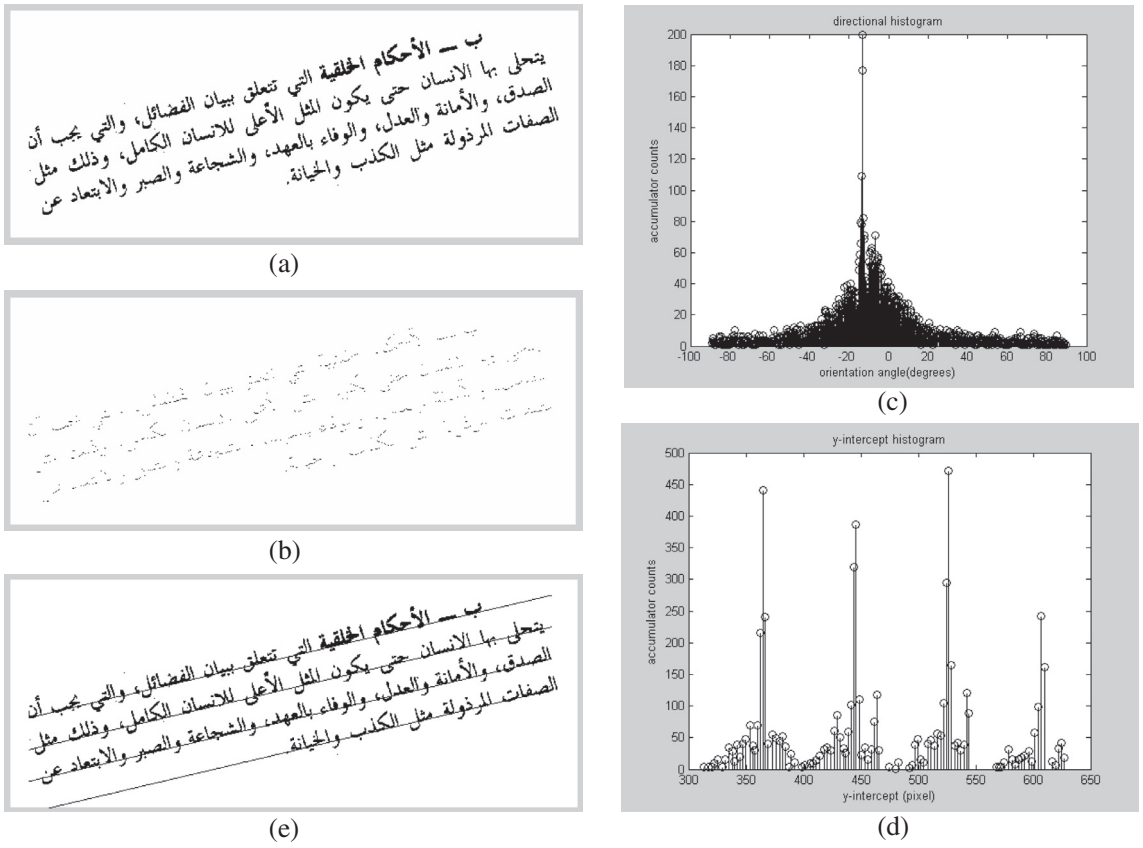


Figure 3 Baseline detection algorithm without skew correction; (a) original document image; (b) filtered edge image; (c) directional histogram; (d) y-intercept histogram; (e) baselines (text lines) of original document.

Figure 4 Samples of printed document images: (a)–(b) skewed document images from Algerian newspapers; (c) skewed document image from Arabic book.;(d)–(f) skew-corrected document images of the images (a)–(c) respectively.

Moreover, experiments on Printed texts with different fonts (Arabic Transparent, Simplified Arabic and Traditional Arabic), of different sizes and styles (bold and italic) are also considered. The orientations of the printed documents are all correctly identified (100% success rate), and these documents are then successfully skew-corrected by the detected skew angle. This is justified because, in the most commonly used fonts in printed Arabic documents, the lower edges emphasize clearly the baselines.

For handwritten texts, the performance is tested on samples taken from student's copybooks where the script is guided (the copybooks' sheets are blue striped). Total number of text lines in these sample images is about 1400. Each line of handwritten text consists of 10 words on average. The present technique shows also 100% success rate in extracting the skew angle of all these text lines. Fig. 5 shows a few samples of skewed and skew-corrected document images taken from students' copybooks.

Once the skew angle θ is computed, we can successfully detect the lower baselines of printed/handwritten documents without skew correction using y-intercept histogram described above. Consequently, this algorithm can be used to detect the

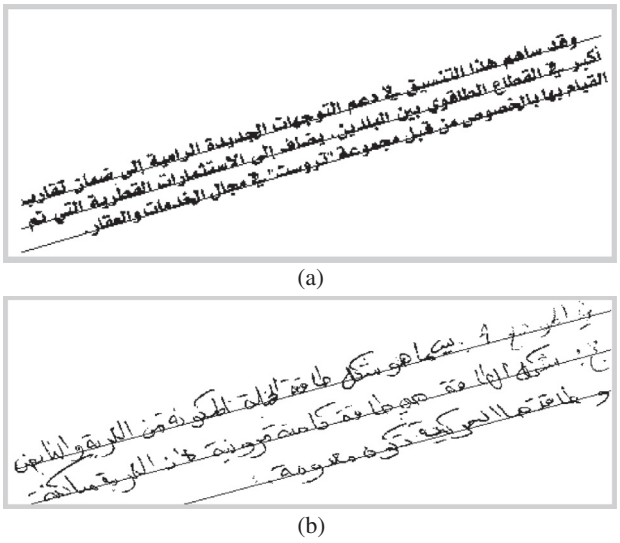


Figure 6 Baseline detection and text line extraction without skew correction: (a) printed document; (b) handwritten document.

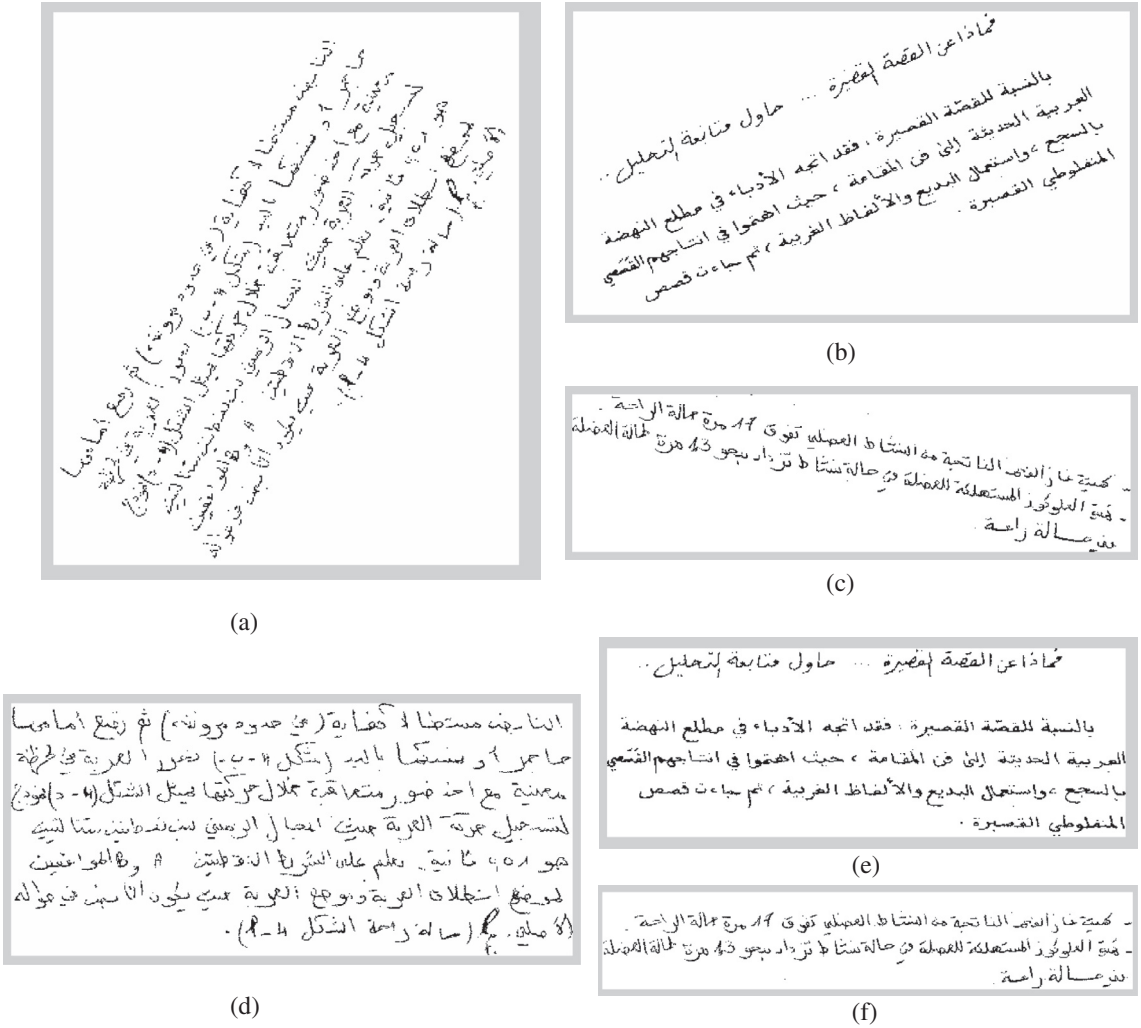


Figure 5 Samples of handwritten text taken from students' copybooks: (a)–(c) skewed document images; (d)–(f) skew-corrected document images (a)–(c) respectively.

baselines and extract the text lines from Arabic documents as shown in Fig. 6.

This technique shows also good results in detecting baselines for short Arabic handwritten writing where the low edges emphasize clearly the baselines even for a single word. For IFN/ENIT database, we have detected the baselines of Tunisian town/village names without skew correction using a directional and y-intercept histograms described above. Fig. 7 shows some examples from the IFN/ENIT database and their corresponding baselines where each estimated baseline is the true one or very close to it.

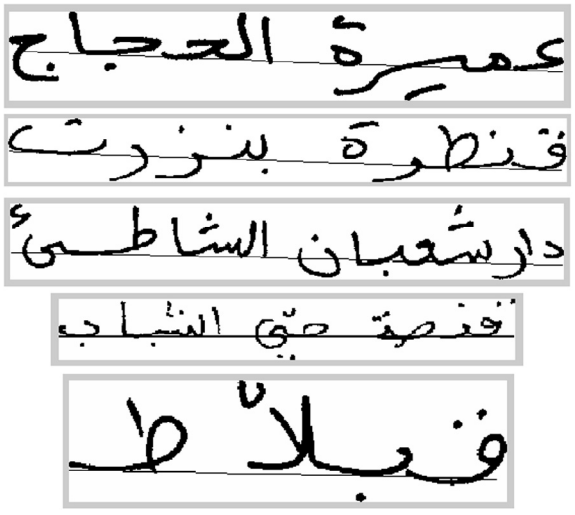
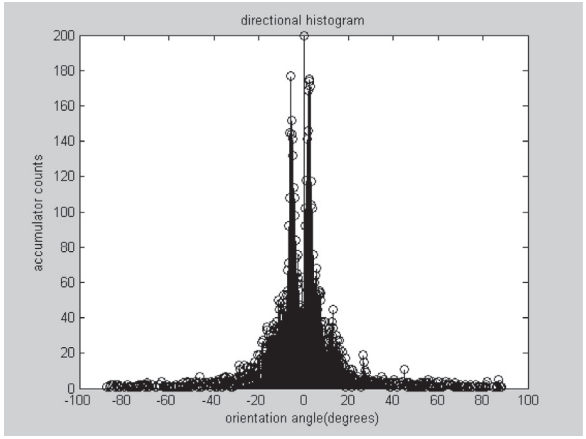


Figure 7 Examples from the IFN/ENIT database and their corresponding baselines.



(a)

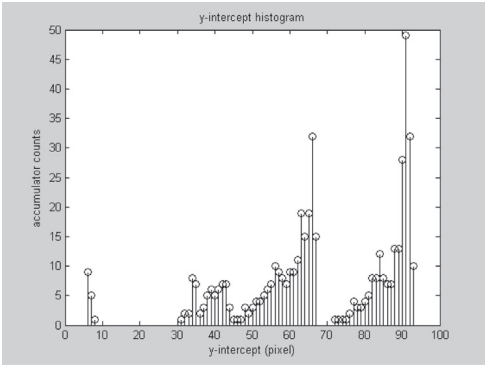
(b)

Figure 8 An example where only one cluster corresponds to the true baseline: (a) directional histogram; (b) true baseline.

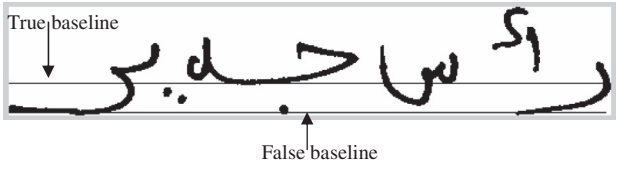
For most of the cases in IFN/ENIT database, this algorithm works better in detecting the baseline. However, it fails in some cases where the detected angle and/or y-intercept, which correspond to the highest peak's clusters, are not the angle and/or y-intercept of the true baseline.

After analyzing the errors committed by this system, we observe that the confusion is caused by ascenders and descenders, which alter the distribution of the pixel coordinates in such a way that false and true baselines occur together. These ascenders and descenders affect the skew angle and/or the y-intercept. From the experiment it is observed that the false baselines occur when the slant angle is larger than 4°. In this case the clusters whose angles are larger than 4° are not considered in skew angle evaluation. But if there isn't any cluster with angle less than 4°, we retain the most significant cluster as solution. Fig. 8 shows one example with two clusters where only the first cluster (angle = 1.23°) corresponds to the true baseline.

The second type of error occurs when the skew angle is correctly estimated but the y-intercept isn't. As we know, the baseline should appear in the middle zone of the image; therefore, the algorithm must find peaks in this area. Fig. 9 shows



(a)

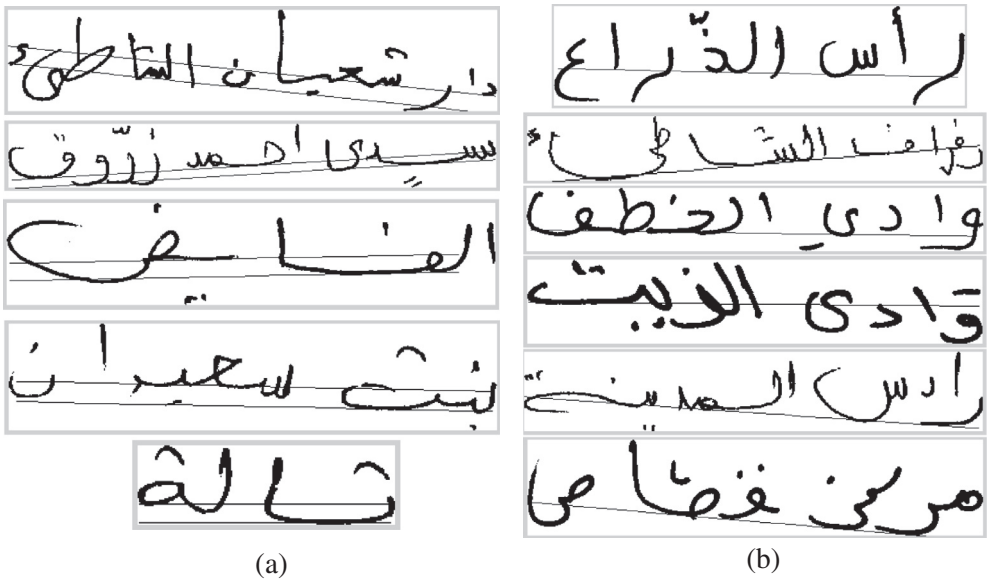


(b)

Figure 9 Example where the correct y-intercept corresponds to the second most significant cluster: (a) y-intercept histogram; (b) true and false baselines.

Table 2 Comparisons to other methods in the literature tested on the IFN/ENIT database.

Method	Baseline estimation accuracy (%)
PCA (Burrow, 2004)	82
Hough Projection (Pechwitz and Maergner, 2003)	83
Skeleton-based (Pechwitz and Maergner, 2003)	88
Proposed method	95



(a) (b)

Figure 10 Examples where baselines are maladjusted to the true baseline: (a) examples where subwords have different baselines; (b) examples where one part of city name corresponds perfectly to the true baseline but the remaining part is lying along the lower edge of the isolated characters.

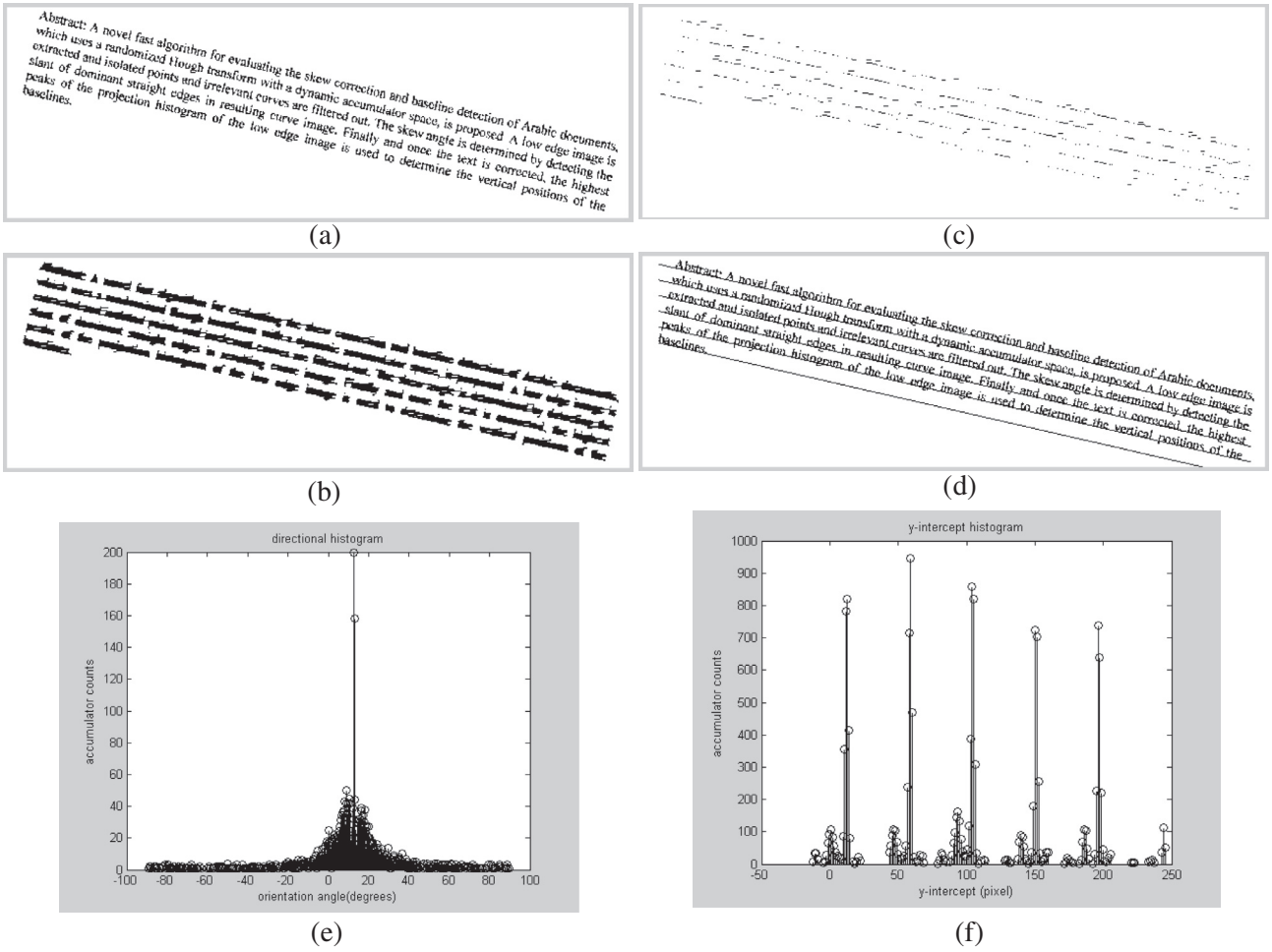


Figure 11 Example illustrating the text line extraction algorithm for English text: (a) skewed text image; (b) smoothed image; (c) filtered low edge of smoothed image; (d) extracted baselines (text lines); (e) directional histogram; (f) *y*-intercept histogram.

an example where the skew angle is correctly estimated but the true y-intercept corresponds to the second most significant cluster in the y-intercept histogram.

The third type of confusion occurs when both the skew angle and the y-intercept are incorrectly estimated. In this case the solution is the combination of the two previous solutions. With these improvements, we achieved a success rate of 95%. These results show that our algorithm provides good performance compared to results reported in the literature. Table 2 shows comparisons to other methods tested on the IFN/ENIT database.

Among the frequent confusions, which are counted in error rate are the following:

Some confusion errors are due to city names which have subwords with the same slant but with different baseline's y-intercepts. These baselines may be different from each other. As solution the y-intercept of the global baseline is chosen as the mean of the all y-intercepts of the baselines under question. Fig. 10a shows examples where words (subwords) have different baselines. Another type of error is the confusion of the relevant baseline with the lower limit line of words composed of isolated characters as 'ز, ر, ن, و'. Fig. 10b shows that one part of city names corresponds perfectly to the true baseline but the remaining part is lying along the lower edge of the isolated characters.

Compared to the approaches in the literature, our algorithm processes less number of pixels and its results seem more accurate (as shown in Table 2). The other methods require either multiple steps such as the projection histogram, which is based on the image rotation (Al-Rashaideh, 2006) or more complex calculations as Standard Hough transform and skeleton based methods (Pechwitz and Margner, 2002, 2003). Compared to the PCA method (Burrow, 2004) our algorithm requires less computational time especially if the lower edges emphasize the lower baseline; a smaller amount of lower edge pixels is enough and the baseline is directly detected without rotating the image. Furthermore, we can apply all mentioned approaches to the lower edge image rather than the entire image. Thus, we can reduce considerably the computational complexity of these algorithms.

The advantage of the technique is that it can be also applied to documents in other languages where the straight edges are actually present in the image.

For the languages based on the letters of the Latin alphabet (English, French...), we use the same algorithm described in (Yin, 2001) to calculate the skew angle. After applying the horizontal run-length smoothing procedure and locating the black–white transitions on the text image, we determine the skew angle by the randomized Hough transform. Then the baselines are extracted using y-intercept histogram without skew correction. For experiments, 400 pages from different books and newspapers are considered. An example illustrating the baseline extraction algorithm for English skewed text image is shown in Fig. 11.

Experiments on Printed texts with different fonts, sizes and styles were also considered. As results, the baselines of the printed documents are all correctly estimated.

Note that each detected baseline described above corresponds to one text line, which may help us to successfully segment the text into individual text lines without skew correction.

All experiments above prove that the proposed algorithm is very efficient especially in the case where straight edges really exist in the image.

5. Conclusion

A novel fast algorithm for evaluating the skew correction and baseline detection of Arabic documents is proposed. The main novelty of our approach is that, as the lower baselines belong to the lower edge of words, we apply a randomized Hough transform to the relevant pixels of lower edge image to detect the skew angle. Then the baselines are extracted using y-intercept histogram with or without skew correction. To the best of our knowledge, such an algorithm has been applied neither for skew correction nor for baseline detection in Arabic script recognition field.

The main benefits and advantages of this algorithm are:

The entire document image is reduced into a relevant lower edge image using a simple labeling algorithm. Thus, our algorithm is based only on the analysis of both directional and y-intercept histograms of the lower edge pixels.

For uniformly skewed document we can just make a small part of document (approximately one or two lines) to detect the skew angle, which make the algorithm faster and more accurate.

For y-intercept detection, we use voting only with the significant clusters (peaks) in directional histogram (one y-intercept histogram for uniformly skewed documents and more than one for multi-skewed documents). Thus, we reduce considerably the computational cost compared to voting for (slope, y-intercept) simultaneously.

The directional histogram is not enough to locate straight lines especially in the multi-skewed images and short Arabic writing. Then we must verify it by analyzing the y-intercept histogram. Using y-intercept histogram, this work can contribute significantly for text line extraction from document images of printed and handwritten texts.

Besides, the advantage of the technique is that it can be applied to different applications for straight segment detection.

The main drawback of this system is the short Arabic handwritten writing (IFN/ENIT database) in some cases where the lower edge points do not emphasize straight lines or the subwords are not strictly aligned.

For future study, we plan to extend and adapt our algorithm to detect the curved baseline where the lower baseline should be adapted to each subword and not globally set (Boubaker et al., 2009).

References

Al-Rashaideh, H., 2006. Preprocessing phase for Arabic word handwritten recognition. Russ. Acad. Sci. 6 (1), 11–19.

Boubaker, H., Kherallah, M., Alimi, A.M., 2009. New algorithm of straight or curved baseline detection for short Arabic handwritten writing. In: Proc. 10th International Conference on Document Analysis and Recognition.

Burrow, P., 2004. Arabic Handwriting Recognition (M.Sc. thesis). University of Edinburgh, England.

El-Hajj, R., Likforman-Sulem, L., Mokbe, C., 2005. Arabic handwriting recognition using baseline dependant features and hidden Markov modeling. In: Proc. 8th International Conf. on Document Analysis and Recognition (ICDAR'05), vol. 20. IEEE, pp. 1520–5263, 5.

Farooq, F., Govindaraju, V., Perrone, M., 2005. Preprocessing methods for handwritten Arabic documents. In: Proc. 8th International Conference on Document Analysis and Recognition (ICDAR'05), vol. 1. IEEE, pp. 267–271.

O'Gorman, L., Kasturi, R. (Eds.), 1995. Document Image Analysis. IEEE Computer Society Press, New York.

Papandreou, A., Gatos, B., Perantonis, S.J., Gerardis, I., 2014. Efficient skew detection of printed document images based on novel combination of enhanced profiles. IJDAR 17, 433–454.

Parhami, B., Taraghi, M., 1981. Automatic recognition of printed Farsi texts. Pattern Recogn. 14 (1–6), 395–403.

Pechwitz, M., Maergner, V., 2002. Baseline estimation for Arabic handwritten words. In Frontiers in Handwriting Recognition, 479–484.

Pechwitz, M., Maergner, V., 2003. HMM-based approach for handwritten Arabic word recognition using the IFN/ENIT – database. In: ICDAR. IEEE Computer Society, pp. 890–894.

Pechwitz, M., Snoussi Maddouri, S., Maergner, V., Ellouze, N., Amiri, H., 2002. IFN/ENIT – database of handwritten Arabic words. In: Proceedings CIFED'02, pp. 129–136.

Rehman, A., Saba, T., 2011. Document skew estimation and correction: analysis of techniques, common problems and possible solutions. Appl. Artif. Intell. 25, 769–787.

Saba, T., Sulong, G., Rehman, A., 2011. Document image analysis: issues, comparison of methods and remaining problems. Artif. Intell. Rev. 35 (2), 101–118.

Shafii, M., Sid-Ahmed, M., 2015. Skew detection and correction based on an axes-parallel bounding box. IJDAR 18, 59–71.

Tang, Y.Y., Lee, S.W., Suen, C.Y., 1996. Automatic document processing: a survey. Pattern Recogn. 29, 1931–1952.

Xu, L., Oja, E., Kultanen, P., 1990. A new curve detection method: randomized Hough transform (RHT). Pattern Recogn. Lett. 11, 331–338.

Yin, P.Y., 2001. Skew detection and block classification for printed documents. Image Vis. Comput. 19, 567–576.

Ziaratban, M., Faez, K., 2008. A novel two-stage algorithm for baseline estimation and correction in Farsi and Arabic handwritten text line. In: Proc. 19th International Conference on Pattern Recognition (ICPR 2008).