International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.7, No.3 (2014), pp.421-432
http://dx.doi.org/10.14257/ijsip.2014.7.3.34

A Novel Skew Estimation Approach Based on Same Height Grouping

Wassim Al-Khawand[1*], Seifedine Kadry[2], Riccardo Bozzo[3] and Khaled Smaili[4]

[1]*School of Engineering Sciences and Technologies, University of Genoa–UNIGE, Genoa, Italy*
[2]*School of Engineering, American University of the Middle East, Kuwait*
[3]*DITEN-Dept. of Electrical, Electronic, Telecommunications Engineering and Naval Architecture University of Genoa, Genoa, Italy*
[4]*Faculty of Sciences, Lebanese University, Lebanon*
3854303@studenti.unige.it, skadry@gmail.com, Riccardo.Bozzo@unige.it, ksmeily@hotmail.com

Abstract

In this paper, we proposed a method to detect the rotation angle for the back side image of a container. Our technique consists of sorting the segmented characters according to the X axis, then detecting the segments belonging to the container number, after that, we divide the segments by groups having the same height, and finally the rotation angle of the image will be the average of the skewed angles related to each group. Our approach is robust, efficient and capable of handling any font and size of characters; regarding its complexity for an image having N lines and M characters, the worst CPU time usage and the worst memory usage is equal to O (NxM) while the network usage and disk usage for one image is O (1) which led –while using an old laptop- to a response time around 0.45 milliseconds to detect the rotation angle of an image when rotated from -45^o to +45^o with an precision error between ±0.2^o. The high accuracy and the fast response time for detecting the rotation angle of container images make our approach suitable for online OCR critical applications.

Keywords: *Skewed Image, Rotation Angle, Line Slope, Container Number*

1. Introduction

Due to the incurred problems related to skewed images taken from real-life scenarios, many papers tackled this issue in order to detect and adjust the rotated images.

Skew is the alignment of lines of text with respect to the horizontal axis and can be either clockwise (positive) or anti-clockwise (negative) [1].

The skew of an image can be global (*i.e.*, all the lines have the same orientation), multiple (i.e. different lines –or set of lines- have different orientations) or non-uniform (*i.e.*, lines having wavy shape) [2, 3].

There are many different methodologies for detecting and correcting a skew in a given document or page but mostly every technique has some limitations (*e.g.*, some techniques provide us speed but are only suitable for small text, other techniques provide us accurate results but are slow, and others are costlier in case they are good in speed and accuracy) [4]; it is worthy to note that, and as described in [5], the low accuracy in determining image rotation angle leads to an incorrect image processing results.

ISSN: 2005-4254 IJSIP

Due to the high importance of skew in document image processing, we will briefly list a summary of the available methods (knowing that some methods can be combined together in order to form other new methods): Projection Profile, Fast Fourier Transform, Hough Transform, Radon Transform, Wavelet Transform, Nearest Neighbor Clustering, Cross-Correlation, Piece-Wise Painting Algorithm, Transition Counts, Morphology, Histogram Analysis, Moments and other methods.

Worthy to note that, some degree of skew is inevitable when a document is scanned or a photo is taken manually or automatically; on the second hand, an image may appear slant due to the object position and orientation with respect to the camera and, it is almost impossible to take an undistorted image from a moving object; worthy to note that three possible kind of tilts may occur: horizontal, vertical and a combination of both [6].

Because it is very difficult -if not totally impossible- to prevent skew, it is therefore preferable to detect and correct the skew of the document image at the preprocessing stage in order to avoid the disturbance of skew to the further processing [7].

2. Context Description

Due to the high importance of containers in the shipping field, and in order to facilitate and accelerate trade (*e.g.*, Customs gates, terminal operators gates, quays, …) by automatically recognizing the container number, and because a skewed image may create problems to many Optical Character Recognition applications, our proposed method focuses on detecting the skew and the orientation for the back side image of a container. Figure 1 illustrates an image taken for the back side of a container.

Figure 1. Back Side Container Image

As shown in Figure 1, all the text lines in the container image are parallel in the horizontal direction and thus to detect the skew and orientation of such images, it will be enough to detect the skew and orientation of any line; to this end and in order to accelerate the whole process, we will confine our work to the interesting part of such images. Fig. 2 illustrates the interesting part of the back side container image -which always exist in the upper right area- where the first line contains the container number and the second line contains the container type.

Figure 2. Interesting Part of the Back Side Container Image

Although ISO 6346 is an international standard that covers the identification of the containers, but there is no standard for other text properties (*e.g.*, letters' size, font, spaces between letters and numbers …).

Because the longer the text line is, the better the skew and orientation detection of an image will be, our approach will use the container number to detect the skew and orientation of the whole image.

3. Proposed Methodology

Our proposed method consists of a Preparatory phase and an Execution phase.

3.1. Preparatory Phase

The preparatory phase consists of relying on any existing character segmentation approach in order to retrieve the coordinates of the useful segments related to the interesting part of the back side container image [8-14], because all we need to execute our approach is, for each segment "i", the upper-left coordinates (represented hereafter by X^i_{min}, Y^i_{min}) and the lower-right coordinates (represented hereafter by X^i_{max}, Y^i_{max}).

Worthwhile mentioning that, and in the most digital image processing, images are read from top to bottom and from left to right (*e.g.*, as shown in Figure 3, the X and Y axes are little bit different than the normal ones).

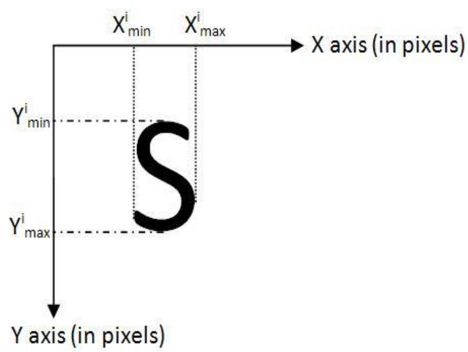

Figure 3. Coordinates for the Segment "i" which contains the Letter "S"

3.2. Execution Phase

The execution phase is the kernel of our approach and, as described below, it is very simple and consists of:

- Container number segments detection
- Rotation angle calculation

In order to acquire the best response time, our approach consists of working all the time in the memory instead of going to any other computer peripheral.

3.2.1. Container Nmber Segments Detection: This phase is composed of the following three 3 sub-phases:

1) First character detection

All container numbers are written from left to right because they are Latin characters; and thus, in order to detect the first character, we will proceed as the followings:

- First, we sort the segments retrieved in the previous section (*i.e.*, 3.1) according to their X^i_{min} value (where i goes from the first till the last segment), and therefore there will be no problem if the segments were scrambled (*i.e.*,retrieved in any order);
- Second, we calculate for each segment "i" its center coordinates represented hereafter by (X^i_{avg}, Y^i_{avg}), in the following way:

$$X^i_{avg} = (X^i_{min} + X^i_{max}) / 2$$
$$Y^i_{avg} = (Y^i_{min} + Y^i_{max}) / 2$$

- Third, the first segment in the array (*i.e.*, the segment having the coordinates X^1_{min}, Y^1_{min}, X^1_{max}, Y^1_{max}) will be the first character of the container number; to generalize our approach which can be implemented in several different environments, we will represent the first character by (X^{first}_{min}, Y^{first}_{min}, X^{first}_{max}, Y^{first}_{max}) because, and if in the following sub-phase we don't succeed to find a second character belonging to the same line, the second segment of the sorted array will be considered as being the first character in its line, and so forth till finding two segments belonging to the same line.

2) Second character detection

Having detected the first character of the container number line, we will enter the loop having the index i that goes from the segment that directly follows the "first" character till the last segment of the sorted array, looking for the first occurrence of a segment meeting the following criterias:

$$Y^i_{min} <= Y^{first}_{min} <= Y^i_{max} \text{ or}$$
$$Y^i_{min} <= Y^{first}_{max} <= Y^i_{max}$$

Having retrieved the first and the second segment (the second segment will be represented hereafter by X^{second}_{min}, Y^{second}_{min}, X^{second}_{max} and Y^{second}_{max}) of the container number, we will calculate the slope S related to these two characters as being:

$$S= (Y^{second}_{avg} - Y^{first}_{avg})/(X^{second}_{avg} - X^{first}_{avg})$$

As illustrated in the Figure 4 and in the Figure 5, a slope can be positive or negative (the slope is equal to zero if the line is parallel to the X axis)

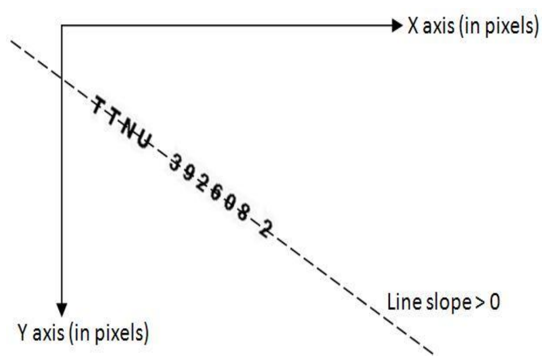

Figure 4. Line having a Positive Slope

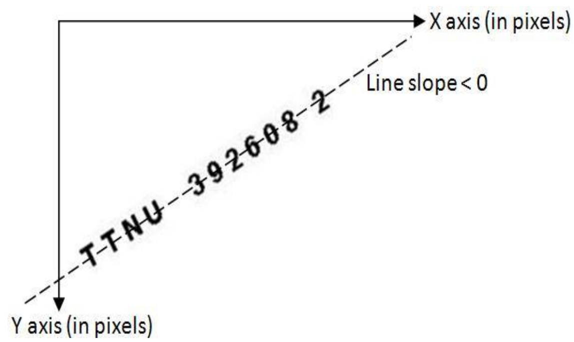

Figure 5. Line having a Negative Slope

3) Other characters detection

Still being in the same loop, we continue from the segment directly after the last retrieved one (the last retrieved segment - represented hereafter by X^{last}_{min}, Y^{last}_{min}, X^{last}_{max}, Y^{last}_{max}- is initially the "second" segment) looking for the next segment belonging to the container number (represented hereafter by X^{next}_{min}, Y^{next}_{min}, X^{next}_{max}, Y^{next}_{max}). The next segment will be the segment crossed by the line which passes by the center of the last retrieved segment and having the slope S; after retrieving the next character, we will adjust the previously calculated slope S as being the average of all the previously calculated slopes.

Finally, we start over looking for the remaining segments of the container number (*i.e.*, 3.2.1.3); to generalize our approach, we will continue looking for the remaining segments belonging to the same line, and if at the end of this process, the total number of segments is below a certain threshold, we will restart the Execution Phase (*i.e.*, Section 3.2) again while disregarding the segments previously selected.

N.B.: It is important to note that:

a. The equation of the line passing by the center of the last retrieved character (X^{last}_{avg},Y^{last}_{avg}) and having the slope "S" is equal to:

$$Y= S(X- X^{last}_{avg})+ Y^{last}_{avg}$$

b. The line L passing by the center of the last retrieved character (X^{last}_{avg},Y^{last}_{avg}) and having the slope S crosses the segment "i", if:

- In case the slope is positive, the line L passes by/below the upper-right coordinates of the segment "i" (*i.e.*, by/below X^{i}_{max}, Y^{i}_{min}) and by/above its lower-left coordinates (*i.e.*, by/above X^{i}_{min}, Y^{i}_{max}), which means:

$$Y^{i}_{min} <= S(X^{i}_{max} - X^{last}_{avg}) + Y^{last}_{avg} \text{ and}$$
$$Y^{i}_{max} >= S(X^{i}_{min} - X^{last}_{avg}) + Y^{last}_{avg}$$

Figure 6 illustrates this case.

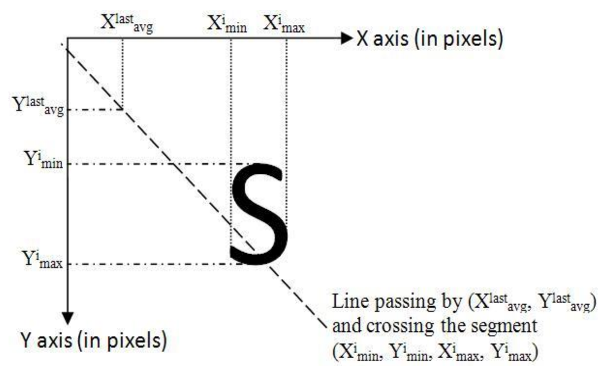

Figure 6. Example of a Line having a Positive Slope and crossing a Segment

-In case the slope is negative or equal to zero, the Line L passes by/below its upper-left coordinates (X^i_{min}, Y^i_{min}) and by/above its lower-right coordinates (X^i_{max}, Y^i_{max}), which means:

$Y^i_{min} <= S(X^i_{min} - X^{last}_{avg}) + Y^{last}_{avg}$ and
$Y^i_{max} >= S(X^i_{max} - X^{last}_{avg}) + Y^{last}_{avg}$

3.2.2. Rotation Angle Calculation: The height of the characters changes after being rotated, and even worse, the height of many characters will be different when rotated by the same angle but in an opposite orientation; Figure 7 shows a letter F having initially a height of 12 pixels when the image is not rotated (leftmost F), a height of 14 pixels when the image is rotated by -30^o (the middle F) and a height of 11 pixels when the image is rotated by $+30^o$ (rightmost F).

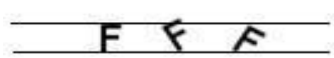

Figure 7. Different Heights for the Same Character after being Rotated

To achieve the best rotation angle detection, we will proceed as the following:

1) We categorize the container number segments by groups having the same height; it is worthy to note that groups having just one segment are discarded because no slope can be calculated and there is a high probability to be subject to distortion;

2) We calculate the slope related to each group as being the slope between the first and the last segment (according to the X axis) of the group, then we calculate their average;

3) The arctangent of the previously calculated average will be the rotation angle of the image.

4. Flowchart

Figure 8 illustrates the whole flowchart of our proposed method.

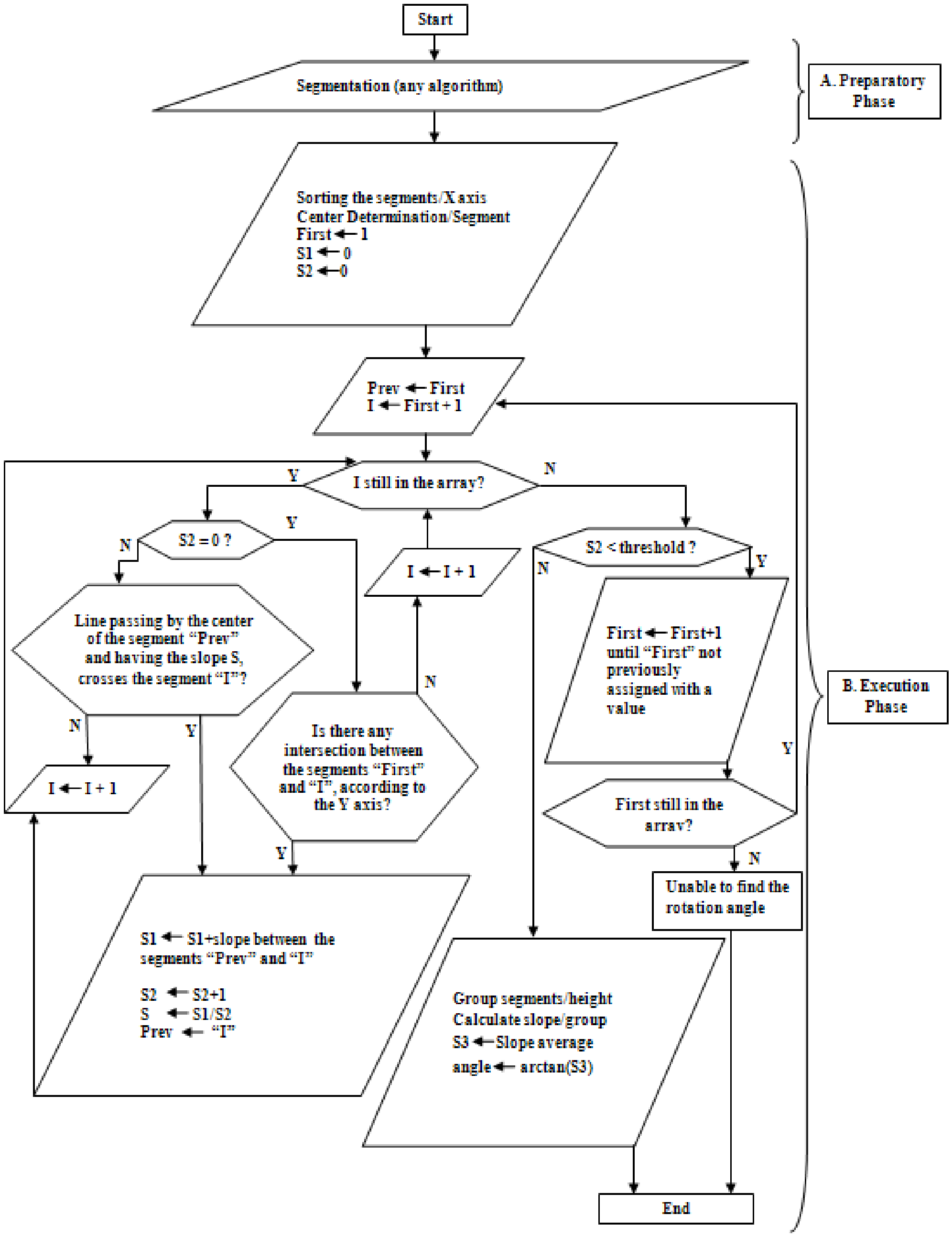

Figure 8. General Flowchart of our Proposed Method

5. Experiments

Our proposed method was initially designed to detect the rotation angle for skewed back side container images and it succeeded to return very good results regarding its performance and accuracy.

We tested our approach on 100 containers randomly selected among 100 different companies; The result of our experiments returned a response time around 0.45 milliseconds while using a laptop having a 2.00 GHz processor and 2 GB of RAM, which makes our proposed method very convenient and highly desirable for real-time applications; on the other hand, our approach was very accurate because it returned an error between $\pm 0.1^o$ for an image rotated between $\pm 15^o$ (*i.e.*, an average accuracy equals to 99.34%), and an error between $\pm 0.2^o$ for an image rotated between $\pm 45^o$ (*i.e.*, an average accuracy equals to 99.56%).

It is worthy to note that the existence of spaces before or in the middle of the container number doesn't have any negative effect on the result.

We will illustrate in the Figures 9, 10 and 11 some samples from our experimental results. To make the figures more readable, each figure will contain two parts where the upper one represents the rotated image and contains two lines (the first one represents the Container Number and the second one represents the Container Type) and the lower part of the image represents the result (*i.e.*, elapsed time and detected skew angle).

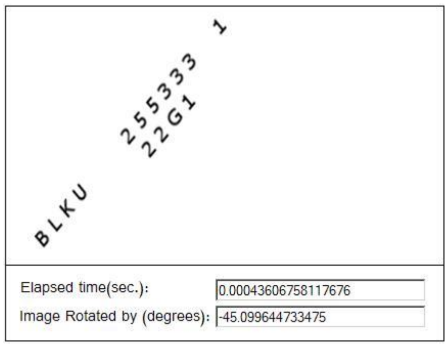

Figure 9.Image rotated by -45^o

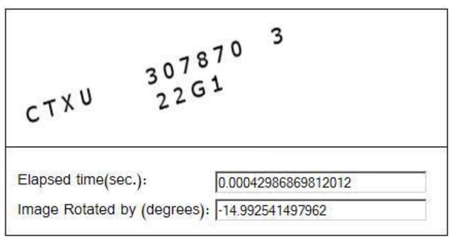

Figure 10. Image rotated by -15^o

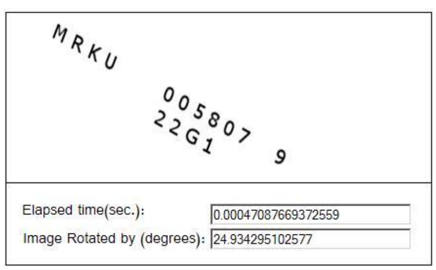

Figure 11. Image rotated by +25^o

Finally, we would like to highlight that our proposed method can tolerate and deal with distorted segments, subject to bad image quality or due to image filtering, because we designed our approach in a way that all the segments -part of the container number- and having the same height, are grouped together then, the average of all the slopes is taken into consideration, while disregarding all the groups having one segment.

6. Application Field

Although this paper was intended to detect the rotation angle of images taken for the backside of a container, but it can also be implemented to detect the rotation of the front –Figure 12-, back –Figure 1-, left, right or the top side of any container image; worthy to note that it can also be used for other kind of images (e.g. Container owner name –Fig. 13- and vehicle plate number –Fig. 14-).

Figure 12. Container front Side

Figure 13. Image Showing the Owner of the Container

Figure 14. Vehicule Plate

7. Related Works

In this study, we will present some excerpts of similar works as mentioned in other papers.

a) In [15], the authors presented a comparative study between different methodologies where they illustrated the accuracy and the average time for images skewed from 1^o to 25^o.

Methodology	Accuracy	Average time (in seconds)
Projection profile technique	99%	2.83
Peaks and valleys analysis	99.09%	1.96
Connected component analysis	99.5%	2.55
Radon transform	100%	0.1404
Fast Fourier transform	100%	29.36
Horizontal projection profile+Hough transform	99%	0.8644

Wavelet transform+Hough transform	99.6%	0.56
Hough transform	99.6%	0.59
Gabor folter + Radon transform	100%	1.5288

b) In [7], the authors presented the mean error and the standard deviation for three methods after being applied to images skewed between -3.8^o and 4.2^o.

Skew angle (Degree)	Hough transform. method	Cross correlation method	Cao method (based on straight-line fitting)
-3.8	-4.0	-3.82	-3.85
-2.8	-2.9	-2.89	-2.78
-1.8	-1.9	-1.83	-1.79
-0.8	-1.1	-0.74	-0.86
0.1	0.0	0.10	0.02
0.2	0.1	0.10	0.23
1.2	1.0	1.07	1.22
2.2	2.1	2.27	2.19
3.2	3.0	3.09	3.20
4.2	4.1	4.13	4.21
Mean error	0.15	0.068	0.029
Standard deviation	0.06707	0.03944	0.02601

Also in [7], it is mentioned that the average skew detection time of their method is 1.26 seconds, while Hough transformation method is 22.3 seconds and the cross correlation method is 4.52 seconds.

c) In [16] and for images skewed from 1^o to 20^o, the authors presented the error of their proposed approach according to the actual skew angle.

Actual Skew Angle	Error Calculation
1	-0.03
5	-0.17
10	-0.45
20	-0.05

d) In [17], the authors observed that 515.16s are required to estimate the skew angle when they employed Hough Transform (HT) for whole image of size 512x512, a time of 35.41s when they applied HT-based method of Le *et al.*, a time of 30.07s when they employed the HT-based method of Pal&Chaudharys and a time of 10.35s when they employed their method that does not involve HT; the authors also presented the skew angle estimation for their method when applied to images skewed from 3^o to 30^o.

True angle	Skew angle estimation (may vary according to the used dpi)
3	3.74
5	5.59
10	10.58
20	20.59
30	30.56

e) In [18], Chou and colleagues limited searching for the skew range between ±15^o.
f) In [19], all the documents of the datasets were randomly rotated in 10 different angles, ranging from -15^o to +15^o.
g) In [20], a simple method based on skew detection on the polygon center of gravity is proposed with an accuracy up to 87%.
h) In [21], it is mentioned that the cross-correlation method is suited for small skew angles up 10^o, the simplest method based on the polygon gravity centers for skew estimation has an accuracy lower than 87%, the absolute error for the algorithm without dilatation for the single-line test is below 0.3^o for a 300dpi image, below 0.2^o for a 50dpi image and below 0.6^o for a 25dpi image.
i) In [22], it is mentioned that the average processing time is 77.998 milliseconds when CAM-based architecture is implemented and 128.367 milliseconds when RAM-based architecture is implemented.
j) In [23], it is mentioned that the average deviation error is estimated within two degrees from the true angle.
k) In [24], it is mentioned that the Rotation from -5^o to 5^o presumed an error deviation of 0.2^o
l) l) In [25], it is mentioned that the average deviation error of the skew corrected text lines is from -3^o to +3^o
m) In [26], it is mentioned that the skew angle error varies from 0^o to 0.6^o

8. Future Work and Conclusion

Our approach consists of detecting the rotated image angle in fast and accurate manner and thus, it is suitable for online OCR applications. Our approach was intended for detecting the rotation angle of an image taken for the back side of a container, so it was designed to read any font and letter size but having the same height (for a non rotated container image) and thus, further enhancements should be applied to deal with capital and small letters belonging to the same line; the second issue that needs further research is to calculate the skew of a document image having multiple orientations for different lines (or set of lines).

References

[1] S. Kumar and S. Gopinath, “Two Phase Data Reduction for Hough Transformation to Detect Skew”, International Journal on Computer Science and Engineering (IJCSE), ISSN, vol. 3, no. 10, **(2011)**, pp. 0975-3397.
[2] S. B. Rezaei, A. Sarrafzadeh and J. Shanbehzadeh, “Skew Detection of Scanned Document Images”, International Multi Conference of Engineers and Computer Scientists, IMECS, vol. 1, **(2013)**.
[3] K. IUliu, E. Stefan and S. Christoph, “Fast Seamless Skew and Orientation Detection in Document Images”, Fraunhofer Institute for Intelligent Analysis and Information Systems (IAIS) SchlossBirlinghoven, 53754 Sankt Augustin, Germany, **(2010)**.

[4] R. Singh and R. Kaur, "Skew Detection in Image Processing", Int. J. Computer Technology & Applications (IJCTA), ISSN 2229-6093, vol. 4, no. 3, **(2013)**, pp. 478-485.
[5] W. Al-Khawand, S. Kadry, R. Bozzo and K. Smaili, "Reading Skewed Images Without Image Rotation", British Journal of Mathematics & Computer Science, ISSN, vol. 4, issue 7, **(2014)**, pp. 2231-0851.
[6] C. N. Paunwala, S. Patnaik and M. Chaudhary, "An efficient Skew Detection of License Plate Images Based on Wavelet Transform and Principal Component Analysis", International Conference on Signal and Image Processing (ICSIP), **(2010)**.
[7] Y. Cao, S. Wang and H. Li, "Skew Detection and Correction in Document Images Based on Straight-Line Fitting", Pattern Recognition Letters, vol. 24, issue 12, **(2003)**, pp. 1871-1879.
[8] H. Mobahi, S. Rao, A. Yang, S. Sastry and Y. Ma "Segmentation of Natural Images by Texture and Boundary Compression", International Journal of Computer Vision (IJCV), vol. 95, no. 1, **(2011)**, pp. 86-98.
[9] K. J. Batenburg and J. Sijbers, "Adaptive thresholding of tomograms by projection distance minimization", Pattern Recognition, vol. 42, no. 10, **(2009)**, pp. 2297-2305.
[10] P. Arbelaez, M. Maire, C. Fowlkes and J. Malik, "Contour Detection and Hierarchical Image Segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, **(2011)**.
[11] Z. Ren and G. Shakhnarovich "Image Segmentation by Cascaded Region Agglomeration", CVPR, **(2013)**.
[12] R. Achanta, "SLIC superpixels compared to state-of-the-art superpixel methods", IEEE TPAMI, **(2012)**.
[13] J. Carreira, R. Caseiro, J. Batista and C. Sminchisescu, "Semantic segmentation with second-order pooling", In ECCV, **(2012)**.
[14] V. Karthikeyan, R. Sindhu, K. Anusha and D. S. Vijith, "Vehicule License Plate Character Segmentation – A STUDY", International Journal of Computer and Electronics Research, vol. 2, **(2013)**.
[15] P. K. Aithal, "A Fast and Novel Skew Estimation Approach using Radon Transform", International Journal of Computer Information Systems and Industrial Management Applications", ISSN 2150-7988, vol. 5, **(2013)**, pp. 337-344.
[16] Q. Xiaorui, "Fast Skew Angle Detection Algorithm for Scanned Document Images", Third Pacific-Asia conference on Circuits, Communications and System (PACCS), **(2011)**.
[17] P. Shivakumara, "A Novel Technique for Estimation of Skew in Binary Text Document Images Based on Linear Regression Analysis", Official Indian Journal for Academy, Proceedings in Engineering Sciences, vol. 30, part 1, **(2005)**, pp. 69-85.
[18] C.-H. Chou, S.-Y. Chu and F. Chang, "Estimation of Skew Angles for scanned Documents Based on Piecewise Covering by parallelograms", Pattern Recognition, vol. 40, **(2007)**, pp. 443-455.
[19] A. Papandreou, "Document Image Skew Estimation Contest", International Conference on Document Analysis and Recognition (ICDAR), **(2013)**.
[20] D. Brodić, "The Evaluation of the Initial Skew Rate for Printed Text", Journal of Electrical Engineering, vol. 62, no. 3, **(2011)**, pp. 134-140.
[21] D. Brodić and D. R. Milivojević, "An Algorithm for the Estimation of the Initial Text Skew", Information Technology and Control, ISSN 1392-124X, vol. 41, no. 3, **(2012)**.
[22] Y.-P. Chen, T.-D. Yeh, F.-C. Ni and C.-H. Chang, "A CAM-Based Pattern Accumulated Vector Method for Real-Time Character Recognition of License Plates", IJCSNS International Journal of Computer Science and Network Security, vol. 9, no. 1, **(2009)**.
[23] H. Hase, K. Tanabe, T. H. H. Tran and S. Tokai, "Multi-Font Rotated Character Recognition Using Periodicity", The Eighth IAPR International Workshop on Document Analysis Systems, DAS'08, **(2008)**.
[24] A. Papandreou and B. Gatos, "A Novel Skew Detection Technique Based on Vertical Projections", International Conference on Document Analysis and Recognition, **(2011)**.
[25] A. F. Mollah, S. Basu, N. Das and R. Sarkar, "A Fast Skew Correction Technique for Camera Captured Business Card Images ", IEEE Annual India Conference (INDICON), **(2009)**.
[26] Y. Zhongda, D. Junyu, W. Zhiqiang and S. Jianxiang "A Fast Image Rotation Algorithm for Optical Character Recognition of Chinese Documents ", International Conference on Communications, Circuits and Systems Proceedings, **(2006)**.

