

Reading Skewed Images without Image Rotation

Wassim Al-Khawand^{1*}, Seifedine Kadry², Riccardo Bozzo³,
Khaled Smaili⁴

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

ABSTRACT

Aims: To extract characters from skewed images without image rotation.

Study design: This study is designed to be implemented at the gates of Customs, Port Authorities, Terminal Operators and it can also be implemented for vehicles traffic management.

Place and Duration of Study: Customs premises, between September and November 2013.

Methodology: The proposed method consists of sorting the segmented characters according to the X axis, then assigning a Program Line Number to each character based on the skew angle and finally sorting the Program Line Numbers according to their intersection with the Y axis.

Results: Our approach is capable of handling any font and size of characters and it is robust and efficient; regarding its complexity for an image having N lines and M characters, the worst CPU time usage and worst memory usage equal to $O(N \times M)$ while the network usage and disk usage for one image is $O(1)$ which led to a 0.11 millisecond response time to extract all container number digits.

Conclusion: Non human intervention at the entry/exit of containers gates which speeds up the clearance process and eliminates the human errors.

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

1. INTRODUCTION

Due to the incurred problems related to rotated images taken from real-life scenarios, many papers tackled this issue in order to detect and rotate the skewed images. The existing algorithms consist of rotating the image after calculating its inclination angle [1-3] before proceeding into the next step of the Optical Character Recognition (OCR) process.

On the one hand, image rotation can never add details to the image and thus, the result of an adjustment of a skewed image can easily deteriorate the quality of the image (colors less intense, haloes creation around the characters) as illustrated in Fig.1 and Fig.2, and

* E-mail address: 3854303@studenti.unige.it, telephone: +961 3 835410

31 may also result in losing some of the surrounding pixels and/or characters as shown in Fig.3
32 and Fig.4.

33

34 SUDU 184795 6

Figure 1. Container number original image.

35

SUDU 184795 6

Figure 2. Quality deterioration plus haloes after two 25 ° rotations.

SUDU 184795 6

Figure 3. Container number skewed image.

JDU 184795

Figure 4. Lost of pixels/characters after 25° rotation.

36 On the other hand, image rotation is time consuming due to the complexity of its
37 computation which is considered as a big problem in real-time systems because it affects the
38 overall performance of the character recognition process and therefore it will be better if it
39 can be avoided.

40

41 2. PROPOSED METHOD

42

43 Based on the above described drawbacks of image rotation, our proposed method
44 consists of extracting the characters of an image subject to a later Optical Character
45 Recognition (OCR) without rotating the image. Our approach is very simple and consists of
46 the following two phases: preparatory and execution phases.

47

48 2.1 Preparatory Phase

49

50 The preparatory phase consists of relying on any existing characters' segmentation and
51 determination of the image rotation angle.

52

53 2.1.1 Characters' Segmentation

* E-mail address: 3854303@studenti.unige.it, telephone: +961 3 835410

54 Any segmentation approach can be used in order to detect the useful segments from a
 55 rotated image [4-11], and all we need is the upper-left coordinates represented by (X_{min}, Y_{min})
 56 and the lower-right coordinates represented by (X_{max}, Y_{max}) of each segment.
 57 It is worthy to note that in digital image processing, images are read from top to bottom and
 58 from left to right, so as shown in Fig. 5, the X and Y axes are a little bit different than the
 59 normal used ones.

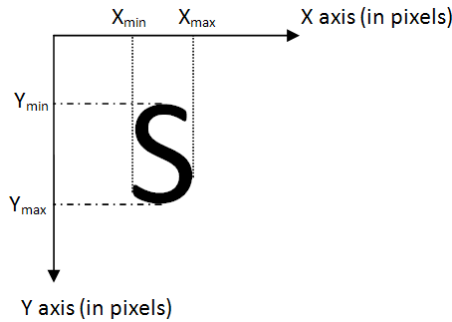


Figure 5. Letter S coordinates.

61

62 **2.1.2 Image Rotation Angle Calculation**

63

64 Any skew detection approach can be used in order to determine the rotation angle of the
 65 image (represented hereafter by θ), worthwhile to mention that such algorithms are classified
 66 into the following four main categories [12]: Hough transform [13,14], projection profile
 67 [15,16], nearest neighbor clustering [17,18] and interline cross correlation [19-21].

68 All we need from this sub-phase is to know the image rotation angle in order to start the
 69 execution of our proposed approach.

70

71 **2.2 Execution Phase**

72

73 The execution phase is the kernel of our approach and is divided into Segments Line
 74 Determination and Program Lines Ordering.

75

76 **2.2.1 Segments Line Determination**

77

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

80 First, we start by determining the coordinates (X, Y) of the center of each segmented
 81 character retrieved in the preparatory phase, in the following way:

$$82 \quad X = (X_{min} + X_{max}) / 2$$

$$83 \quad Y = (Y_{min} + Y_{max}) / 2$$

84 where (X_{min}, Y_{min}) represents the upper-left corner of the segmented character and
 85 (X_{max}, Y_{max}) represents the lower-right corner of the segmented character.

Second and because Latin characters are written from left to right, we will sort the segmented characters according to their $X_{\min}$ value and thus there will be no problem if the segmented characters which were retrieved in the preparatory phase are scrambled (i.e. retrieved in any order).

Third, we initialize the Program Line Number (PLN) variable by zero then we enter the first loop where the variable "I" will hold the first till the last segment of the previously sorted array, where we will:

a. Check if the segment I having –for example- the center (X_1, Y_1) , was previously assigned with a Program Line Number; if the answer is positive we will pass to the next step of the first loop (i.e. we put in I the next segment of the sorted array and we go back to step a) otherwise, we will increment the PLN by one and we assign it to I and then we will enter a second loop where the variable "J" will hold the segment which exists after I till the last segment of the sorted array, where we will:

b. Check whether the segment J was previously assigned with a Program Line Number; if the answer is positive we will pass to the next step of the second loop (i.e. we put in J the next segment of the sorted array and we go back to step b) otherwise, we check whether the Imaginary Line passing by (X_1, Y_1) crosses the segment J; if the answer is positive, PLN is assigned to J otherwise nothing will be done. Finally, we will pass to the next step of the second loop (i.e. we put in J the next segment of the sorted array then we go back to step b).

It is important to note that:

1. If an image is rotated by θ , all the lines inside the image will be rotated by the same value.
2. The slope of the line passing by the two points (X_1, Y_1) and (X_2, Y_2) is equal to $(Y_2 - Y_1)/(X_2 - X_1)$ which is equal to the tangent of θ .

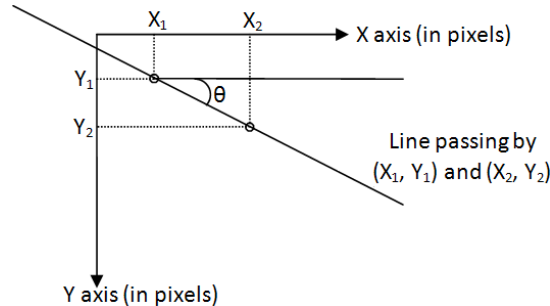


Figure 6. Relation between angle rotation and line slope.

Fig.7 shows a line having a positive slope while Fig.8 shows a line having a negative slope; it is worthy to note that a line parallel to the X axis has a slope equal to zero.

116

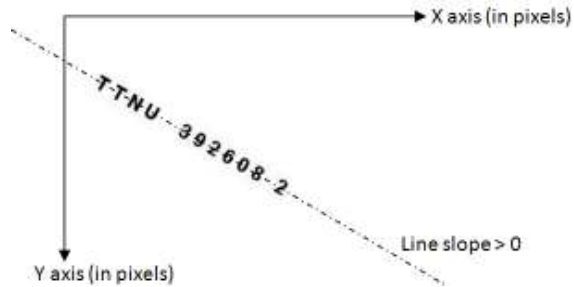


Figure 7. Line having a positive slope.

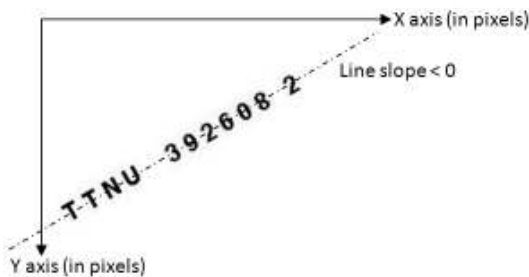
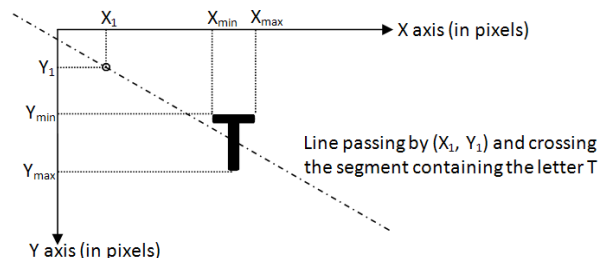


Figure 8. Line having a negative slope.

- 117 3. The equation of the imaginary line passing by (X_1, Y_1) and having the slope “S” is
 118 equal to:
 119 $Y = S(X - X_1) + Y_1$
- 120 4. The Imaginary Line passing by (X_1, Y_1) and having the slope S crosses a segment if:
 121 - In case the slope is positive, the Imaginary Line passes by/below its upper-right
 122 coordinates $(X_{\max}, Y_{\min})$ and by/above its lower-left coordinates $(X_{\min}, Y_{\max})$, which
 123 means:
 124 $Y_{\min} \leq S(X_{\max} - X_1) + Y_1$ and
 125 $Y_{\max} \geq S(X_{\min} - X_1) + Y_1$
- 126 - In case the slope is negative or equal to zero, the Imaginary Line passes by/below
 127 its upper-left coordinates $(X_{\min}, Y_{\min})$ and by/above its lower-right coordinates $(X_{\max},$
 128 $Y_{\max})$, which means:
 129 $Y_{\min} \leq S(X_{\min} - X_1) + Y_1$ and
 130 $Y_{\max} \geq S(X_{\max} - X_1) + Y_1$



131

Figure 9. Example of a line crossing a segment.

- 132 5. All the segments which are crossed by the Imaginary Line passing by (X_1, Y_1) and
 133 having the slope S will have the same PLN because they are in the same line of the
 134 rotated image.
 135 6. At the end of the B1 phase, all the segments will be assigned with a PLN but the
 136 PLNs will not reflect the real ordering of the image lines.
 137

138 2.2.2 Program Lines Ordering

139

140 At this stage, all the segments are classified according to their PLNs but which may not
 141 be similar to the real order of the image lines because our approach can deal with scrambled
 142 segments. To solve this issue, the PLNs will be sorted according to their Imaginary Line
 143 intersection with the Y axis. Below, the three figures (Fig. 10, 11 and 12) will illustrate the
 144 problem and its solution.

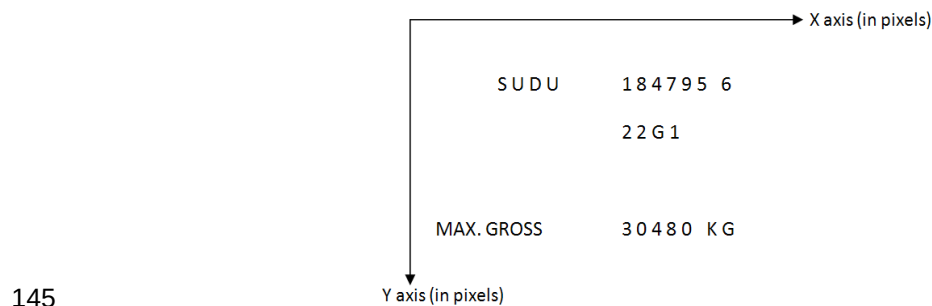


Figure 10. Original binarized upper-right back container image.

Fig.11 shows the PLNs state before the Program Line Ordering phase (i.e. this problem occurred due to sorting the segments according to their X_{min} value because our approach can deal with scrambled segmented segments)

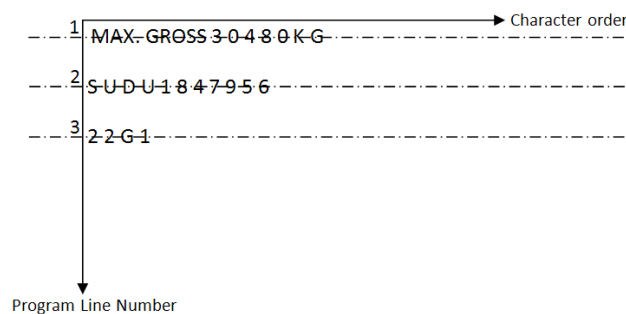


Figure 11. PLNs before Program Lines Ordering phase.

Fig.12 shows the PLNs ordered according to their Imaginary Line intersection with the Y axis.

146

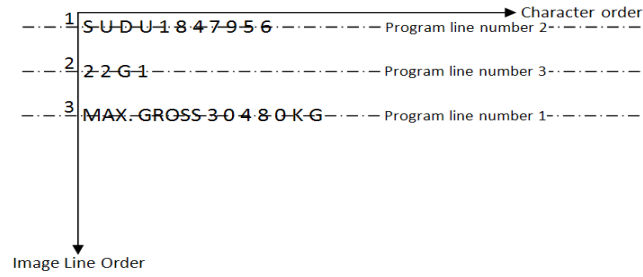
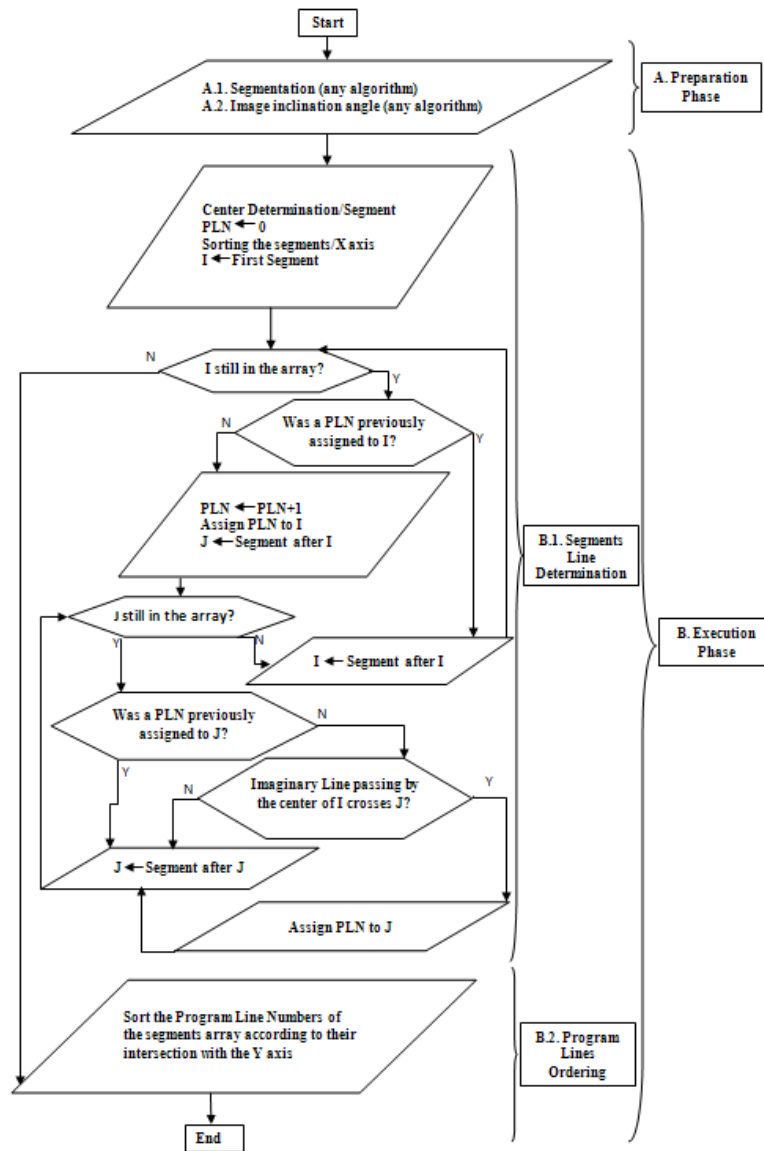


Figure 12. PLNs after the Image Line Ordering phase.

2.3 Flowchart



148

Figure 13. Flowchart of the whole proposed method.

* E-mail address: 3854303@studenti.unige.it, telephone: +961 3 835410

149 3. EXPERIMENTS

150

151 Our proposed method was initially designed for skewed images taken from the container
152 back side and it succeeded to return very good results regarding its performance and ability
153 to deal with rotated images.

154 The result of our experiments returned a response time around 0.11 millisecond for
155 executing our approach on an image containing the container number and the container type
156 while using a laptop having a 2.00 GHz processor and 2 GB of RAM, which makes our
157 proposed method very convenient and highly desirable for real-time applications.

158 Our proposed approach is very robust because regarding its efficiency and complexity,
159 supposing that we have N lines and M characters in a container image, the worst CPU time
160 usage is $O(N \times M)$ and the worst memory usage is $O(N \times M)$ while the network usage and the
161 disk usage for one image is $O(1)$.

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

164 Below, we will illustrate some samples from our experimental results. To make the figures
165 more readable, each figure will contain two parts where the part above the middle thick line
166 represents the original image and contains two lines (the first one represents the Container
167 Number and the second one represents the Container Type) and the part below the middle
168 thick line represents the extracted segments without applying any image rotation.

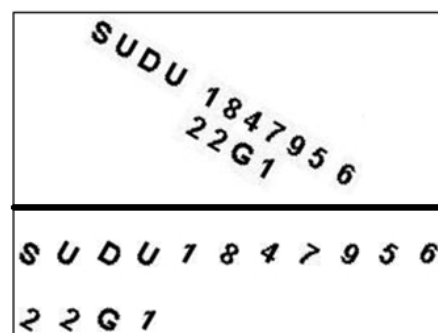
169



170

Figure 14. Image rotation = 0°

171

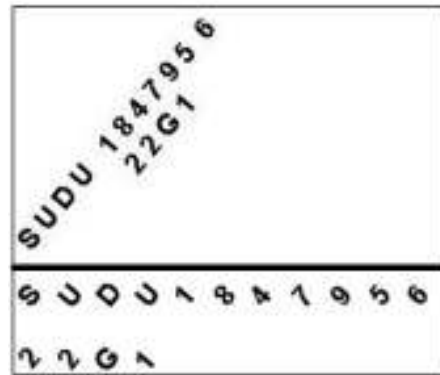


172

Figure 15. Image rotation = -30°

* E-mail address: 3854303@studenti.unige.it, telephone: +961 3 835410

173



174

Figure 16. Image rotation = 50°

175 Worthwhile mentioning that the inclination of the extracted segments doesn't represent
176 any issue because any OCR application can easily recognize them after being trained.

177 Finally, we would like to highlight that our proposed method can tolerate and deal with
178 distorted segments, due to bad image quality or subject to image filtering, because we
179 designed our approach in a way that a segment is affected with a PLN if the Imaginary Line
180 crosses it (i.e. disregarding if it is higher or lower from the center of the segment).

181

182 **4. APPLICATION FIELD**

183

184 Although this paper was intended to read the container number of a skewed image (from
185 the front –Fig. 17-, from the back –Fig. 18-, left, right or top side of the container), it can also
186 be used to read the container owner name –Fig. 19- and the vehicle plate number –Fig. 20-.

187



188

Figure 17. Container front side.

* E-mail address: 3854303@studenti.unige.it, telephone: +961 3 835410



Figure 18. Container back side.



Figure 19. Image showing the owner of the container.



Figure 20. Vehicle plate.

189

190

191

192 5. COMPARATIVE STUDY

193 In this study, we will present the benefit from being able to extract segments from a
 194 skewed image without rotating it; to this end, below we will list the execution time for image
 195 de-skewing as mentioned in their corresponding paper:

- 196 a. The best computation time for skew correction is 0.017 second for a resolution equal
 197 to 0.45 Mega pixel for a 800x600 image [22].
- 198 b. The time cost varies from 200 to 1000 millisecond [23].
- 199 c. Around 14.7 millisecond is needed to rotate 256x256 image [24].
- 200 d. The average processing time is 77.998 millisecond if implemented on a CAM-based
 201 architecture and 128.367 millisecond when implemented on RAM-based
 202 Architecture [25].
- 203 e. The average processing time goes from 17 millisecond for a 0.45 mega pixel image
 204 till 110 millisecond for 3.0 mega pixel image [26].

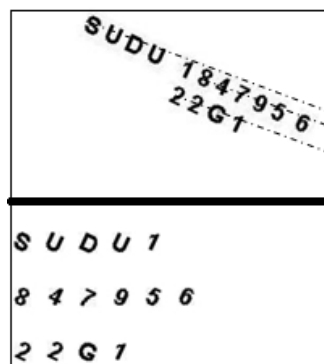
* E-mail address: 3854303@studenti.unige.it, telephone: +961 3 835410

205 6. FUTURE WORKS

206 Because our approach was intended for extracting the container number from a skewed
207 container image, so it was designed to read any font and letter size -but having the same
208 height- and thus further enhancements should be done to deal with capital and small letters
209 belonging to the same line; the second issue that needs further research is to calculate, from
210 the segmented characters, the image rotation angle without relying on any other existing
211 algorithm because they are not 100% accurate while detecting the rotation angle of an
212 image which may lead in some cases to incorrect results.

213 Fig.21 illustrates an example of an inaccurate angle rotation value which led to an
214 incorrect result because the container number was split into two lines. The dash dot lines in
215 the Fig.21 represent the Imaginary Lines where the upper part (i.e. the part above the middle
216 thick line) of the figure represents the original image and the lower part represents the result.

217



218

Figure 21. Incorrect angle rotation.

219 Finally, we will list an extract of the deviation error as mentioned in their corresponding
220 paper:

- 221 a. The average deviation error is estimated within two degrees from the true angle [27]
- 222 b. The Rotation from -5° to 5° presumed an error deviation of 0.2° [12]
- 223 c. The average deviation error of the skew corrected text lines is from -3° to 3° [22]
- 224 d. The skew angle error varies from 0° to 0.6° [23]
- 225 e. The skew angle error is between -3° and 3° degrees [26]

226

227

228 AUTHORS' CONTRIBUTIONS

229

230 The author Wassim Al-Khawand analyzed, developed and wrote this manuscript, the co-
231 author Seifeddine Kadry closely directed, greatly supported and deeply reviewed this study,
232 while the co-authors Riccardo Bozzo and Khaled Smaili directed, supported and reviewed
233 this work. All authors read and approved the final manuscript.

234 REFERENCES

235

* E-mail address: 3854303@studenti.unige.it, telephone: +961 3 835410

- 236 [1] T.D. Duan, T.L.H. Du, T.V. Phuoc and N.V. Hoang, "Building an Automatic Vehicle
237 License-Plate Recognition System", International Conference in Computer Science,
238 RIVF'05, 2005.
- 239 [2] J.C.M. Lee, "Automatic character recognition for moving and stationary vehicles and
240 containers in real-life images", International Joint Conference on Neural Networks,
241 IJCNN '99, Vol. 4, 1999.
- 242 [3] A.F. Mollah, N. Majumder, S. Basu and M. Nasipuri "Design of an Optical Character Recognition System for Camera-based Handheld
243 Devices", IJCSI International Journal of Computer Science Issues, Vol. 8, No 1, 2011.
- 244 [4] H. Mobahi, S. Rao, A. Yang, S. Sastry and Y. Ma "Segmentation of Natural Images by
245 Texture and Boundary Compression", International Journal of Computer Vision (IJCV),
246 95 (1), pg. 86-98, 2011.
- 247 [5] K J. Batenburg and J. Sijbers, "Adaptive thresholding of tomograms by projection
248 distance minimization", Pattern Recognition, vol. 42, no. 10, pp. 2297-2305, 2009.
- 249 [6] P. Arbelaez, M. Maire, C. Fowlkes and J. Malik "Contour Detection and Hierarchical
250 Image Segmentation", IEEE Transactions on Pattern Analysis and Machine Intelligence,
251 Vol. 33, 2011.
- 252 [7] Z. Ren, G. Shakhnarovich "Image Segmentation by Cascaded Region Agglomeration",
253 CVPR, 2013.
- 254 [8] R. Achanta et Al. "SLIC superpixels compared to state-of-the-art superpixel methods",
255 IEEE TPAMI, 2012.
- 256 [9] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik "Contour detection and hierarchical
257 image segmentation", IEEE TPAMI, 33(5), 2011.
- 258 [10] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu "Semantic segmentation with
259 second-order pooling", In ECCV, 2012.
- 260 [11] V. Karthikeyan, R. Sindhu, K. Anusha and D.S. Vijith, "Vehicule License Plate Character
261 Segmentation – A STUDY", International Journal of Computer and Electronics
262 Research, Vol. 2, 2013.
- 263 [12] A. Papandreou and B. Gatos, "A Novel Skew Detection Technique Based on Vertical
264 Projections", International Conference on Document Analysis and Recognition, 2011.
- 265 [13] X. Fan, G. Fan and D. Liang "Joint Segmentation and Recognition of License Plate
266 Characters", IEEE International Conference on Image Processing, ICIP, 2007.
- 267 [14] C. Singh, N. Bhatia and A. Kaur, "Hough transform based fast skew detection and
268 accurate skew correction methods", Pattern Recognition, vol.41, pp. 3528-3546, 2008.
- 269 [15] J. Jagannathan, A. Sherajdheen, R.M.V. Deepak and N. Krishnan "License plate
270 Character Segmentation using horizontal and vertical projection with dynamic
271 thresholding", International Conference on Emerging Trends in Computing,
272 Communication and anotechnology (ICE-CCN), 2013.
- 273 [16] N. Liolios, N. Fakotakis and G. Kokkinakis "Improved document skew detection based
274 on text line connected component clustering". International Conference on Image
275 Processing, volume 1, pp. 1098-1101. Proceedings. 2001.
- 276 [17] Sarfraz M., Mahmoud S.A. and Rasheed Z. "On Skew Estimation and Correction of
277 Text", Computer Graphics, Imaging and Visualisation, CGIV '07, 2007.
- 278 [18] Y. Lu and C.L. Tan "A nearest-neighbor chain based approach to skew estimation in
279 document images". Pattern Recognition Letters, Vol.24, pp. 2315–2323, 2003.
- 280 [19] T. Akiyama and N. Hagita; "Automatic entry system for printed documents," Publisher
281 Elsevier Science, Pattern Recognition vol. 23 , issue 11, pp.1141 – 1154, 1990.
- 282 [20] M. Chen and X. Ding "A robust skew detection algorithm for grayscale document image",
283 Proceedings of the Fifth International Conference on Document Analysis and
284 Recognition, ICDAR '99, 1999.
- 285 [21] A. Chaudhuri, S. Chaudhuri "Robust detection of skew in document images", IEEE
286 Transactions on Image Processing, vol.6, pp.344–349, 1997.
- 287

- 288 [22] A.F. Mollah, S. Basu, N. Das and R. Sarkar, "A Fast Skew Correction Technique for
289 Camera Captured Business Card Images ", IEEE Annual India Conference (INDICON),
290 2009.
- 291 [23] Y. Zhongda, D. Junyu, W. Zhiqiang and S. Jianxiang "A Fast Image Rotation Algorithm
292 for Optical Character Recognition of Chinese Documents ", International Conference on
293 Communications, Circuits and Systems Proceedings, 2006
- 294 [24] Xiao-Gang JIANG, Jian-Yang ZHOU, Jiang-Hong SHI and Hui-Huang CHEN "FPGA
295 Implementation of Image Rotation Using Modified Compensated CORDIC", 6th
296 International Conference on ASIC Proceedings, ASICON, 2005.
- 297 [25] Y.-P. Chen, T.-D. Yeh, F.-C. Ni and C.-H. Chang, "A CAM-Based Pattern Accumulated
298 Vector Method for Real-Time Character Recognition of License Plates", IJCSNS
299 International Journal of Computer Science and Network Security, Vol. 9 No. 1, 2009.
- 300 [26] A.F. Mollah, S. Basu, N. Das and R. Sarkar, "A Fast Skew Correction Technique for
301 Camera Captured Business Card Images ", Annual IEEE India Conference, INDICON,
302 2009.
- 303 [27] H. Hase, K. Tanabe, T.H.H. Tran and S. Tokai "Multi-Font Rotated Character
304 Recognition Using Periodicity", The Eighth IAPR International Workshop on Document
305 Analysis Systems, DAS'08, 2008.