Segmentation of Printed Gray Scale Dot Matrix Characters

Martin Grafmüller
Vision and Fusion Laboratory, Karlsruhe Institute of Technology (KIT)
76131 Karlsruhe, Germany

and

Jürgen Beyerer
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB)
76131 Karlsruhe, Germany

Abstract

There are many kinds of printed dot matrix fonts. They mainly differ in characters' structure since some of them are built with touching dots and some not. Two more variations an efficient character segmentation method has to cope with are fonts printed in italic and illumination changes depending on the environmental conditions. Hence, a procedure is introduced that uses the projection profile of the mean of several least gray values to estimate the skew angle of text lines and the characters' slant angle. Furthermore, based on the projection profile the decision boundaries for line and character segmentation are estimated with a least squares estimator. This has the advantage that the boundaries are adapted for every image. Finally, skew and slant angles are corrected and the characters are segmented. The robustness and accuracy is shown in experiments on data captured with an industrial camera.

Keywords: dot matrix characters, gray scale segmentation, slant and skew angle estimation

1. Introduction

In many industrial applications character recognition gets more and more important. The main tasks are reading serial numbers on packages or metal parts to record or track ways of transportation. Another application is reading dates of packaging, or expiring dates on food packaging to decide whether goods can still be delivered. In these applications the characters are printed or lasered on the materials mostly. This is the reason that depending on the application different fonts are used. Commonly, the characters are printed in a dot matrix font. With the font the structure of the characters can also vary significantly since they can be printed with dots that are touching or not, i.e., the dots of the characters are not connected. Due to that, segmentation of the characters gets more difficult since the decision where the character starts or ends is not necessarily unique. It even gets more difficult if the font is not a fixed pitch or monospace font. Furthermore, characters printed on an uneven surface can be slanted or skewed. In some cases wrong segmentation can also be caused by binarization of the image, i.e., this can cause characters that are merged or split depending on the binarization threshold.

Many character segmentation methods have been proposed in recent decades. The most common methods are based on projection analysis, connected component processing, or segmentation based on recognition [1], [2]. Some of them can even cope with touching, overlapping or broken characters, which are the main difficulties in character segmentation. However, most of the segmentation procedures are working on binary images. This has some shortcomings, since binarization can cause touching or fragmented characters. This means, that if binarization can be avoided many segmentation errors can be omitted. One segmentation method that works on gray scale images is proposed in [3]. The method for character segmentation is based on the ratio of vertical gray level projection and the sum of vertical gradients. According to the author, the procedure works good even on touching characters under severe conditions. Another method is introduced in [4]. They use projection profiles in combination with topographic features for pre-segmentation. Finally, the non-linear segmentation path is found by using multi stage graph search. They show that the method performs better than other methods that are based on binarization of the images. In [5] a method for character segmentation of license plates is introduced. The method is based on projection profile analysis as well but works in combination with the Hough transform. According to the authors it avoids rotation correction and is furthermore robust to illumination changes. Another approach for license plate character segmentation is introduced in [6]. Their method is based on binarized images on which the degradation—fragmented, overlapped, or connected characters—of the characters is adaptively detected. Based on the detection result corresponding morphological operations are applied to correct the degradation and thus improve the segmentation. Pan et al. introduced in [7] a segmentation method for characters in binarized license plate images as well. Their method estimates the skew angle of the text lines in horizontal and vertical direction with a least squares approach. For segmentation they use an improved projection method, which additionally removes noise regions in between the characters. A pitch-based segmentation method for dot matrix characters is proposed in [8]. For pitch estimation the author investigates three methods that are based on auto-correlation, Fourier analysis, and peak-valley analysis, respectively. The results show that the peak-valley analysis method performs best, but it fails in the case of skewed text lines or italic fonts. For this reason we introduce a method for segmentation of dot matrix printed characters from gray scale images without constraint to fixed pitch fonts. The segmentation decision is mainly based on a least squares estimation with respect to the image, i.e., the decision boundaries are adapted to any image. Additionally, the skew angle of the text lines and the characters' slant angle can be estimated and corrected.

The paper is organized as follows. In Section 2 the segmentation procedure for text line and character segmentation is introduced. Experimental results can be found in Section 3. Finally, in Section 4 conclusions and some remarks for future work are given.

2. Segmentation Procedure

In this section we introduce the segmentation procedure that extracts text lines and single characters from gray scale images. The input image is denoted by $G(m,n) \; : \; \{1,\ldots,M\} \times \{1,\ldots,N\} \to \{0,\ldots,255\}$, where M and N indicate the number of rows and columns, respectively. Furthermore, we assume that one image contains text only, printed in black with a skew angle within the limits of θ_{min} and θ_{max}. For the segmentation of the characters we first need to discuss how the text lines are segmented out of the image.

Segmentation of Text Lines

For text line segmentation and skew estimation the horizontal projection profile of the input image depending on different angles is calculated.

$$P_h(m,\theta) := \sum_{n\in N_K} G(m - \lfloor n \tan\theta \rfloor, n) \qquad (1)$$

with

$$\theta \in [\theta_{min}, \theta_{max}] \;, \quad m = 1,\ldots,M \;,$$

where $\lfloor x \rfloor$ indicates the largest integer $\leq x$ and N_K denotes the set of indices corresponding to the K least gray values of row m. K is usually chosen bigger than one to avoid that only one outlier is considered, but much less than N. It is chosen depending on the image dimension and the expected noise in the image. For the estimation of the skew angle the mean of the horizontal projection profile over all vertical pixels m is calculated according to

$$\bar{P}_h(\theta) := \frac{1}{M}\sum_{m=1}^{M} P_h(m,\theta), \quad \theta \in [\theta_{min}, \theta_{max}] \;.$$

Finally, the estimated skew angle is given by

$$\hat{\theta} := \arg\max_{\theta} \bar{P}_h(\theta) \;.$$

For the estimation of text lines and background the horizontal projection profile is divided into two subsets. The first one contains indices corresponding to the horizontal projection profile with all values smaller than the profile mean of the estimated skew angle

$$\bar{C}_0 := \left\{ m \in \{1,\ldots,M\} \mid P_h(m,\hat{\theta}) < \bar{P}_h(\hat{\theta}) \right\} \;,$$

and the second one with all indices for which the profile is bigger than or equal to the profile mean

$$\bar{C}_1 := \left\{ m \in \{1,\ldots,M\} \mid P_h(m,\hat{\theta}) \geq \bar{P}_h(\hat{\theta}) \right\} \;.$$

The in-subset-bounds $b_{\bar{C}0}(m)$ and $b_{\bar{C}1}(m)$ of the projection profile are determined with a least squares estimator assuming a quadratic polynomial with respect to $\bar{C}_0$ and $\bar{C}_1$, respectively. The quadratic polynomial estimation is used to make the procedure more robust to illumination differences in the image. Note, since the approximation is quadratic the bound is not constant but depends on m. Based on these bounds the upper and the lower bound for the decision of text line or background is performed by adding a constant value to the in-subset-bounds. This corresponds to a positive or negative shift of the parabolic in-subset-bounds. To determine the constant shift factor s_1 for the upper decision boundary a subset

$$C_1 := \left\{ m \in \{1,\ldots,M\} \mid P_h(m,\hat{\theta}) \geq b_{\bar{C}1}(m) \right\} \;,$$

is defined, which contains all m that correspond to the values of the horizontal projection bigger than or equal to the in-subset-bound $b_{\bar{C}1}(m)$. The minimization of the criterion function

$$J_1(s) := \sum_{m\in C_1} \left[\left(P_h(m,\hat{\theta}) - (b_{\bar{C}1}(m) + s) \right) \cdot P_h(m,\hat{\theta}) \right]^2$$

with respect to s yields that the upper in-subset-bound is shifted in such a way that the weighted distance to all projection values corresponding to C_1 is minimal. The weighting that is achieved through the multiplication by $P_h(m,\hat{\theta})$ yields to a stronger influence of bigger values of $P_h(m,\hat{\theta})$ corresponding to C_1. The minimum for the criterion function $J_1(s)$ is achieved for

$$s_1 = \frac{\sum\limits_{m\in C_1} \left(P_h(m,\hat{\theta}) - b_{\bar{C}1}(m) \right) \cdot P_h^2(m,\hat{\theta})}{\sum\limits_{m\in C_1} P_h^2(m,\hat{\theta})} \;,$$

and the upper decision bound is given by

$$b_1(m) := b_{\bar{C}1}(m) + s_1 \;, \quad m = 1,\ldots,M \;.$$

For the determination of the lower decision bound another criterion function

$$J_0(s) := \sum_{m=1}^{M} \left(b_{\bar{C}1}(m) - (b_{\bar{C}0}(m) + s) \right)^2 + s^2$$

is defined. The minimization of this function with respect to s yields to an offset that shifts the lower in-subset-bound to where the squared distance to the lower and upper in-subset-bound is minimal. Thus the shift factor for the lower in-subset-bound is

$$s_0 = \frac{\sum\limits_{m=1}^{M} \left(b_{\bar{C}1}(m) - b_{\bar{C}0}(m) \right)}{2M} \;,$$

and the lower decision bound is finally given by

$$b_0(m) := b_{\bar{C}0}(m) + s_0 \;, \quad m = 1,\ldots,M \;.$$

With both decision bounds the image can vertically segmented according to

$$\hat{\omega}(m) := \arg\min_{\omega} \left(P(m,\hat{\theta}) - b_\omega(m) \right)^2 \qquad (2)$$

with

$$\omega = 0, 1 \;, \quad m = 1,\ldots,M \;,$$

where $\hat{\omega}$ can either be 0 or 1 corresponding to text lines and background, respectively. Equation (2) can be efficiently solved by a dynamic programming algorithm, e.g., the Viterbi algorithm.

In Figure 1 an example for text line segmentation is given. In the top left corner we can see the input image, which contains a fixed pitch italic font with non-touching dots. The input image corrected by the estimated skew angle $\hat{\theta} = -7°$ is given on the top right hand side. Below the images a plot is given that shows

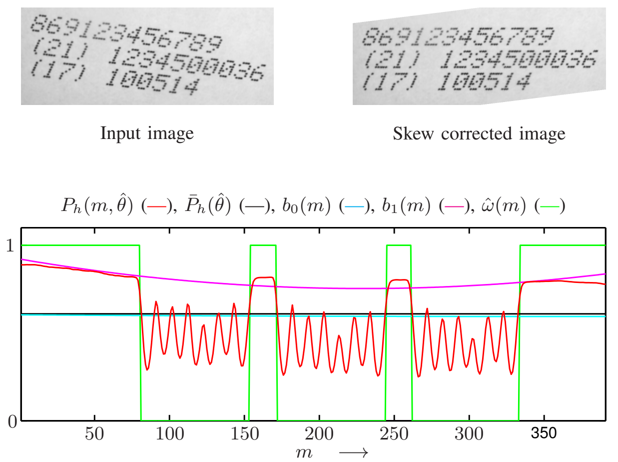

Figure 1. The input image with skewed text lines of an italic dot matrix font is given on the left hand side on top. On the upper right hand side the skew corrected image is given, with an estimated skew angle $\hat{\theta} = -7°$.

the horizontal projection profile $P_h(m, \hat{\theta})$ (—), the mean of the projection profile $\bar{P}_h(\hat{\theta})$ (—), the lower decision bound $b_0(m)$ (—), the upper decision bound $b_1(m)$ (—), and the segmentation function $\hat{\omega}(m)$ (—). Note, for a better illustration the result of $P_h(m, \hat{\theta})$, $\bar{P}_h(\hat{\theta})$, $b_0(m)$, and $b_1(m)$ plotted in Figure 1 is scaled between zero and one. If we look at the projection profile we can see the small peaks in between the text lines. They appear since the dots of the characters are not touching. The lower decision bound $b_0(m)$ does not approximate the foreground data very accurately, however, it is close to the highest peaks caused by the non-touching dots. Hence, these peaks do not affect the segmentation. If we look at the upper decision bound we can see that it almost approximates the highest values of the profile exactly. Thus, this decision bound is adapted to the slight illumination changes.

To further improve the robustness of the method we also use prior knowledge. Hence, we only accept lines with a certain height and they also have to have a certain distance to each other.

Finally, the result is a set of certain text line images, which have to be segmented further since we are interested in single characters. The segmentation procedure for characters is introduced in the following subsection.

Segmentation of Single Characters

The segmentation procedure for characters is basically the same as for text lines. The only difference is that either the input text line image $G_{line}(m, n) : \{1, \ldots, M\} \times \{1, \ldots, N\} \rightarrow \{0, \ldots, 255\}$ has to be rotated by $90°$ or the projection has to be applied in vertical direction. Thus, according to equation (1) the vertical projection of the text line image is given by

$$P_v(n, \phi) := \sum_{m \in M_K} G_{line}(m, n + \lfloor m \tan \phi \rfloor)$$

with

$$\phi \in [\phi_{min}, \phi_{max}] \ , \quad n = 1, \ldots, N \ ,$$

where ϕ indicates the slant angle of the characters and M_K denotes the set of indices corresponding to the K least gray values in column n. The further procedure is the same as described for text line segmentation up to equation (2) except that rows and columns are swapped. In Figure 2 one result of the character segmentation procedure is illustrated. On top we can see the input text line image for which the slant angle is estimated. The result of slant correction is shown in the image in the middle, whereas the estimated angle is $\hat{\phi} = -25°$. The plot on the bottom shows similarly to Figure 1 the vertical projection profile $P_v(n, \hat{\phi})$ (—), the mean of the projection profile $\bar{P}_v(\hat{\phi})$ (—), the lower decision bound $b_0(n)$ (—), the upper decision bound $b_1(n)$ (—), and segmentation function $\hat{\omega}(n)$ (—). Note, due to the vertical projection the variable m of the previous subsection becomes n. All functions given in Figure 2 are scaled between zero and one.

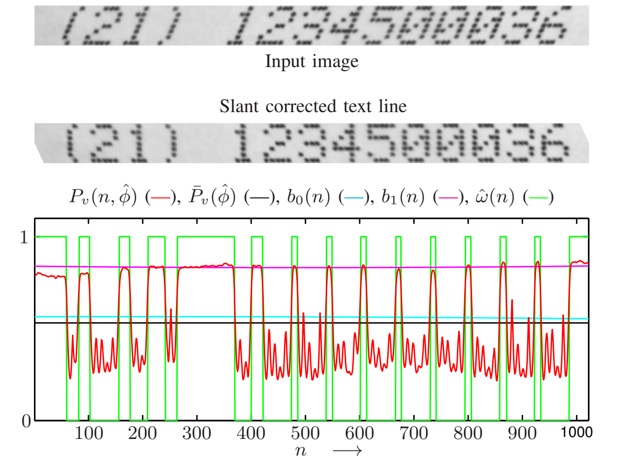

Figure 2. The input image is given on top. In the middle the slant corrected text line is given, with an estimated angle $\hat{\phi} = -25°$.

If we look at the estimated upper decision bound $b_1(n)$ in Figure 2, we can see that it is close to the maximum values—which represent the spacing between the characters—of the projection profile. Furthermore, we can see that the lower decision bound is very high. For that reason the characters are not split since the lower decision bound is adapted to the gaps in between the characters, where the projection profile shows higher peaks.

Similar to the text line segmentation method we also take prior knowledge into consideration. Since it is known that characters have certain dimensions it can be assumed that the character width must not fall below a certain minimal character width. Furthermore, the ratio of character width to character height is limited. If this ratio exceeds the limit the character segmentation method is applied recursively. That is, in the case of two merged characters this part is considered as one text line and the segmentation bound is determined to split these two characters. In fact it turned out that the recursive call of the character segmentation method correctly breaks many merging characters into their parts.

3. Experimental Results

For the experiments 126 images are captured with an industrial camera. The images contain different kind of fonts printed on paper or carton with different surfaces, i.e., matte or glossy surfaces. Furthermore, the images are captured in different angles to simulate slight distortions and rotations, and some are even a little blurred. The total number of text lines in all images is 458. They mostly contain numerals, dots, or parenthesis.

Table I
RESULT OF TEXT LINE SEGMENTATION OF 126 IMAGES WITH A TOTAL OF 458 TEXT LINES. K INDICATES THE NUMBER OF CONSIDERED VALUES IN % WITH RESPECT TO N AND SEGMENT. RATE THE NUMBER OF CORRECT SEGMENTED TEXT LINES IN %.

K [%]	segment. rate [%]	K [%]	segment. rate [%]	K [%]	segment. rate [%]
1	89.96	5	96.29	9	98.47
2	91.7	6	97.16	10	98.47
3	94.98	7	97.6	15	98.47
4	95.41	8	98.03	18	98.47

Table II
RESULT OF CHARACTER SEGMENTATION OF 451 TEXT LINES WITH A TOTAL OF 5527 CHARACTERS. K INDICATES THE NUMBER OF CONSIDERED VALUES IN % WITH RESPECT TO M AND SEGMENT. RATE THE NUMBER OF CORRECT SEGMENTED CHARACTERS IN %.

K [%]	segment. rate [%]	K [%]	segment. rate [%]
1	97.79	7	97.79
2	97.92	10	97.79
3	97.92	15	97.74
5	97.92	20	97.61

For this experiment the following settings are chosen. The skew angle of the text lines is within the limits $\theta_{min} = -12°$ and $\theta_{max} = 12°$. Moreover, we set the minimal line distance and the minimal line height to one and ten pixels, respectively. Table I shows the segmentation result for different number of K in percent with respect to N. As can be seen increases the rate of the correct segmented text lines continuously until K reaches 9% of N and remains constant up to $K = 18\% \cdot N$. All of the wrongly segmented text lines are still merged, most of them due to blurring or other degradations. For higher K the segmentation rate decreases due to the splitting of some text lines. An image with two text lines that are merged due to wrong segmentation is illustrated in Figure 3.
For character segmentation the result of $K = 10\% \cdot N$ excluding the three wrongly segmented text line images is used. These images are removed since correct character segmentation is not possible with this method. Hence, the experiment is performed with 451 text line images containing 5527 characters. The limits for the expected slant angles are chosen to $\phi_{min} = -10°$ and $\phi_{max} = 10°$. Furthermore, we set the minimal character distance to three pixels and the maximum feasible ratio of character width to character height to 1.1. Similar to the previous experiments we varied here K as well. However, in this case K is relating to the columns M of the text line images. We start with $K = 1\% \cdot M$, which for all images considers one pixel only for the calculation of the projection profile. This results in 5405 correctly segmented characters, which corresponds to 97.79%. The best result with 5412 segmented characters is given if K is chosen between 2% and 5% of M. With the further increase of the considered pixels the segmentation rate starts to decrease, which is caused by more splitting characters. The experimental result of character segmentation is summerized in

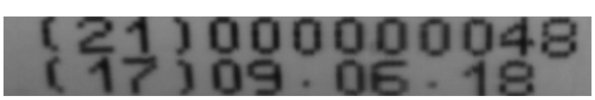

Figure 3. The two text lines are merged due to wrong segmentation.


Figure 4. Some merged characters due to wrong segmentation.


Figure 5. Two split characters due to wrong segmentation.

Table II. Some examples of merged characters are illustrated in Figure 4. We can see in the images on the left that some wrongly segmented characters are caused by a worse print. In some cases the characters are not even identifiable for humans. But some are not segmented correctly since for the segmentation method the space between the characters is not sufficient, see the image in the middle in Figure 4. The images on the right illustrate wrongly segmented characters caused by the parenthesis that is not touching the numeral but due to the projection profile the characters seem to be touching. On the other hand some characters are split into parts. Two examples can be found in Figure 5. This is mainly caused by the fixed—does not vary with respect to the font—ratio between character width and character height.

A. Discussion

The experiments in the previous section show the performance of the proposed segmentation method on images containing several kind of fonts printed on packages. Even dot matrix characters where the dots are not touching can be segmented, since the decision boundaries are estimated for every image independently, i.e., the decision boundaries are adaptively determined for every image, which makes the segmentation robust to slight illumination changes and low contrast. It has been demonstrated that the method can be employed to gray scale images, which avoids segmentation errors due to binarization. Furthermore, the procedure estimates and corrects the skew angle of the text lines and the slant angle of the characters, respectively.
We also want to mention some topics for further investigations, especially for speeding up the procedure. The minimum search for the sum of the projection profile is very time consuming, since it has to be performed for every row or column in the image, respectively. Time consuming as well is the skew and slant angle estimation, since for every angle the projection profile has to be determined. In the case of underlined characters the number K of the considered values in the projection profile has to be adapted, otherwise the method fails.

4. CONCLUSION

In this paper we have introduced a new methodology for segmentation of printed dot matrix characters. The segmentation procedure is mainly based on projection profiles on which the decision boundaries are adaptively estimated with a least squares approach. It has been pointed out that the procedure is applicable to gray scale images, which avoids character merging and degrading due to binarization. Furthermore, the procedure estimates and corrects the skew angle of the text lines and the characters' slant angle, respectively. Finally, the robustness of the procedure has been shown on data captured with an industrial camera.

One topic for future work is to improve the minimum search for the projection, since this is the main bottleneck with respect to computational complexity.

5. References

[1] Yi Lu, "Machine printed character segmentation – An overview," *Pattern Recognition*, vol. 28, no. 1, pp. 67–80, 1995.

[2] Richard G. Casey and Eric Lecolinet, "A Survey of Methods and Strategies in Character Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 690–706, 1996.

[3] F. LeBourgeois, "Robust Multifont OCR System from Gray Level Images," *International Conference on Document Analysis and Recognition*, vol. 0, p. 1, 1997.

[4] Dong-June Lee and Seong-Whan Lee, "A new methodology for gray-scale character segmentation and recognition," *International Conference on Document Analysis and Recognition*, vol. 1, pp. 524–527, 1995.

[5] Yungang Zhang and Changshui Zhang, "A New Algorithm for Character Segmentation of License Plate," *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pp. 106–109, 2003.

[6] Shigueo Nomura, Keiji Yamanaka, Osamu Katai, Hiroshi Kawakami, and Takayuki Shiose, "A novel adaptive morphological approach for degraded character image segmentation," *Pattern Recognition*, vol. 38, no. 11, pp. 1961–1975, 2005.

[7] Mei-Sen Pan, Jun-Biao Yan, and Zheng-Hong Xiao, "Vehicle license plate character segmentation," *International Journal of Automation and Computing*, vol. 5, pp. 425–432, 2008.

[8] Berrin A. Yanikoglu, "Pitch-based segmentation and recognition of dot-matrix text," *International Journal on Document Analysis and Recognition*, vol. 3, pp. 34–39, 2000.