Towards the Rectification of Highly Distorted Texts

Stefania Calarasanu[1], Séverine Dubuisson[2] and Jonathan Fabrizio[1]
[1]*EPITA-LRDE, 14-16, Rue Voltaire, F-94276, Le Kremlin Bicêtre, France*
[2]*Sorbonne Universités, UPMC Univ Paris 06, CNRS, UMR 7222, ISIR, F-75005, Paris, France*

Keywords: Text Rectification, Text in Perspective.

Abstract: A frequent challenge for many Text Understanding Systems is to tackle the variety of text characteristics in born-digital and natural scene images to which current OCRs are not well adapted. For example, texts in perspective are frequently present in real-word images, but despite the ability of some detectors to accurately localize such text objects, the recognition stage fails most of the time. Indeed, most OCRs are not designed to handle text strings in perspective but rather expect horizontal texts in a parallel-frontal plane to provide a correct transcription. In this paper, we propose a rectification procedure that can correct highly distorted texts, subject to rotation, shearing and perspective deformations. The method is based on an accurate estimation of the quadrangle bounding the deformed text in order to compute a homography to transform this quadrangle (and its content) into a horizontal rectangle. The rectification is validated on the dataset proposed during the ICDAR 2015 *Competition on Scene Text Rectification*.

1 INTRODUCTION

Retrieving the textual information from born-digital and real-scene images can often be a challenging task due to the variety of text properties (color, size, font, orientation) but also to external causes, such as lighting conditions (shadows, specularity, reflections, *etc*.), cluttered backgrounds, possible occlusions, poor image resolution and quality, or situations where the the text plane is not parallel to the camera one. These circumstances do not only affect the text detection process but also the text recognition stage. Unfortunately, most of the current OCRs have low performances on recognizing curved, inclined, vertical or perspective distorted texts. Such text examples are illustrated in Fig. 1. Our contributions concern a

Figure 1: Examples of real scene images with deformed texts from the ICDAR 2015 *Competition Scene Text Rectification dataset* (Liu and Wang, 2015).

rectification method that can simultaneously correct rotation, shearing and perspective deformations. It uses a homography that maps the image coordinates onto the world coordinate system and brings the deformed texts to a front-parallel view. The validation of this method is done on a recent dataset, used during the ICDAR 2015 *Competition on Scene Text Rectification* (Liu and Wang, 2015). It contains a very large amount of challenging texts, extracted from synthetic and real scenes, with different transformations. The organization of the paper is as follows: we first give a brief overview of the state of the art in Sec. 2, then the entire rectification procedure in Sec. 3, while the experimental results are presented in Sec. 4. Concluding remarks and perspectives are given in Sec. 5.

2 RELATED WORK

In the literature, the problem of managing distorted texts has been handled in different ways. Some works ((Santosh and Wendling, 2015), (Almazan et al., 2013)) tackled this by proposing powerful recognition stages capable of managing distorted characters. On the opposite, many works first rectify the distortions, then recognize the text. A first category of approaches relies on feature learning but, when texts are strongly distorted, these methods fail to provide a correct transcription. In such cases, the rectification procedure is a better alternative. Some text rectification procedures also target specific cases, such as multi-oriented, italic or texts in perspective. Some

Calarasanu, S., Dubuisson, S. and Fabrizio, J.
Towards the Rectification of Highly Distorted Texts.
DOI: 10.5220/0005772602410248
In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 4: VISAPP*, pages 241-248
ISBN: 978-989-758-175-5

works have exclusively focused on rectifying italic texts to enhance the performance of OCRs that have difficulties in providing an accurate transcription of sheared texts. The authors in (Zhang et al., 2004) proposed an approach based on the statistical analysis of stroke patterns extracted from the wavelet decomposition of text images. In (Fan and Huang, 2005) authors introduced a method that rectifies italic texts using a shear transform. First, the characters are classified into three classes of angles. Then, the shear angle is determined differently for each character based on its corresponding italic class. Perspective recovery needs to be applied when the camera optical axis is not perpendicular to the text plane. When a text is in perspective, the characters change their original structure. This makes the OCR performs poorly and produces low accuracy scores. However, a series of works proposed recognition modules capable of identifying oriented characters or texts in perspective. The authors in (Lu and Tan, 2006) proposed a recognition technique capable of recognizing characters in perspective by extracting perspective invariant features, such as character ascenders and descenders or number of centroid intersections. Cross ratio spectrum and Dynamic Time Wrapping techniques were employed during the recognition process in (Li and Tan, 2008), (Zhou et al., 2009). In (Phan et al., 2013) SIFT features were extracted to recognize texts in perspective in different orientations. To correct the perspective distortion, many works rely on the homography transformation (Myers et al., 2005), (Ye et al., 2007), (Cambra and Murillo, 2011), (Kiran and Murali, 2013). In (Ye et al., 2007), the rectification is done based on a correlation between a set of feature points and a plane-to-plane homography transformation. The extension of this work, presented in (Cambra and Murillo, 2011), consists of an optimization of the parameters of the homography. The method in (Kiran and Murali, 2013) implied a first stage where text borders are captured using a geometry based segmentation and then feature points are extracted using the Harris corner detector. The authors in (Merino-Gracia et al., 2013) performed a parallel rectification using a homography and a shearing transform. The method first proposes a horizontal foreshortening by detecting the upper and lower lines bounding the text region. Next, vertical foreshortening and shearing are done by using a linear regression based on the variation of shear characters. The authors in (Chen et al., 2004) used an affine transformation to correct the perspective deformations, but the method requires the camera parameters to be known. Such an assumption was also required in the work in (Clark et al., 2001). The borderline analysis was implied in (Ferreira et al., 2005). The main problem of these approaches is that they rely on the hypothesis that text regions are bounded by rectangles. The work in (Zhang et al., 2013) used the Transformed Invariant Low-rank Textures algorithm to rectify English, Chinese characters and digits. The method presented in (Busta et al., 2015) proposed a skew text rectification in real scene images based on five skew estimators used for character segmentation (or polygon approximation): vertical dominant, vertical dominant on convex hull, longest edge, thinnest profile and symmetric glyph. In (Myers et al., 2005), the authors used a projective transformation to correct text in perspective. The parameters used for the rectification are derived from a series of features extracted from each text line, such as top and baselines of a text, or the dominant vertical direction of character strokes. In (Yonemoto, 2014) a correction method based on a quadrangle estimation is proposed, which supposes that the text contains a sufficient number of horizontal and vertical strokes. Authors in (Hase et al., 2001) proposed a generic method to correct inclined, curved and distorted texts. Text is first classified with respect to the alignment and distortion of its characters, then different types of corrections are applied. A rectification approach for license plate images was proposed (Deng et al., 2014) using the Hough transform and different types of projections. The method, based on finding parallel lines, consists of applying two transformations: a horizontal tilt and a vertical shear transform. Many of the approaches discussed above correct the text of individual text lines. Some works proposed rectification algorithms on whole documents. The work in (Stamatopoulos et al., 2011) targeted the rectification of distorted documents. It performs a curved surface projection, a word baseline fitting and a horizontal alignment. The authors in (Liang et al., 2008) proposed a rectification method for planar and curved documents by estimating $3D$ document shapes from texture flow information. Contrary to works previously cited, that use the same global approach and which imply an affine transformation for perspective correction followed by a shearing rectification to correct the perspective distortions, our proposed method has the advantage of using a single affine transformation that can rectify, individually or simultaneously, texts subject to rotation, shearing and perspective deformations. This transformation is obtained from a very accurate quadrangle estimation of the distorted text, described in the next section.

3 PROPOSED METHOD

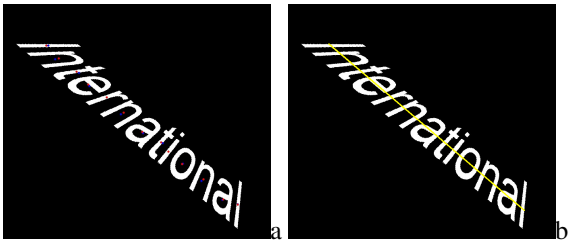
Let us consider a text string as a set of N characters defined as $\mathcal{C} = \{C_i\}_{i=1..N}$, where C_i is the individual CC corresponding to the i^{th} character. We call $\mathcal{G} = \{G_i\}_{i=1..N}$ the set of the centroids corresponding to each C_i in $\mathcal{C}$. Similarly, we denote by $\mathcal{W} = \{W_i\}_{i=1..N}$ the set of the weighted centroids that belong to each C_i in $\mathcal{C}$. We classify the CCs into two categories: *extremity* CC*s* corresponding to the first (C_{e1}) or last (C_{e2}) characters of the text string (in the order of reading); and *inner* CCs corresponding to any of the characters that are located between the two extremity CCs.

3.1 Overview of the Rectification Process

The perspective rectification process relies on finding a homography matrix in order to rectify the text image. This is done implying several stages. The method first relies on a CC filtering, described in Sec. 3.2 during which punctuation signs and point over some characters are temporarily removed. The filtering is followed by an extremity CC identification procedure, discussed in Sec. 3.3, which aims at identifying of the first and last characters of a text string. The process then precisely estimates a quadrangle (see Sec. 3.4) that bounds the distorted text. The four points that define the quadrangle are then used to compute the homography matrix. Finally, this homography matrix is used to map all the points of the deformed text onto a parallel-front plane, as explained in Sec. 3.5.

3.2 Connected Component Filtering

Before applying the rectification, we need to filter the CCs and remove punctuation marks such as ".", "," or ":" and points over some characters such as "i" and "j". Such a removal is needed because the entire text correction is based on the relative position of a CC with respect to the other ones. We define l_d^i the length of the diagonal of the box bounding C_i computed as: $l_d^i = \sqrt{h_{C_i}^2 + w_{C_i}^2}$, where h_{C_i} and w_{C_i} are respectively the height and width of the bounding box of C_i. Hence, C_i is kept during the filtering procedure as long as its diagonal satisfies the following constraint: $l_d^i > l_d^{av} \cdot T_{pt}$, with $l_d^{av} = \frac{\sum_{i=1}^{N} l_d^i}{N}$ where l_d^{av} is the average of all diagonal lengths and T_{pt} is a threshold that was experimentally set to 0.35. This constraint removes all CCs whose diagonal is considerably smaller than the average diagonal.



a b

Figure 2: Centroids and reference line fitting using LSM: (a) weighted centroids (red); (b) the reference line that best fits the weighted centroids (yellow).

3.3 Extremity Connected Components

After the CC filtering, we need to find the two extremity CCs, *i.e.* corresponding to the first and last characters. This requires the steps described below.

Fitting the Reference Line. An approximation of the text orientation is obtained by using the Least-Square Method (LSM) to fit a reference line to the set of weighted centroids $\mathcal{W}$. The slope of this line, called L_{Ref} gives an approximation of the text orientation. Fig. 2 shows examples of centroids and a reference line for the text string "International".

Finding the Two Extremity CCs. First we identify the two closest neighbors of each CC C_i, denoted as C_i^{n1} and C_i^{n2}. If C_i is the first extremity, then its two nearest neighbors will be the two following characters. If C_i is the last extremity, they will be its two preceding characters. If C_i an inner CC, its two closest neighbors will be its predecessor and its successor. Let W_i^{n1} and W_i^{n2} be the weighted centroids of the two neighbors of C_i. We then define l_i^{n1} and l_i^{n2} the lines that unite W_i and, respectively, W_i^{n1} and W_i^{n2}: $l_i^{n1} = (W_i^{n1}, W_i)$, $l_i^{n2} = (W_i^{n2}, W_i)$ Let θ_i be the orientation angle of C_i, computed as $\theta_i = angle(l_i^{n1}, l_i^{n2})$. All CCs for which this angle is smaller than 45 degrees are selected as extremity CC candidates. If more than two CCs satisfy this constraint, we compute the largest distance between each pair of candidate CCs. The pair of CCs for which the distance between their centroids is the largest are identified as the two extremities C_{e1} and C_{e2}, with $e1, e2 \in [1, N]$. This stage is illustrated in Fig. 3(a) and 3(b).

Identifying the First and Last Extremities. Based on the slope $m(L_{Ref})$ of L_{Ref} (see below), we can find the two extremities C_{e1} and C_{e2}. If the slope $m(L_{Ref}) \in [-0.1, 0.1]$, the text is considered as horizontal and hence we determine the first and last characters depending on the x-coordinates of the weighted

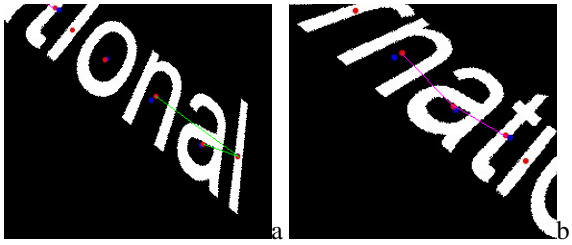


Figure 3: The procedure for finding the extremity CCs: (a) the angle between the lines (in green) uniting the centroids of an extremity CC and the centroids of its two closest neighbors; (b) the angle between the lines (in magenta) uniting the centroid of an inner CC ("a") and the centroids of its two closest neighbors ("n" and "t").

centroids of the two CCs. If $m(L_{Ref}) < -0.1$, the text is inclined following a bottom-left to top-right direction. In this case we choose the CC closer to the bottom origin point defined as $O_b = (0, y_{max})$. If $m(r) > 0.1$, the text follows a top-left to bottom-right direction and the first and last characters are chosen based on the smallest distance between the upper origin point $O_u = (0,0)$ and the two centroids W_{e1} and W_{e2}.

3.4 Quadrangle Approximation

We are now interested in finding the quadrangle that best fits a text string. This consists in identifying the four lines that bound the text, referred here as the bottom (L_b), upper (L_u), left (L_l) and right (L_r) lines.

Bottom and Top Boundary Line Fitting. Let us consider $\mathcal{P}^u = \{P_i^u\}$ and $\mathcal{P}^b = \{P_i^b\}$ the sets containing the upper and lower extremity points of C_i respectively. In order to find these points we use the slope of the reference line as a guideline:

1. We consider lines parallel to L_{Ref} at different distances d in two directions (positive and negative) corresponding to the upper and bottom points. We denote these lines L_s^d, where s is the direction sign. The procedure consists in, for each direction sign, increasing d by one, computing intersections between L_s^d and CC and storing them into $\mathcal{P}^u$ and $\mathcal{P}^b$, until L_s^d do not intersect any CC anymore (Fig. 4(a) and 4(b)).
2. The LSM is then used to fit L_u on the set of points $\mathcal{P}^u$ and L_b on $\mathcal{P}^b$ (Fig. 4(c)).
3. Finally, we check if L_u and L_b correctly bound the text string. If L_u or L_b intersects the set of CCs $\mathcal{C}$, the lines are shifted (parallel to L_u or L_b) until they perfectly bound the text string (Fig. 4(d)).

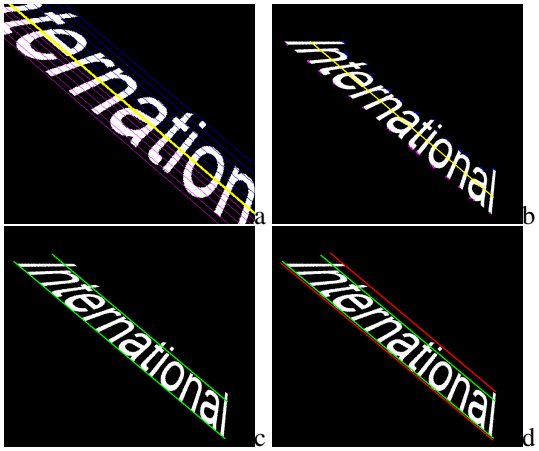


Figure 4: Bottom and up boundary line fitting procedure: (a) parallels to the reference line in both directions: upper (blue) and bottom (magenta); (b) extremity points: upper (blue) and bottom (magenta); (c) LSM line fitting of the lower and bottom extremity points; (d) shifting of the initial upper and lower lines.

Left and Right Boundary Line Fitting. We call $\mathcal{P}^l = \{P_i^l\}$ the set containing the left extremity points and $\mathcal{P}^r = \{P_i^r\}$ the set containing the right extremity points. To get the left and right boundary lines, the positions of the first and last CCs are used, following the stages described below:

1. Let us define L_{Ref}^P the line perpendicular to L_{Ref} that passes through W_i. The procedure consists of tracing parallels to L_{Ref}^P until the parallel lines do not intersect anymore the CC extremities. All border points that belong to both the last parallel line and the extremity CC are considered as extremity points and stored into the two sets $\mathcal{P}^l$ and $\mathcal{P}^r$ (Fig. 5(a) and 5(b)).
2. For each of the two sets $\mathcal{P}^l$ and $\mathcal{P}^r$ average left point P_{av}^l and right point P_{av}^r are computed:

$$P_{av}^l = \left(\frac{\sum_j^{|\mathcal{P}^l|} x_{P_j^l}}{|\mathcal{P}_l|}, \frac{\sum_j^{|\mathcal{P}^l|} y_{P_j^l}}{|\mathcal{P}_l|} \right), \quad P_{av}^r = \left(\frac{\sum_j^{|\mathcal{P}^r|} x_{P_j^r}}{|\mathcal{P}_r|}, \frac{\sum_j^{|\mathcal{P}^r|} y_{P_j^r}}{|\mathcal{P}_r|} \right)$$

3. We look for the lines L_l (resp. L_r) that best fit P_{av}^l (resp. P_{av}^r) and C_{e1} (resp. C_{e2}). Let us define L_P the line perpendicular to L_{Ref} passing through P_{av}^l. The best fitting left line is obtained by rotating L_P until it covers the maximum number of border points (Fig. 5(c)). The same reasoning is done to obtain L_r (see Fig. 5(d)).
4. Finally, we check if L_l and L_r correctly bound the text string. If L_l or L_r intersects the set of CCs $\mathcal{C}$, the lines are shifted (parallel to L_l or L_r) until they perfectly bound the text string.

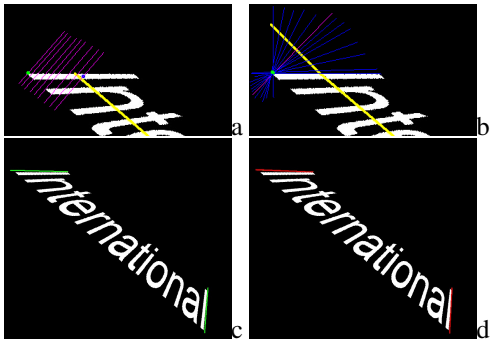


Figure 5: Left and right boundary line fitting procedure: (a) finding the left extremity points; (b) line variation to find the best fitting left lines; (c) left and right boundary lines; (d) left and right boundary lines after shifting.

Figure 6: Perspective distortion rectification: (a) bounding quadrangle estimation; (b) rectified text.

3.5 Homography

The relationship that maps a point (x,y) from a projective plane to a point (x',y') from another projective plane is defined as: $(x',y',1)^T = H(x,y,1)^T$, where H the homography matrix. To compute H we need eight points: four points from the image plane and their corresponding four points from the real world plane. In our case, we use the four corners of the *input* quadrilateral that bounds the distorted text and provide the coordinates of the *output* rectangular plane onto which we want to map this text string. By using approximations of lines L_u, L_b, L_l and L_r we can get the four corners (P_1, P_2, P_3 and P_4) as their intersections. The perspective deformation is then corrected transforming all points (x,y) that belong to the input quadrangle and get their corresponding positions (x',y') into the output rectangle. Fig. 6 illustrates a rectification result after applying the homography.

4 EXPERIMENTAL RESULTS

We evaluate the performance of our rectification method on two datasets proposed for the ICDAR 2015 *Competition on Scene Text Rectification* (Liu and Wang, 2015). The first dataset contains 2500 English and Chinese synthetic texts obtained by applying on 1000 text samples random deformation types, such as rotation, shearing, horizontal fore-shortening and vertical fore-shortening with different parameters. The second dataset is derived from real-scene MSRA-TD500 dataset (Yao, 2012). It contains 60 image samples of English, respectively 60 images with Chinese texts, subject to orientation and perspective distortions.

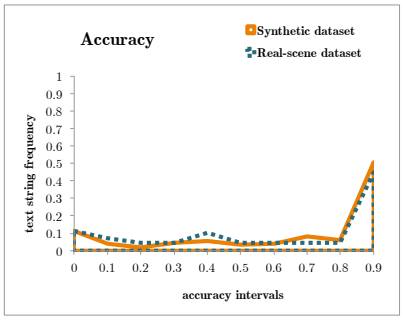


Figure 7: Quality-quantity histograms for text rectification. Accuracy values for: (a) synthetic text; (b) real-scene text.

The evaluation of the rectification method is done using the performance measurements proposed during the ICDAR 2015 *Competition on Scene Text Rectification* (Liu and Wang, 2015). The OCR accuracy (*Acc*) is computed between the recognition result R of a rectified text string and its corresponding GT transcription G defined as: $Acc(R,G) = \frac{1-L(R,G)}{\max(|R|,|G|)}$, where $|R|$ and $|G|$ represent the length of the two strings and $L(R,G)$ the Levenshtein distance between R and G. The rectification performance metric considers the OCR results before and after the rectification and is defined as: $RP(R,D,G) = Acc(R,G) - Acc(D,G)$, where D is the recognition result of the distorted text before applying the rectification. *RP* reflects both the impact of the rectification method on the final OCR transcription but also the level of difficulty of the text recognition process. In our experiments we used the Tesseract OCR engine (Smith, 2007) to obtain the recognition results.

We define Acc^b_{OV} and Acc^a_{OV} as the overall accuracy performance before and after the rectification over a dataset of N text strings, computed as: $Acc^b_{OV} = \frac{\sum_i^N Acc(D_i,G_i)}{N}$ and $Acc^a_{OV} = \frac{\sum_i^N Acc(R_i,G_i)}{N}$ Similarly, we define the overall rectification performance *RP* as: $RP_{OV} = \frac{\sum_i^N RP(R_i,D_i,G_i)}{N}$.

The validation of the proposed rectification approach is exclusively done based on our results, as no result of any participant at the ICDAR 2015 *Competition on Scene Text Rectification* is publicly available. Table 1 exposes the scores from both datasets.

Figure 8: Rectification results on the synthetic dataset: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

Discussion on the Results Obtained on the Synthetic Dataset. Fig. 8 illustrates rectification results of some synthetic deformed texts. The accuracy on the synthetic dataset (see Table 1) is evaluated to approximately 0.72. On the other hand, the accuracy before the rectification is very low (0.08), which indicates the difficulty to deal with text string deformations and also the efficiency of our method. Hence, the rectification performance *RP* is equal to 0.64. Fig. 7 illustrates the quantity-quality histograms (Calarasanu et al., 2015) containing the distributions of accuracy values. By looking at the frequency in the last bin of the histogram, one can notice that half of the rectified texts have obtained almost perfect recognition accuracies (*i.e.* accuracy values in the intervals $[0.8, 0.9[$ and $[0.9, 1]$). Approximately 10% of the texts got a low accuracy rate, belonging to interval $[0, 0.1[$ which corresponds to rectification failures.

Table 1: Rectification evaluation results on ICDAR 2015 *Competition on Scene Text Rectification* datasets.

DATASET	Acc^a_{OV}	*RP*	Acc^b_{OV}
Synthetic	0.721979	0.637037	0.0849421
Real-scene (EN)	0.65149	0.187165	0.464326

Most of the problems mainly come from an incorrect approximation of the quadrangle that bounds the deformed text. The left and right bounding lines do not always get the best orientation of the extremity

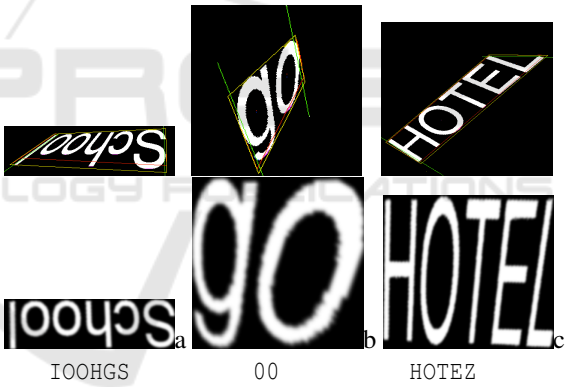


Figure 9: Rectification failures: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom); (a) upward rectified text strings; (b) rectified text strings with disproportionate character sizes; (c) text with an extremity character "T".

Figure 10: OCR recognition failures due to (a) challenging font, (b) small text size and (c) complex design: original image (top), text rectification result (middle) and OCR transcription using Tesseract (bottom).

Figure 11: OCR recognition before and after the rectification process: original image (top), text rectification result (middle), OCR transcription before the rectification and OCR transcription after the rectification using Tesseract (bottom).

CC. For characters such as "A", "T" or "L" the procedure looks for the lines that maximize the number of border points. However, these lines have a different direction than the one of the characters and hence this can produce inaccurate rectification as seen in Fig. 9(c). Furthermore, an imprecise approximation of the quadrangle can be caused by the upper and/or bottom bounding lines which can cause the disproportion of character sizes as seen in Fig. 9(b). Here, the upper and lower lines are not approximations, but the unique lines passing through the upper and bottom extremity points which produces wrong lines (*i.e.* not parallel to the real direction of the text). The OCR also produces erroneous text transcriptions when the deformed texts are upward (Fig. 9(a)).

One of the advantages of the proposed rectification method is its ability to correct very challenging deformations, that make text strings unreadable. Although the rectification is not always very accurate, which consequently leads to imprecise OCR transcriptions, the visual results remain notable. From a visual point of view, we succeed in transforming unreadable texts into readable ones. Such examples are depicted in Fig. 8. The recognition performance of Tesseract when dealing with inclined texts varies depending on the case. For example, the OCR performs better on the last part of the sequence "Gt. Yarmouth" ("mouth") than on the apparently easier to read part "Gt. Ya", depicted in the last example of Fig. 8.

Discussion on the Results Obtained on the Real-scene Dataset. The accuracy score obtained on the real-scene dataset (see Table 1) is slightly lower than the one obtained on the synthetic dataset (approximately 0.65). The rectification performance is very low, equal to 0.19, due to many reasons such as: challenging fonts that are not correctly handled by the OCR (Fig. 10(a)); texts with small characters, which can affect the rectification process; complex text designs in which characters are composed of multiple CCs (Fig. 10(b)) for which the rectification method fails, as it can only handle characters represented by one CC; the dataset contains few challenging text distortions (mainly oriented texts and few text strings in perspective) which explains the low value of *RP* in Table 1 compared to the synthetic dataset (Fig. 11); dataset size (60 images, versus 2000 images in the synthetic dataset) which leads to less representative scores.

Moreover, the accuracy histogram in Fig. 7 shows that approximately the same proportion of deformed texts have been incorrectly rectified as in the case of the synthetic datatset. On the other hand, the distribution of accuracy values is compacted into the intervals $[0.3, 0.5[$, $[0.8, 0.9[$ and $[0.9, 1]$, whereas the values computed on the synthetic dataset were more scattered. Nonetheless, both histograms in Fig. 7 present a similar behavior which validates the fact that the proposed rectification method is independent of the text type, *i.e.* synthetic or natural.

5 CONCLUSION

In this paper we have presented a perspective rectification method that accurately corrects highly deformed text strings. The proposed perspective rectification relies on a homographic transformation that maps the camera coordinates onto a parallel-front plane. The homographic transformation is powerful as it handles both rotation and perspective projections, including shearing effects, but depends on how accurate is the estimation of the bounding quadrangle of a distorted text. Our proposed two-stage procedure to find this quadrangle consisted in approximating a top and bottom boundary lines based on a reference line computed using the LSM. Secondly, we provide a precise estimation of the lines bounding the extremity characters by iterating all possible lines until finding the one that best bounds the two CCs. This technique implies however some limitations. Namely, the deformed text should contain characters represented by single CCs and moreover the text should be upward only (rotated or not). The experiments were conducted on the two ICDAR 2015 *Competition on Scene Text Rectification* datasets, for which the recti-

fication procedure gives similar performance results. A slightly lower recognition accuracy was obtained on the real-scene dataset due to a number of reasons, such as the low performance of Tesseract OCR on texts with complex fonts or designs. On the other hand, many texts in this dataset present only rotation or slight perspective deformations, compared to the synthetic dataset, which contains more challenging texts that are subject to multiple transformations at the same time. The difficulty of the synthetic dataset is also proven by the high rectification performance score. We have demonstrated that the proposed rectification method can successfully correct oriented, sheared or perspective distorted texts. We have also shown that we could rectify unreadable texts and obtain satisfactory OCR accuracy scores. Future perspectives focus on a deeper analysis on the shape of some characters such as "A", "L" or "T" namely a study of their symmetry to prevent inaccurate quadrangle approximations. The evaluation of the rectification procedure is influenced by the used OCR. Tesseract expects a very accurate text rectification and often fails when the characters are slightly inclined. For example, the letter "t" is often interpreted as "f", "l" as the symbol "\", "L" as "Z". Hence, more performant OCRs are being considered for further tests such as CuneiForm, ABBYY or OmniPage.

ACKNOWLEDGEMENTS

This work was supported by FUI 14 (LINX project).

REFERENCES

Almazan, J., Fornes, A., and Valveny, E. (2013). Deformable hog-based shape descriptor. In *ICDAR*, pages 1022–1026.

Busta, M., Drtina, T., Helekal, D., Neumann, L., and Matas, J. (2015). Efficient character skew rectification in scene text images. In *ACCV*, pages 134–146.

Calarasanu, S., Fabrizio, J., and Dubuisson, S. (2015). Using histogram representation and earth mover's distance as an evaluation tool for text detection. In *ICDAR*.

Cambra, A. and Murillo, A. (2011). Towards robust and efficient text sign reading from a mobile phone. In *ICCV*, pages 64–71.

Chen, X., Yang, J., Zhang, J., and Waibel, A. (2004). Automatic detection and recognition of signs from natural scenes. *TIP*, 13(1):87–99.

Clark, P., Mirmehdi, D., and Doermann, D. (2001). Recognizing text in real scenes. *IJDAR*, 4:243–257.

Deng, H., Zhu, Q., Tao, J., and Feng, H. (2014). Rectification of license plate images based on hough transformation and projection. *TELKOMNIKA IJEE*, 12(1):584–591.

Fan, K. C. and Huang, C. H. (2005). Italic detection and rectification. *JISE*, 23:403–419.

Ferreira, S., Garin, V., and Gosselini, B. (2005). A text detection technique applied in the framework of a mobile camera-based application. In *CBDAR*, pages 133–139.

Hase, H., Yoneda, M., Shinokawa, T., and Suen, C. (2001). Alignment of free layout color texts for character recognition. In *ICDAR*, pages 932–936.

Kiran, A. G. and Murali, S. (2013). Automatic rectification of perspective distortion from a single image using plane homography. *IJCSA*, 3(5):47–58.

Li, L. and Tan, C. (2008). Character recognition under severe perspective distortion. In *ICPR*.

Liang, J., DeMenthon, D., and Doermann, D. (2008). Geometric rectification of camera-captured document images. *PAMI*, 30(4):591–605.

Liu, C. and Wang, B. (2015). Icdar 2015 competition on scene text rectification. http://ocrserv.ee.tsinghua.edu.cn/icdar2015_str/.

Lu, S. and Tan, C. (2006). Camera text recognition based on perspective invariants. In *ICPR*, volume 2, pages 1042–1045.

Merino-Gracia, C., Mirmehdi, M., Sigut, J., and González-Mora, J. L. (2013). Fast perspective recovery of text in natural scenes. *IVC*, 31(10):714–724.

Myers, G., Bolles, R., Luong, Q.-T., Herson, J., and Aradhye, H. (2005). Rectification and recognition of text in 3-d scenes. *IJDAR*, 7(2-3):147–158.

Phan, T. Q., Shivakumara, P., Tian, S., and Tan, C. L. (2013). Recognizing text with perspective distortion in natural scenes. In *ICCV*, pages 569–576.

Santosh, K. and Wendling, L. (2015). Character recognition based on non-linear multi-projection profiles measure. *FCS*, 9(5):678–690.

Smith, R. (2007). An overview of the tesseract ocr engine. In *ICDAR*, pages 629–633.

Stamatopoulos, N., Gatos, B., Pratikakis, I., and Perantonis, S. (2011). Goal-oriented rectification of camera-based document images. *TIP*, 20(4):910–920.

Yao, C. (2012). Detecting texts of arbitrary orientations in natural images. In *CVPR*, pages 1083–1090.

Ye, Q., Jiao, J., Huang, J., and Yu, H. (2007). Text detection and restoration in natural scene images. *VCIR*, 18(6):504–513.

Yonemoto, S. (2014). A method for text detection and rectification in real-world images. In *ICIV*, pages 374–377.

Zhang, L., Lu, Y., and Tan, C. (2004). Italic font recognition using stroke pattern analysis on wavelet decomposed word images. In *ICPR*, volume 4, pages 835–838.

Zhang, X., Lin, Z., Sun, F., and Ma, Y. (2013). Rectification of optical characters as transform invariant low-rank textures. In *ICDAR*, pages 393–397.

Zhou, P., Li, L., and Tan, C. (2009). Character recognition under severe perspective distortion. In *ICDAR*, pages 676–680.