


Rectification of Optical Characters as Transform Invariant Low-rank Textures

Xin Zhang*, Zhouchen Lin [†], Fuchun Sun*, Yi Ma[‡]

*School of Computer Science, Tsinghua University, Beijing, China 100084

[†]Key Lab. of Machine Perception (MOE), School of EECS, Peking University, Beijing, China 100871. (zlin@pku.edu.cn)

[‡]Microsoft Research Asia, Beijing, China

Abstract—Character rectification is very important for character recognition. Front view standard character images are much easier to recognize since most character recognition algorithms were trained with such data. However, the existing text rectification methods only work for a paragraph or a page. We discover that the modified TILT algorithm can be applied to rectify many *single* Chinese, English, and digit characters robustly. By changing the character image into a low-rank texture image via binarization and graylevel inversion, the modified TILT method applies a rank minimization technique to recover the deformation and the proposed algorithm can work for almost all characters. To further enhance the robustness of the proposed algorithm, the modified TILT algorithm is extended for short phrases that consist of multiple characters. Extensive experiments testify to the effectiveness of the proposed method in rectifying texts with significant affine or perspective deformation in real images, such as street signs taken by mobile phones.

I. INTRODUCTION

Texts embedded in natural image always provide large amount of semantic information of a scene. An efficient scene text reading system can be of great help for many applications, such as image automatic annotation, robot navigation, and visual perception aid. Different from document images captured by scanners, most scene images are captured by cameras or cell phones. Those devices are quite easy to use which allow taking pictures from any viewpoint. The flexibility of using cameras also introduces the distortion problem. So distorted texts are abundant in daily life images, e.g., street views, road signs, and menus. Those distorted texts bring about recognition challenges because some features are no longer similar to the undistorted training text images. Thus the performance of traditional text recognition systems (such as Optical Character Recognition) is greatly reduced. So it is of great importance to rectify distorted text images.

There have been lots of techniques, e.g., [1], developed in the past to preprocess and rectify distorted text documents. Almost all these techniques rely on the global regular layout of the texts to rectify the distortion. That is, the rectified texts should lie on a set of horizontal and parallel lines, which are often in a rectangular region. Hence, many different methods have been developed to estimate the rotation or skew angle based on statistics of the distorted texts compared to the standard layout. Such methods include those based on projection profiles [2][3], Hough transform for gradient/edge

directions [4], morphology of the text region [5], and cross-correlation of image blocks [6], etc.

While many methods have been developed to rectify a large region (say a paragraph or a page) of texts, to our best knowledge, few methods aim to work at the level of an *individual character* or a *short phrase*. However, short phrase images are very common in daily life, such as book titles, menus, and street signs. Applying previous text rectification algorithms on these kinds of texts poses a new challenge: such text regions only consist of very few characters or words. Thus, one can no longer rely on the dominant horizontal or vertical layout of texts to do rectification.

In this paper, we propose a new method that can effectively rectify texts at all scales: an individual character, a short phrase, or a large paragraph. Although many Chinese and English characters or words do not have dominant horizontal or vertical edges, we discover that the robust matrix rank of the text image is a good indicator of their standard position (see Figure 3). This is because it is a *holistic measure of regularity* for the image region of interest and thus can harness a much richer set of local or global regularities (such as bilateral symmetry) than edge orientations (see Figure 1 for examples). Our work is inspired by a recent work on *transform invariant low-rank textures* (TILT) [7]. The original TILT algorithm works well for images with rich repeated or symmetric textures, such as building facade and animal face. However, a straightforward application of TILT to text images can only bring about a moderate performance in rectification because text images contain much less texture and there may not be perfect symmetry for some characters. So we propose binarizing and inverting the images, such that the characters are of value one (white) and the background is of value zero (black). Such preprocessing, although simple, can greatly enhance the rectification performance of TILT. We further extend our method to handle multiple characters on a line that share the same transform. Since multiple characters can provide more text distortion clues, it will enhance the estimated distortion accuracy for short phrase images. In this paper, we deal with both affine and perspective projections that suffice for the mobile phone based recognition scenario, although the method in principle can be generalized to handle other parametric transforms (say for curved surfaces). To simplify the presentation, we

primarily use Chinese characters to illustrate our ideas and demonstrate the results, although the techniques could be used for texts of other languages (as long as their characters are rich of regularity).

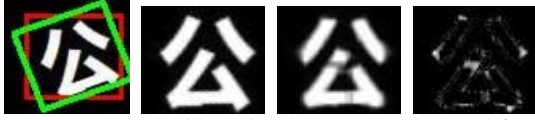


Figure 1. Two examples illustrating the TILT model. From left to right: the deformed text D , the rectified text $D \circ \tau$, the low-rank part A , and the sparse-error part E of the rectified text.

II. SINGLE CHARACTER RECTIFICATION AS LOW-RANK TEXTURES

Standard fonts of English and Chinese characters share something in common: they are typically rich of regular structures. Many Chinese characters are rich of horizontal and vertical strokes and English characters are rich of local or global (bilateral) symmetry. Hence, mathematically, if we view the character image as a matrix, the matrix will have plenty of linear correlation among its columns and rows with some proper processing. Thus, front view character image can be considered having Low-rank Textures. In addition to that, Zhang et. al [7] proposed an algorithm that can estimate the distortion of image so long as the image has low rank texture. However, different from other low-rank textures containing rich textures, such as building facade, the texture of a character image is relatively less. So in the following, we will first discuss how to make the text image as a low rank texture. Then we will discuss how to estimate the distortion parameters for a single character image.

A. Making a Text Image as Low-rank Textures

Since characters consist of strokes, if the entire column in a single stroke is the same, the linear correlation among columns will be greatly encouraged. However, for a scene character image, there is always some noise that can destroy the linear correlation among text columns. Thus, noise can destroy the low rankness of a character image. So, in order to enhance the linear correlation of a text image, we preprocess the text image with two steps.

The first step is to binarize the text image. Binarization makes a column or a row of a single stroke similar, thus enhancing the linear correlation among the columns and rows of the front view character images. Figure 2 shows how binarization process reduces the rank of a text image. The left image is the binary image after using Otsu’s [8] algorithm on the noisy image. The right image is the graylevel image with random noise. Although the noisy image seems quite like the binary image, their rank values differ significantly. Thus binarization can greatly reduce the rank of text image.

Obviously, there are two options for binarization: one is to have a white character on a black background, and the other is to have a black character on a white background. Although

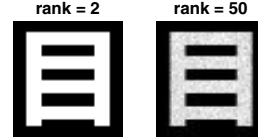


Figure 2. Rank values for the binary image (left, whose rank is 2), and the original graylevel image (right, whose rank is 50).

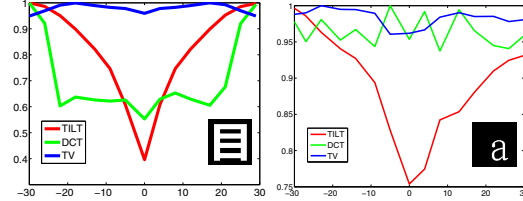


Figure 3. Values of TILT, DCT, and TV objective functions (y-axis) of the input character (shown as the embedded icon) against the rotation angle (x-axis).

the inversion could only affect the rank of the image by one, in practice it could significantly affect the performance of the TILT algorithm. This is because unlike generic low-rank textures, pixels that define a character usually occupy a small portion of the image. So if the background is set as white, the objective function will be dominated by the background and is less sensitive to the pose of the character. In fact, the entire character is likely to be interpreted as the sparse error by the algorithm. We have observed that in practice, the algorithm indeed works much better on white characters.

B. Rectifying Single Characters

After binarizing the text image, we can rectify the distorted character image. The left of Figure 3 shows the values of the objective function of TILT on a typical Chinese character at different rotation angles. This figure shows one good properties of low rank textures. It can be seen that the rank value achieves its lowest point when the character is in its front view position. So minimizing robust low rank objective function helps correct the pose of characters. To illustrate why this is an effective objective function, we also plot the corresponding values of the total variation (TV) and the l_1 norm of the discrete cosine transform (DCT), both of which intend to measure the “sparsity” of the image. As we can see from the plots, the robust low rank objective function is much more sensitive to the pose variation of the character than DCT, whereas TV is not so informative about the pose at all. Although the chosen Chinese character does have many horizontal and vertical strokes (in which case both TV and DCT are expected to perform well), rank value is a better indicator for upright position for those characters. As it turns out, this is the case for *almost all* Chinese characters and English words too, which is the fundamental reason why character rectification is possible!

Similar to the work of TILT [7], we can seek to recover the deformation τ for deformed character by solving the

following optimization problem:

$$\min_{A,E,\tau} \|B(A)\|_* + \lambda\|E\|_1, \quad \text{s.t.} \quad D \circ \tau = A + E, \quad (1)$$

where D is the input distorted character image, A is a low rank matrix, E is a sparse error matrix which is used to remove some small imperfect parts to make the character image as an ‘‘approximately’’ low rank matrix. $B(\cdot)$ is the binarization operator. $\|\cdot\|_*$ and $\|\cdot\|_1$ are the nuclear norm (sum of all singular values) and l_1 -norm (sum of absolute values of all entries) of a matrix, respectively. In our context, τ typically belongs to the group of 2D affine or projective transforms. λ is a parameter for balancing the low rank part and the sparse error part. $D \circ \tau$ means applying transform τ to text image D . Figure 1 shows one example for this model. The upright red rectangle indicates the initial transformed character/text D . It is rectified by a proper transform τ , indicated by the green quadrilateral, as $D \circ \tau$. After rectification, $D \circ \tau$ can be interpreted as the sum of a low rank component A and a sparse one E .

The above problem (Eq. (1)) can be solved from its linearized version (Eq. (2)) using the alternating direction method (ADM) [9]. Interested readers may refer to the original paper on TILT [7] or [9] for more algorithmic details.

$$\min_{A,E,\Delta\tau} \|B(A)\|_* + \lambda\|E\|_1, \quad \text{s.t.} \quad D \circ \tau + J\Delta\tau = A + E, \quad (2)$$

where J is the Jacobian: derivatives of the image $D \circ \tau$ with respect to the transformation parameters, which is actually a 3D tensor.

To be precise, our algorithm works as follows: for an input image, we first conduct a simple foreground and background detection, and then set the intensity value of the foreground character as one and set the background zero. We conduct such a binarization for the image D at every iteration when the image is updated with the incremental transform $\Delta\tau$ found by solving the linearized problem (2).

III. SHORT PHRASE RECTIFICATION

The algorithm proposed above is mainly for rectifying a single character. However, in real applications, a user is often provided with an image with multiple characters on a single line, which can be a short phrase on a street sign or an entry on a restaurant menu. Although it is possible to separate the image into individual characters and apply the above algorithm to each sub-image independently, such a strategy has some disadvantages. First, without rectifying the deformation, it may not be easy to segment the characters accurately. This is a chicken-and-egg problem. Second, there may be characters with few strokes. Rectifying such ‘‘simple’’ characters may not be robust enough and may lead to recognition failures. Third, after independent rectification, it is nontrivial to align all the characters as the computed



生日蛋糕 中国光大 小时营业

Figure 4. **Rectification of Chinese characters individually in an image.** The top row is the input images and the bottom row is the outputs of affine TILT. Notice that individual characters may be rectified with slightly different scales.

transforms may be different from one another. This has been exemplified by Figure 4.

Actually, if all the characters are on the same plane, they should undergo the same affine or projective transform τ . So a better strategy is to rectify all the characters jointly. However, although the images of individual characters may be low-rank, the image of multiple characters on a line may no longer be low-rank anymore *w.r.t. its minimal dimension*. So the previous rank minimization objective function may not work on the joint image as robustly as on an individual character or on a paragraph of texts. To remedy this issue, we propose the following optimization problem for rectifying multiple characters on the same line:

$$\min_{A_i,E,\tau} \sum_{i=1}^n \|A_i\|_* + \|A\|_* + \lambda\|E\|_1, \quad (3)$$

$$\text{s.t.} \quad D \circ \tau = A + E, \quad A = [A_1, \dots, A_n],$$

where A_i stands for the i -th block of A . The above formulation is motivated by the observation that each character (hence sub-image) is of low rank. Although A_i should ideally correspond to a character, it is unnecessary to segment the image accurately. Only a rough estimate of the number of characters, which can be easily derived from the aspect ratio of the specified region, is needed and A_i can be obtained by equally partitioning the region. For convenience, we call problem (3) ‘‘Multi-Component TILT.’’ Accordingly, the original TILT, i.e., problem (1), is called ‘‘Uni-Component TILT.’’ Although in many cases, affine transform is sufficient for the uni-component case, we typically need to use projective transform for the multi-component case, because multiple characters often cover a larger region where perspective distortion may not be negligible.

Similar to the uni-component TILT, multi-component TILT can be solved by linearizing *w.r.t.* the (projective) transform τ and then computing the increment of τ iteratively, leading to the following sub-problem:

$$\min_{A_i,E,\Delta\tau} \sum_{i=1}^n \|A_i\|_* + \|A\|_* + \lambda\|E\|_1, \quad (4)$$

$$\text{s.t.} \quad D \circ \tau + J\Delta\tau = A + E, \quad A = [A_1, \dots, A_n].$$

The readers may refer to [7], [9] for how to derive the detailed optimization algorithm.



Figure 5. Examples of skewed or rotated characters. From left to right: the original image, the rotated image with a rotation angle $\theta = 20$, the skewed image with a skew value $t = 0.3$.

Table I
COMPARISON EXPERIMENTS ON OUR SINGLE-CHINESE-CHARACTER DATASET

Algorithm	blackfont	songti	kaiti	lishu
Uni-TILT mean	0.6068	0.9678	2.1650	1.4746
Hough mean	6.6568	12.9961	24.9260	20.3495
Uni-TILT std	0.8552	2.3857	1.7106	1.0011
Hough std	14.4840	19.1835	23.8655	21.6200

IV. EXPERIMENTAL RESULTS

In this section, we conduct extensive experiments to evaluate the performance of our algorithm in terms of how it corrects large rotation and skew of the single character as well as multiple characters.

A. Experimental Results on Single Characters

Rectifying Chinese Characters: In order to test single character rectification, we build a single-Chinese-character dataset. It contains 2,500 most commonly used Chinese characters in the four most popular standard fonts. The character image is binary-like and the character is brighter than the background. As most existing methods work for a line or a paragraph of texts, the Hough transform [10] becomes baseline for rectifying a single character. We choose Chinese characters because they are rich of regularity such as vertical or horizontal strokes. So the Hough transform may work well on many of them.

We do two groups of experiments. The first one is to rotate the 10,000 characters by $\theta = 20$ degrees, and the second one is to skew the 10,000 characters by $t = 0.3$ ¹. Figure 5 shows examples of skewed or rotated Chinese characters. For the experiment on rotated characters, we record the difference of the angle between the ground truth and the rectified solution. The histograms of the differences for the TILT algorithm and the Hough transform are shown in Table I. We can see that the accuracy of TILT is rather good and is significantly higher than that of Hough. Figure 6 also shows the result of our algorithm in rectifying the skewed input. *As the Hough transform cannot detect the skew direction robustly (and as expected, it works very badly on skewed characters), there is no need to show its results on rectifying skewed characters.*

Rectifying English Characters, Digits, and Real Images: We also compare our method with the Hough transform on clean English characters and digits, and real images. Part of the results are shown in Figure 7. We can see that whenever there are salient horizontal and vertical strokes

¹Here we do not choose θ and t randomly because 20 rotation degrees and 0.3 skew are the working limit of our method. It will be less challenging if the rotation is uniformly random from -20 degrees to 20 degrees or uniformly random from -0.3 to 0.3 skew.

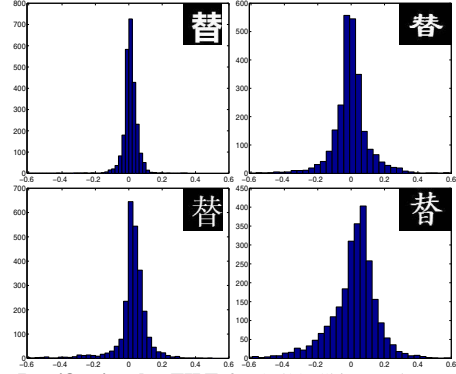


Figure 6. **Rectification by TILT** for 2,500 Chinese characters that are skewed with a skew value $t = 0.3$. The horizontal axis is the difference between the recovered and the ground truth skew values. The vertical axis is the number of characters whose differences in skew values are in the same bin. Top left is for ‘blackfont’. The mean is 0.0292 and the standard deviation is 0.0013. Top right is for ‘lishu’. The mean is 0.0711 and the standard deviation is 0.0079. Bottom left is for ‘songti’. The mean is 0.0643 and the standard deviation is 0.0065. Bottom right is for ‘kaiti’. The mean is 0.1070 and the standard deviation is 0.0105.

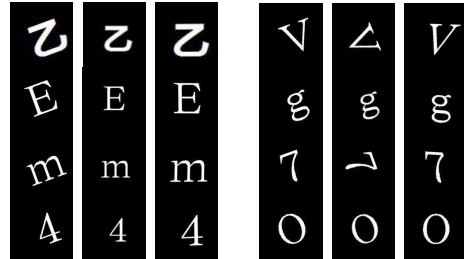


Figure 7. **Comparison between the Hough transform and TILT.** Left: examples that both methods are successful. Right: examples that the Hough transform fails. The first column is the input images. The second and third columns are the results of the Hough transform and TILT, respectively.

in the characters, the Hough transform can work well in rotation. Otherwise, the results of the Hough transform can be quite random. Moreover, the Hough transform cannot rectify skewed characters. In contrast, TILT performs stably and produces nearly perfect rectification results, as long as the characters have certain regularity and global symmetry – no need of dominant horizontal or vertical edges.

B. Experimental Results for Short Phrases

In this section, we test on daily life images which are collected by ourselves. Figure 8 shows part of the results of multi-component TILT on a line of multiple Chinese and English characters. Since the ground truth of image distort is unknown for real images, we use an Optical Character Recognition (OCR) engine to evaluate the rectification result. The recognition performance of current OCR is very sensitive to the deformation in the input characters. That is, it performs well only when the characters are presented in their standard upright position, at which the OCR engines were trained. Most widely used commercial OCR systems can tolerate only very small rotation and skew in the input characters. One of the most popular OCR engine for Chinese character is Hannwang OCR. In our collected dataset, there



Figure 8. Rectification results on images of multiple characters using multi-component TILT. The first, third, and fifth rows are the input images with initial windows (red rectangles) to specify the regions of interest, and the second, fourth, and sixth rows are the outputs of multi-component TILT.

are totally 1,020 distorted characters and the recognition rate for Hanwang OCR is 23.04%. After applying the multi-component TILT algorithm, the recognition rate improves to 56.57%, which is a remarkable improvement.

V. CONCLUSION

In this paper, we have presented an effective method for rectifying Chinese and English characters or words as robust low-rank textures. We have discovered that the objective function of TILT is a good indicator for detecting the upright position of characters. With binarization and inversion, the generic TILT algorithm can be made to work well on almost all Chinese and English characters and words, as well as digits. We also extend the algorithm to handle short phrase, which allows us to rectify images of street signs taken by a mobile phone. As we have shown, the new rectification method can significantly enlarge the working range of conventional OCR systems for deformed input characters.

The current method works well on Chinese characters, English texts, and digits, because their standard fonts are rich of regularity and symmetry and can be interpreted as

(robust) low-rank textures. Our method should work for languages that also have rich regularity. We will test with more languages in the future. Finally, in this paper we only show how the TILT-based method can correct both affine and projective transforms. In principle, our method can be extended to deal with more general classes of nonlinear transforms. We will also explore this possibility in the future.

ACKNOWLEDGMENT

X. Zhang and F. Sun are supported by the National Natural Science Foundation of China (NNSFC) under Grant no. 91120011. Z. Lin is supported by the NNSFC under Grant nos. 61272341, 61231002, and 61121002. Y. Ma is partially supported by the funding of ONR N00014-09-1-0230, NSF CCF 09-64215, NSF IIS 11-16012.

REFERENCES

- [1] J. Liang, D. DeMenthon, and D. Doermann, "Geometric rectification of camera-captured document images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 591–605, 2008.
- [2] M. Pastor, A. Toselli, and E. Vidal, "Projection profile based algorithm for slant removal," *Image Analysis and Understanding (Lecture Notes in Computer Science)*, vol. 3212, pp. 183–190, 2004.
- [3] H. Cao, X. Ding, and C. Liu, "A cylindrical surface model to rectify the bound document image," in *ICCV*, 2003, pp. 228–233.
- [4] C. Singha, N. Bhatia, and A. Kaur, "Hough transform based fast skew detection and accurate skew correction methods," *Pattern Recognition*, vol. 14, no. 12, pp. 3528–3546, 2008.
- [5] B. Yuan and C. L. Tan, "Convex hull based skew estimation," *Pattern Recognition*, vol. 40, no. 2, pp. 456–475, 2007.
- [6] H. Yan, "Skew correction of document images using inter-line cross-correlation," *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 6, pp. 538–543, 1993.
- [7] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma, "TILT: Transform invariant low-rank textures," in *Asian Conference on Computer Vision*, 2010.
- [8] O. Nina, B. S. Morse, and W. A. Barrett, "A recursive otsu thresholding method for scanned document binarization," in *WACV*, 2011, pp. 307–314.
- [9] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," 2009, uIUC Technical Report UILU-ENG-09-2215, arxiv:1009.5055.
- [10] J. Cha, R. H. Cofer, and S. P. Kozaitis, "Extended hough transform for linear feature detection," *Pattern Recognition*, vol. 39, no. 6, pp. 1034–1043, 2006.