Distance Transform Based Active Contour Approach for Document Image Rectification

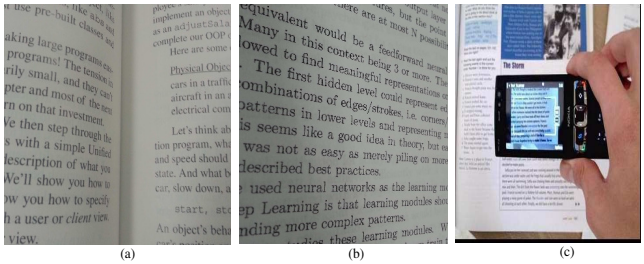
Dhaval Salvi, Kang Zheng, Youjie Zhou, and Song Wang
Department of Computer Science and Engineering
University of South Carolina, Columbia, SC 29208, USA
{salvi,zheng37,zhou42}@email.sc.edu, songwang@cec.sc.edu

Abstract

Digitization of document images using OCR based systems is adversely affected if the image of the document contains distortion (warping). Often, costly and precisely calibrated special hardware such as stereo cameras, laser scanners, etc. are used to infer the 3D model of the distorted image which is used to remove the distortion. Recent methods focus on creating a 3D shape model based on the 2D document image. The performance of these methods is highly dependent on estimating an accurate 2D distortion grid. In the domain of printed document images, the white space between the text lines carries as much information about the 2D distortion as the text lines themselves. Based on this intuitive idea, we build a 2D distortion grid from white space lines, which can be used to rectify a printed document image by a dewarping algorithm. These white space lines are extracted using a propagation technique on the distance transform of the binarized document image, guided by an open active contour algorithm. We compare our proposed method against a state-of-the-art 2D distortion grid construction method and obtain better results. We also present qualitative and quantitative evaluations for the proposed method.

1. Introduction

Optical character recognition (OCR) is an important step in the process of digitizing important documents such as old historic books. OCR research over the last few decades has led to highly accurate digitization of documents. However there is a severe drop in the performance of OCR systems in the presence of distortion (warping) in the scanned/photographed document image as shown in Fig. 1. These systems rely on the document image being planar and having straight horizontal text lines. Therefore, it is critical to remove any distortion that might exist in the document image.



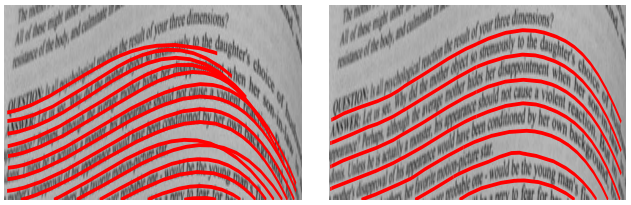
(a) (b) (c)

Figure 1. Examples of distortions: (a) Distortion at book bindings, (b) Perspective projection in camera captured image. (c) One of the applications of proposed method: Mobile document scanner (Image from google).

Previous literature on handling this distortion problem can be categorized in three ways. The first category of related works rely on special hardware, such as stereo cameras [16] [20], laser scanners [12] or structured light devices [2] [3] to solve the distortion problem. These precisely-calibrated systems acquire a 3D shape model of the distorted document, and use the obtained model to rectify the distortion. Although such systems are shown to be highly accurate, the cost and size of such hardware makes them an impractical option for many applications.

The second category of approaches [6, 21, 22, 9, 7, 18, 17, 4] rely on inferring the distortion from the text lines present in the document. These approaches pre-process images using techniques such as binarization, connected component analysis, or character segmentation to estimate the 2D distortion grid. The accuracy of such methods is highly sensitive to the results obtained from the aforementioned preprocessing techniques, and thus may obtain poor results if the underlying preprocessing yeilds unsatisfactory results. Ulges *et al*. [17] use priori layout information and local textline estimation with RAST [1] which relies on connected component analysis. Bukhari *et al*. [4] use Gaussian matched filters to enhance text lines and employ a ridge detection technique to find the center of the text line, after which an active contour model is used to find the top and

978-1-4799-6683-7/15 $31.00 © 2015 IEEE
DOI 10.1109/WACV.2015.106



(a) (b)

Figure 2. An illustration of line tracings. (a) Example of wrongly estimated horizontal text lines as described in [15], (b) White space based line tracing of proposed method.

bottom part of the horizontal text lines.

The third category of methods employ shape-based models to perform the document dewarping. These works can be further divided into methods which: a) have a rigid predetermined shape model, and b) use a deformable shape model to capture more variation among the possible distortion. Methods which use a rigid shape model include Cao *et al*. [6], who propose a model-based method to rectify document image distortion, which makes the assumption that the book surface is a cylindrical surface. Also, Fu *et al*. [8] extend this work by removing the constraint that the camera lens must be strictly parallel to the book surface, and additionally introduce textual information for reconstructing the 3D shape. Meng *et al*. [11] present a metric rectification method which employs a general cylindrical surface to model the page shape, and use the horizontal textline information and vanishing point to sample a grid on this cylindrical surface. Such rigid model based methods are able to rectify a scanned document image but mostly fail to rectify document images captured with a camera, where the distortion does not necessarily follow a rigid model.

Methods which use a deformable shape model, relaxing the fixed-shape constraint, are able to better handle such distortions. Such methods include Liang *et al*. [10], who relax the strong shape assumption by asserting that the document image can be approximated by a developable surface. This assumption is intuitive because document pages can be unrolled without stretching or tearing. Shao *et al*. [13] extend this idea by formulating a locally deformable surface instead of a globally deformable surface. Tan *et al*. [14] propose a shape-from-shading (SFS) method to recover the 3D shape of the document. Tian *et al*. [15] propose a new state-of-the-art 3D rectification framework which utilizes a similarity measure based horizontal line tracing and uses vertical stroke statistics to estimate the vertical direction. As shown in Fig. 2 (a), this method may suffer from inaccurate line tracing when the similarity measure mistakenly associates two patches within the image, thereby affecting the final result.

The proposed method falls in the shape-from-X class of approaches. We humans can easily infer the 2D distortion based on the text lines and the surrounding white space between these text lines, as shown in Fig. 2 (b), independent of the language or fonts present in the document. Based on this intuition, we utilize the white space that exists between the text lines, rather than the actual lines themselves, to estimate the 2D distortion grid. We use this approach because preprocessing methods such as binarization, connected components *etc*. can affect the information extracted from the text lines, whereas the white space surrounding these text lines is less affected by such preprocessing. We propose a distance transform (DT) based line propagation technique guided by an open active contour algorithm, to robustly estimate the horizontal line tracings and build a 2D distortion grid based on the white space.

The key contributions of the proposed work are a) 2D distortion grid estimation that uses the white space between text lines instead of the text lines themselves, and b) leverages the information in the distance transform to intelligently estimate the distance between lines in a propagation approach. After obtaining the 2D distortion grid, we evaluate the proposed method by using the 3D rectification method in [15] to show that the proposed 2D distortion grid results in better dewarping performance. We provide qualitative and quantitative comparison between the proposed work and the state-of-the-art method.

2. Proposed Method

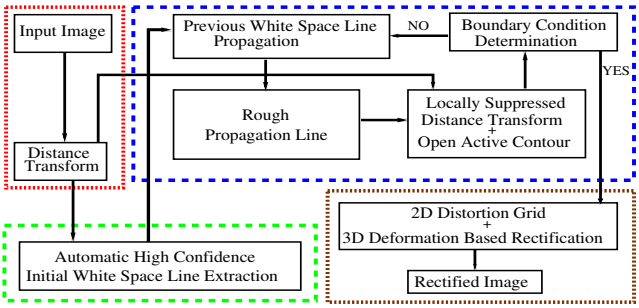


Figure 3. System diagram of the proposed method.

Proposed method can be summarized by the pipeline shown in Fig. 3 and includes the following steps:

1. Compute the distance transform (DT) for given input image as described in Section 2.1, as shown in Fig. 3 with the red dashed box.

2. Extract a high confidence initial white space(WS) line automatically using a bank of Gaussian line filters approach as described in Section 2.2, as shown in Fig. 3 in the green dashed box.

3. Use an iterative WS line propagation process (shown in blue dashed box in Fig. 3) as described in Sec-

tion 2.3 and Section 2.4, to extract all WS lines in the document image and refine them using open active contour algorithm.

4. Construct a 2D distortion grid from the extracted WS lines and use it to rectify the input image using a 3D dewarping algorithm as described in Section 3, as shown in Fig. 3 by the brown dashed box.

2.1. Distance transform (DT)

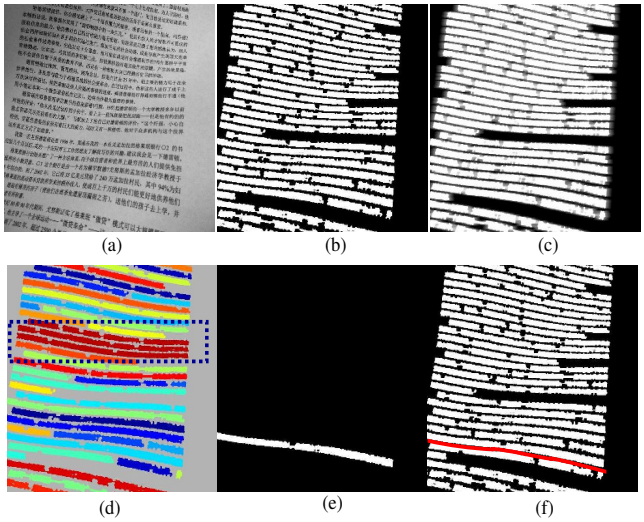
(a) (b) (c)

Figure 4. An illustration of distance transform (DT). (a) Original image, (b) DT of the original image and, (c) Gradient map of the DT.

As discussed above, our key intuition is to use the white space between the text lines in the document image, to obtain a better 2D distortion grid. A typical approach to represent this notion of white space is to use a distance transform (DT) of the binarized document image. The binary image is filtered to remove any noise which might affect the DT. The WS line tracing is then formulated as a process of finding maxima in the Euclidean distance transform $D(\mathbf{x})$ from a binarization of the original image $I(\mathbf{x})$, where $\mathbf{x} = (x, y)$ represents the pixel position. The DT can be visualized as an intensity image, where the intensity at a pixel indicates its distance to the nearest text pixel, as shown in Fig. 4 (b). With this DT, it is more convenient to locate a WS line between any two text lines. This is evident from the gradient map of the DT, as shown in Fig. 4 (c), as the WS line between two text lines tends to fall on the DT maxima, and appears in the gradient map as an edge between the two text lines.

2.2. Automatic extraction of high confidence initial WS line

A document image (as shown in Fig. 5 (a)), usually contains a set of WS lines. In this section, we propose an approach to extract one of these WS lines with high confidence as the initialization L_0 (as shown by the red line in Fig. 5 (f)). With this initial WS line, we can propagate upward and downward iteratively to extract all the WS lines, which will be discussed later in Section 2.3. Specifically, a modified version of the bank of Gaussian line filters algorithm described in [5] is used to extract L_0. First we binarize the DT by a small threshold δ_1, such that pixels with $D(\mathbf{x}) < \delta_1$ result in a binary image as shown in Fig. 5



(a) (b) (c) (d) (e) (f)

Figure 5. An illustration of automatic initial WS line extraction with high confidence. (a) Original text, (b) Binary image acquired by thresholding DT, (c) Output of bank of Gaussian line filters, (d) Connected components result, blue dashed box highlights merging of textual lines in a highly warped region, (e) Largest width connected component, and (f) High confidence initial WS line L_0 shown in red.

(b). Next the algorithm described in [5] is used to obtain a smooth image as shown in Fig. 5 (c). A high threshold δ_2 is used to threshold and binarize the image shown in Fig. 5 (c) and connected components analysis is performed on this binarized image, resulting in blobs as shown in Fig. 5 (d). The amount of document warping might vary across the page, and may result in the merging of multiple textual lines in a highly warped region into a bigger component after using the algorithm from [5], as shown in Fig. 5 (d) by the red component inside the blue dashed box. To address this problem, an average height is calculated using all the connected components and any components with height greater than this average height are discarded. A component with the largest width as shown in Fig. 5 (e) is selected from the remaining components to estimate L_0. This width is estimated from the length of the major axis of an ellipse fitted to the component. The pixels at the bottom edge of this connected component are used as a rough estimation $\hat{L}_0$, which is refined into initial WS line L_0 using the open active contour algorithm described further in Section 2.4.

2.3. Iterative WS line propagation

Once the high confidence initial WS line L_0 described in Section 2.2 is obtained, it is used to extract the remaining WS lines in the document image using an iterative propagation process, performed in upward and downward directions relative to L_0. Let's consider the downward propagation that extracts the WS line L_1 just below L_0. Given the WS line L_0 as shown by the red curve in Fig. 6 (a),

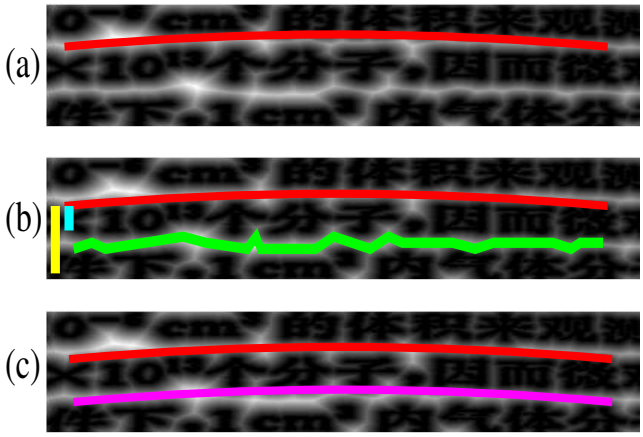


Figure 6. An illustration of WS line propagation. (a) Initial WS line L_0 shown by red curve on top of DT, (b) WS line propagation to obtain a rough propagation line $\hat{L}_1$ shown in green curve on top of the DT, with displacement bounds t_1 and t_2 shown by cyan and yellow vertical lines on the left of (b) respectively, and (c) Extracted WS line L_1 shown in magenta on top of DT.

each pixel $\mathbf{x} = (x, y)$ on this line is displaced downward such that it is located on a new local DT maxima within specific bounds from L_0. These displaced pixels form the rough propagation line $\hat{L}_1$ as shown by the green curve in Fig. 6 (b). Specifically, pixel $(x, y) \in L_0$ is displaced to $(x, y^*) \in \hat{L}_1$ where:

$$y^* = y - \arg \max_{\delta_y \in [t_1, t_2]} D\left(x, y - \delta_y\right)$$

t_1 and t_2 are the thresholds that bound the displacement. These thresholds are computed from L_0 by averaging $D(\mathbf{x})$ for each pixel $\mathbf{x}$ along this line as:

$$\bar{D}(L_0) = \frac{1}{|L_0|} \sum_{\mathbf{x}_i \in L_0} D(\mathbf{x}_i)$$

We set $t_1 = 0.6 \cdot \bar{D}(L_0)$ and $t_2 = 1.2 \cdot \bar{D}(L_0)$, as illustrated by cyan and yellow vertical lines in Fig. 6 (b). This rough propagation line $\hat{L}_1$ is then refined to the WS line L_1 using the open active contour based refinement process described in Section 2.4. With L_1, we can use the same technique to propagate downward and get the next WS line L_2. This process can be repeated to extract all the WS lines below L_0. Similarly, we can use this propagation technique upwards to extract all the WS lines above L_0. The WS line propagation process is stopped at the top and bottom of the image boundaries.

2.4. Open Active Contour based Line Refinement

Active contours, also known as "*Snakes*," are energy minimizing deformable splines and widely used for computer vision tasks, such as segmentation. Snakes are generally defined by an internal elastic energy term which defines the bending enery of the snake and an external edge-based energy term, which guides them towards the desired contour. For each of the rough propagated line $\hat{L}_i$ described in Section 2.3, as shown in Fig. 6 (b) by the green curve, we apply an open active contour algorithm to remove any small local noise and obtain the smooth WS line L_i as shown in Fig. 6(c) by the magenta curve. We use a Gradient Vector

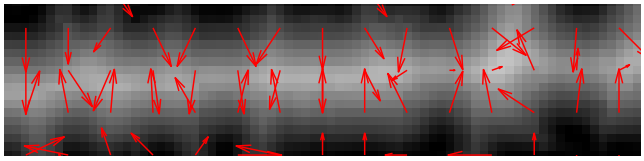


Figure 7. An illustration of a Gradient Vector Flow (GVF) field based on the DT.

Flow (GVF) based open snakes algorithm to obtain the WS line refinement. The DT described in Section 2.1 is used for defining the external energy force, to guide the open snake towards the DT maxima. As illustrated in Fig. 7, the gradient vector flow field is directed towards the DT maxima. The external edge energy term is formulated using the DT as follows:

$$E_{edge} = -\left|\nabla D\left(\mathbf{x}\right)\right|^2$$

We still have to address one special case when using the DT

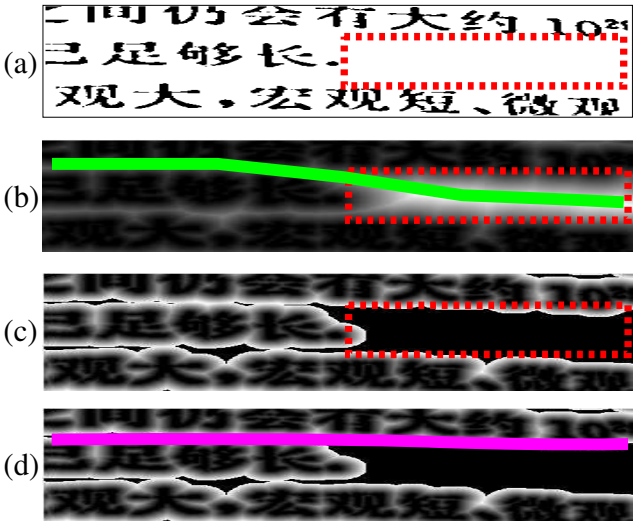


Figure 8. An illustration of a special case using the DT. (a) Document image containing a short text line, red dashed box highlights the extra white space at the end of the short line. (b) Open snake attracted towards artificial DT maxima, as shown by green curve on top of DT inside the red dashed box. (c) Supressed DT shown in the red dashed box. (d) Open snake fitting the suppressed DT maxima, resulting in a refined WS line.

and open active contour to find the WS lines, as illustrated in Fig. 8 (a) and (b). A short text line will cause extra white space at the end of the text line, as shown by the red dashed

box in Fig. 8 (a). This introduces an artifical DT maxima that will be misinterpreted by the open snake-based line estimation as the location for the WS line, as illustrated inside the red box in Fig. 8 (b). Such an error will introduce an undesired deformation in the constructed 2D distortion grid. To address this problem, a DT suppression is performed near the rough WS propagation line $\hat{L}_i$, before using the open snake to obtain the WS line L_i. To perform this local DT suppression for a given rough WS propagation line $\hat{L}_i$, we suppress the DT as follows:

$$D(\mathbf{x}) = \begin{cases} 0 & \texttt{if } D(\mathbf{x}) > \bar{D}(L_{i-1}) + \epsilon \\ D(\mathbf{x}) & \texttt{otherwise} \end{cases}$$

where L_{i-1} is the previous WS line and ϵ is a small integer value, which is set empirically to 5 in our experiments. This operation attenuates the artifical DT maxima caused by the extra white space, and we get a suppressed DT as illustrated in Fig. 8 (c). The open snake now converges to the suppressed DT maxima and generates the desired refined WS line L_i as illustrated in Fig. 8 (d).

3. Text Orientation Estimation and 3D Dewarping

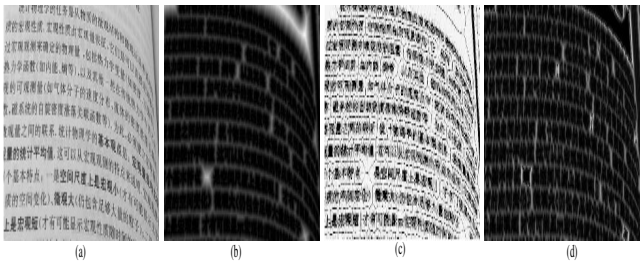


Figure 9. An illustration of vertical direction estimation based on the DT. (a) The original image, (b) DT of original image, (c) Gradient map of the DT and (d) Gabor filter map of the DT.

We use the 3D dewarping algorithm from [15] for image rectification, which requires not only horizontal (WS) lines, but also vertical lines to form a complete 2D distortion grid. As illustrated in Fig. 9(c), we can see from the gradient map of the DT , the vertical direction can also be infered from the white space between the individual characters irrespective of the language used. To infer these local dominant vertical directions we first apply a gabor filter bank to enhance the vertical information contained in the DT. Specifically, we apply a set of vertical orientation biased filters to the DT which is shown in Fig. 9(b), which enhance the vertical direction information contained in the DT. We obtain the gabor filtered map as shown in Fig. 9(d). After robustly estimating the WS lines as described in Section 2, we use a similar optimization as discussed in [15] to find the dominant vertical directions in overlapping local regions in this gabor filter map. A 2D distortion grid is created based on these infered vertical directions and the previously extracted WS lines.

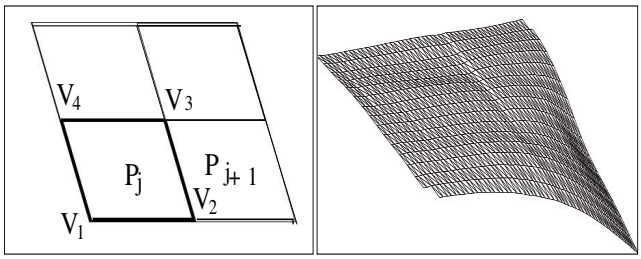


Figure 10. An illustration of 3D deformation estimation as described in [15]. (a) Each 2D distortion cell is a parallelogram in 3D space. (b) 3D parallelogram mesh.

Next, a 3D deformation of the distorted document is estimated from the 2D distortion grid. It is assumed that the camera projection is perspective, and each cell of the 2D distortion grid is a parallelogram in 3D space. A 3D parallelogram mesh is constructed from the 2D distortion grid. Let $\mathrm{V}_i = (X_i, Y_i, Z_i) = (x_i Z_i, y_i Z_i, Z_i)$ denote the 3D location of i-th grid vertex, where (x_i, y_i) is its 2D coordinates. As shown in Fig. 10 (a), The four vertices $V_{1:4}$ form a parallelogram P_j if $\Delta(P_j) \equiv \mathrm{V}_1 + V_3 - V_2 - V_4 = 0$. The following optimization is performed using Singular Value Decomposition to obtain the globally optimal solution for location of each Vertex V_i , as described in [15]:

$$Q\left(\{\mathrm{V}_i\}_{i=1}^n\right) = \sum_{j=1}^{N_p} \|\Delta(P_j)\|^2 + \alpha \sum_{i=1}^{n} \left(X_i - x_i Z_i\right)^2 + \left(Y_i - y_i Z_i\right)^2 \quad (1)$$

We use this formulation to take the 2D distortion grid created using our WS line estimation described in Section 2, to create a 3D deformation and rectify the associated document image.

4. Experiments

In our experiments, we evaluate three methods, namely:

1. **No-dewarp**: In this method, we use the original image without performing any dewarping.
2. **TL-dewarp**: The text line based dewarping described in [15].
3. **Proposed-dewarp**: Proposed WS line based dewarping described in this paper.

For TL-dewarp, we use the code and parameters as used in [15]. We choose a freely available OCR engine (onlineocr.net) to perform OCR on the dewarped images and use

the OCR results for quantitative evaluation of each method disccused above. Three evaluation datasets are collected as follows:

Dataset 1: This dataset consists of 15 English document images from [15].

Dataset 2: This dataset consists of 15 English document images that we collected ourselves.

Dataset 3: This dataset contains 20 international document images, consisting of images in Hindi and Chinese Languages. It contains equal number of documents from [15] and images we captured overselves. This dataset is used only for qualitative evaluation since the OCR used does not work on international document images.

Evaluation criteria: Dataset 1 and Dataset 2 are used for quantitative evaluations using all three methods, for which we use the evaluation method described in [19]. First the ground truth text is created for each image by manually typing the text contained in the image. This evaluation method then uses a text alignment scheme by which it aligns the OCR result for a given method against the ground truth text on the word/character level. Once the text's are aligned, word and character recognition accuracies are estimated. The OCR accuracy metric is defined as $\frac{w}{t}$, where w is the total number of matching words/characters in the alignment and t is the total number of words/characters in the ground truth. We tabulate the results of running No-dewarp, TL-dewarp and Proposed-dewarp on Dataset 1 and Dataset 2 in Table 1. In this table, we report the word level and character level OCR accuracy, averaged over the respective entire datasets. We can see that the Proposed-dewarp method performs the best amongst the three methods discussed. This table also demonstrates the effectiveness of using a dewarping algorithm in general to improve the OCR accuracy of a distorted original image. We perform an additional simple experiment using an extra spell-check step. A word processor software (Microsoft Word) is used to spell-check and correct the OCR results. These spell-checked OCR results for No-dewarp, TL-dewarp and Proposed-dewarp are then evaluated as previously discussed using method described in [19]. We tabulate these results in Table 2. Above experiment demonstrates that even after using an OCR with additional spell-check, Proposed-dewarp method still outperforms No-dewarp and TL-dewarp.

	Dataset 1 (word)	Dataset 1 (character)	Dataset 2 (word)	Dataset 2 (character)
No-dewarp	0.4151	0.6029	0.4398	0.6141
TL-dewarp [15]	0.5460	0.6359	0.5437	0.6406
Prop.-dewarp	**0.6558**	**0.7604**	**0.7001**	**0.8109**

Table 1. Quantitative evaluation results for No-dewarp, TL-dewarp [15] and Proposed-dewarp. For each image in Dataset 1 and Dataset 2, we compute the OCR word and character accuracy as described in [19] and average it over the number of images in the respective datasets.

	Dataset 1 (word)	Dataset 1 (character)	Dataset 2 (word)	Dataset 2 (character)
No-dewarp	0.4292	0.6071	0.4486	0.6192
TL-dewarp [15]	0.5585	0.6390	0.5532	0.6427
Prop.-dewarp	**0.6742**	**0.7659**	**0.7201**	**0.8190**

Table 2. Quantitative evaluation results for No-dewarp, TL-dewarp [15] and Proposed-dewarp, using additional spell check (MS Word is used for spell-check). OCR word and character accuracy is calculated for each image in Dataset 1 and Dataset 2 as described in [19], and averaged over the number of images in the respective datasets.

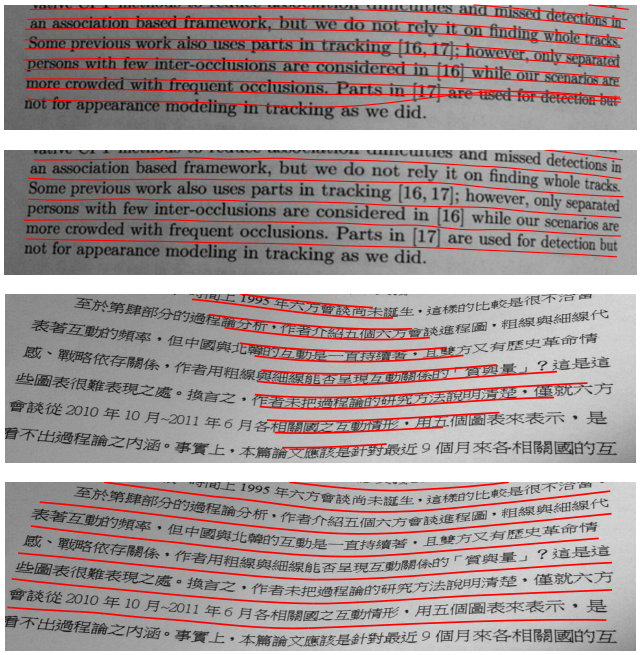


Figure 11. An illustration of failure cases of [15]. First and third row show the failure case for TL-dewarp. Second and fourth row show the result for Proposed-dewarp for the same set of images. First and second row images are from Dataset 2, third and fourth row images from Dataset 3.

Fig. 11 shows some qualitative results where the TL-dewarp method's text line estimation either fails to properly trace the 2D distortion grid lines, or fails to trace enough lines for the 2D distortion grid to cover the image. As mentioned in [15], we varied the step size parameter in a range of 1 to 5, and selected the best possible result. In comparison, Proposed-dewarp robustly estimates the horizontal WS lines for the 2D distortion grid on the same set of images. We provide more qualitative results in Fig. 12 for some select images from all three datasets, where we show the original image plotted with text lines from TL-dewarp, de-

warping result for TL-dewarp, original image plotted with WS lines for Proposed-dewarp, and dewarping result for Proposed-dewarp. We can see from these qualitative results that the Proposed-dewarp is able to locate more meaningful horizontal 2D distortion grid lines than TL-dewarp, and gives a better dewarping result.

Bank of Gaussian line filters parameters: For the bank of Gaussian line filters described in 2.3, we empirically selected range of values for $\theta = \pm 10^{\circ}$ and l in range of 5 to 200.

Snake parameters: For the GVF snakes in our experiments, we used $\alpha = 0.5$, $\beta = 1$ and $\tau = 0.5$ for all the snakes with the number of iterations fixed at 60.

Performance: The code for proposed method is implemented in MATLAB and is not optimized for achieving best runtime performance. Average run time for Proposed-dewarp is similar to TL-dewarp [15] and both take roughly 2-3 minutes for an image of 2592×1936 pixel resolution. All experiments are performed on a quad-core Intel 3.0 Ghz machine with 8G memory.

5. Conclusion and Future Work

In this paper a novel 2D distortion grid estimation method was presented to rectify distorted images based on white space lines that are present between text lines. A distance transform-based line propagation approach is proposed to obtain a robust estimation of the white space lines. These white space lines are then used to build a 2D distortion grid, which is used in a 3D rectification algorithm to dewarp a document image. We demonstrate the robustness and accuracy of our method by providing qualitative and quantitative evaluations, and compare with a state-of-the-art method, achieving better results. In future work, we will extend our framework to handle tables and figures which are embedded in document images.

Acknowledgments This work was supported by NEH HK-50032-12, AFOSR FA9550-11-1-0327, and NSF IIS-1017199.

References

[1] T. M. Breuel. Robust least square baseline finding using a branch and bound algorithm. In *in Document Recognition and Retrieval VIII, SPIE*, 2002.

[2] M. Brown and W. Seales. Document restoration using 3d shape: a general deskewing algorithm for arbitrarily warped documents. In *ICCV*, pages 367–374, 2001.

[3] M. Brown and W. Seales. Image restoration of arbitrarily warped documents. *TPAMI*, 26:1295–1306, 2004.

[4] S. Bukhari, F. Shafait, and T. Breuel. Dewarping of document images using coupled-snakes. In *Proceedings of the third International workshop on camera-based document*, pages 34–41, 2009.

[5] S. S. Bukhari, F. Shafait, and T. M. Breuel. Text-line extraction using a convolution of isotropic gaussian filter with a set of line filters. In *ICDAR*, pages 579–583, 2011.

[6] H. Cao, X. Ding, and C. Liu. Rectifying the bound document image captured by the camera: a model based approach. In *ICDAR*, pages 71–75, 2003.

[7] H. Ezaki, S. Uchida, A. Asano, and H. Sakoe. Dewarping of document image by global optimization. In *ICDAR*, pages 302–306, 2005.

[8] B. Fu, M. Wu, R. Li, W. Li, Z. Xu, and C. Yang. A model-based book dewarping method using text line detection. In *Proceedings of Second International Workshop on Camera Based Document Analysis and Recognition*, pages 63–70, 2007.

[9] H. I. Koo and N. I. Cho. State estimation in a document image and its application in text block identification and text line extraction. In *ECCV*, pages 421–434, 2010.

[10] J. Liang, D. DeMenthon, and D. Doermann. Geometric rectification of camera-captured document images. *TPAMI*, 30:591–605, 2008.

[11] G. Meng, C. Pan, S. Xiang, and J. Duan. Metric rectification of curved document images. *TPAMI*, pages 707–722, 2012.

[12] M. Pilu. Deskewing perspectively distorted documents: An approach based on perceptual organization. *HP White Paper*, 2001.

[13] Y. Shao, X. Liu, X. Qin, Y. Xu, and H. Bao. Locally developable constraint for document surface reconstruction. In *ICDAR*, pages 226–230, 2009.

[14] C. L. Tan, L. Zhang, Z. Zhang, and T. Xia. Restoring warped document images through 3d shape modeling. *TPAMI*, pages 195–208, 2006.

[15] Y. Tian and S. Narasimhan. Rectification and 3d reconstruction of curved document images. In *CVPR*, pages 377–384, 2011.

[16] A. Ulges, C. H. Lampert, and T. Breuel. Document capture using stereo vision. In *Proceedings of the ACM symposium on Document engineering*, pages 198–200, 2004.

[17] A. Ulges, C. H. Lampert, and T. M. Breuel. Document image dewarping using robust estimation of curled text lines. In *ICDAR*, pages 1001–1005. IEEE Computer Society, 2005.

[18] C. Wu and G. Agam. Document image de-warping for text/graphics recognition. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 348–357, 2002.

[19] I. Z. Yalniz and R. Manmatha. A fast alignment scheme for automatic ocr evaluation of books. In *ICDAR*, pages 754–758, 2011.

[20] A. Yamashita, A. Kawarago, T. Kaneko, and K. Miura. Shape reconstruction and image restoration for non-flat surfaces of documents with a stereo vision system. In *ICPR*, pages 482–485, 2004.

[21] A. Zandifar. Unwarping scanned image of japanese/english documents. In *Proceedings of the 14th International Conference on Image Analysis and Processing*, pages 129–136. IEEE Computer Society, 2007.

[22] L. Zhang and C. Tan. Warped image restoration with applications to digital libraries. In *ICDAR*, pages 192–196, 2005.

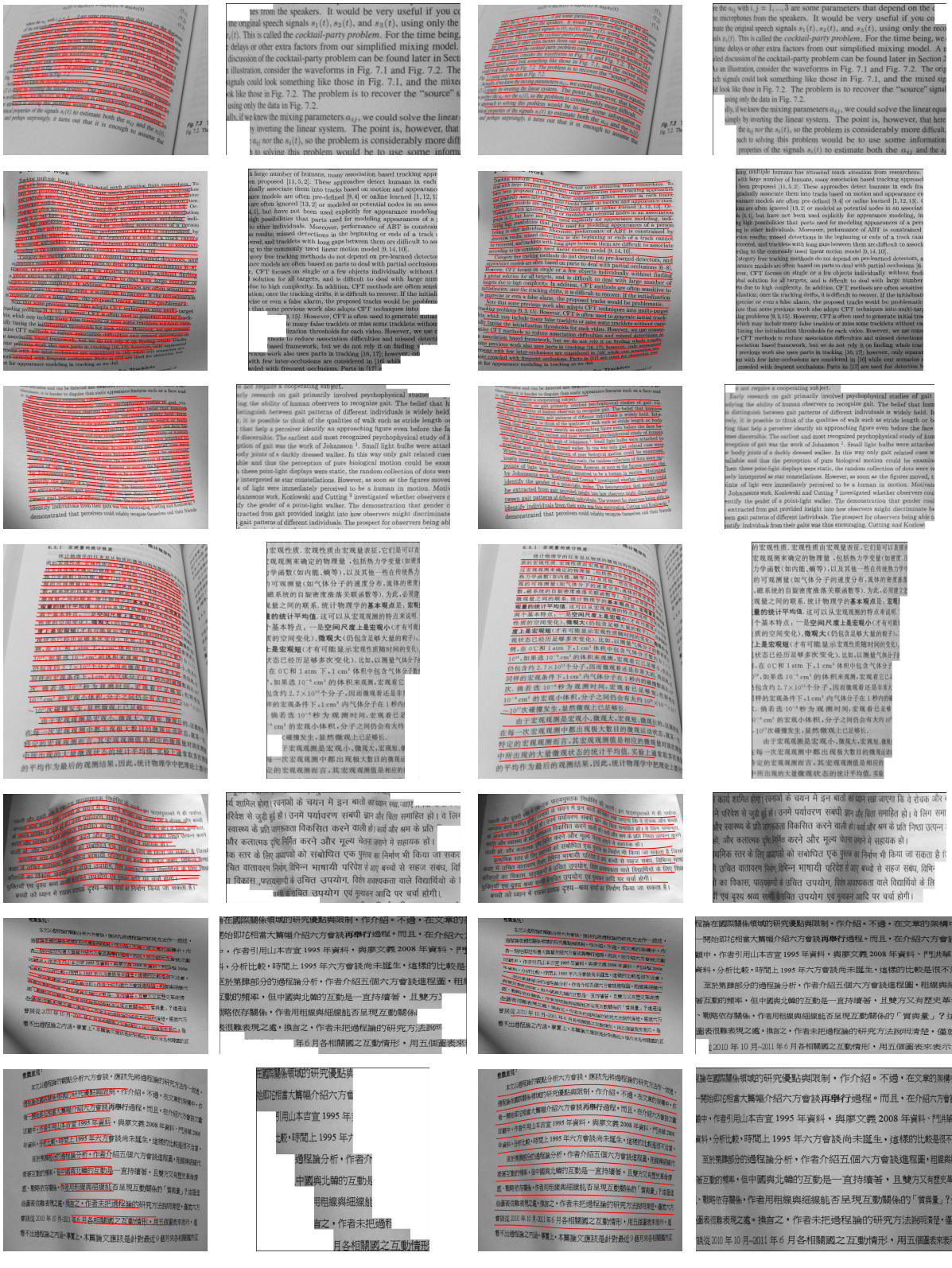


Figure 12. An illustration of qualitative results for TL-dewarp and Proposed-dewarp. The original distorted images plotted with TL-dewarp text lines are shown in the first column. Rectification results for TL-dewarp are shown in the second column. The original distorted images plotted with Proposed-dewarp WS lines are shown in the third column. Rectification results for Proposed-dewarp are shown in the fourth column. Images in row (a) are from Dataset 1, in row (b) and (c) are from Dataset 2, and in row (d)-(g) are from Dataset 3.