Revisiting Document Image Dewarping by Grid Regularization

Xiangwei Jiang[1,2]*, Rujiao Long[3]*, Nan Xue[1], Zhibo Yang[3], Cong Yao[3], Gui-Song Xia[1,2†]

[1]School of Computer Science, Wuhan University, Wuhan, China
[2]LIESMARS, Wuhan University, Wuhan, China   [3]Alibaba-Group, Hangzhou, China

Abstract

This paper addresses the problem of document image dewarping, which aims at eliminating the geometric distortion in document images for document digitization. Instead of designing a better neural network to approximate the optical flow fields between the inputs and outputs, we pursue the best readability by taking the text lines and the document boundaries into account from a constrained optimization perspective. Specifically, our proposed method first learns the boundary points and the pixels in the text lines and then follows the most simple observation that the boundaries and text lines in both horizontal and vertical directions should be kept after dewarping to introduce a novel grid regularization scheme. To obtain the final forward mapping for dewarping, we solve an optimization problem with our proposed grid regularization. The experiments comprehensively demonstrate that our proposed approach outperforms the prior arts by large margins in terms of readability (with the metrics of Character Errors Rate and the Edit Distance) while maintaining the best image quality on the publicly-available DocUNet benchmark.

1. Introduction

The technologies of document digitization (DocDig) have largely facilitated our daily lives by transferring the information written in paper sheets from the physical world to electronic devices. Since the paper sheets are thin, fragile and easily-deformed, it is usually required to carefully capture document images of paper sheets with the scanners to avoid unexpected deformation of papers for digitization. Such a pipeline works well to some extent, however, it quickly loses efficiency when we are using handheld mobile devices, *e.g.*, smartphones, for fast, painless but accurate document digitization. Therefore, the communities of computer vision and document analysis have been making efforts on getting rid of the restriction of DocDig by studying the problem of document image dewarping.

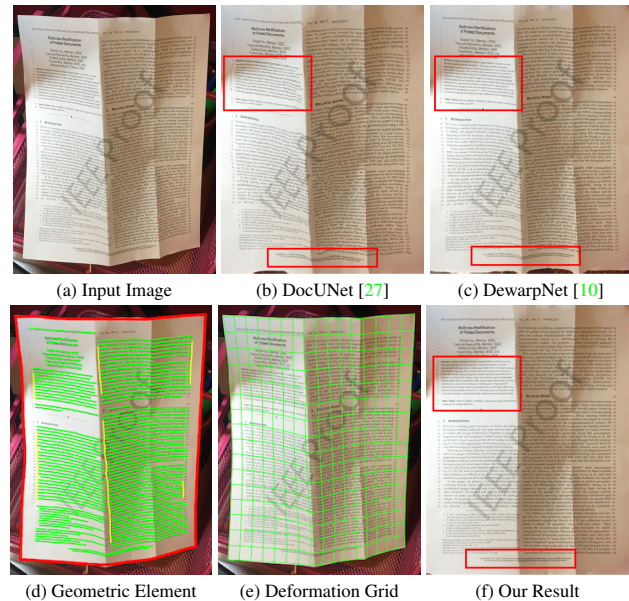There is a rich history for the problem of document im-

Figure 1. Our method leverages the geometric information of text lines and document boundaries to generate a deformation grid which effectively reduces the error rate of character recognition.

age dewarping. In the early pioneering works [2,31,35], this problem was formulated as surface reconstruction from different imaging configurations including multi-view images, binocular cameras as well as structured-lightning depth sensors, achieving accurate results in the lab environment while remaining issues for practical usage. Subsequently, the prior knowledge of paper-sheets in single-view images was extensively studied by detecting the boundaries [5], text lines [17, 18], structured lights [31], and so on [37] in the pre-deep-learning era. As those approaches depend on the detected prior knowledge, they are limited by the detection quality of those low-level visual cues, thus posing an issue of accuracy for the dewarping.

With the resurgence of deep learning in recent years, convolutional neural networks (ConvNets) have also been used for dewarping document image by learning the deformation field between the distorted input images and the expected flattened ones [10,11,21,27,28,39] under various supervision signals such as the synthetic data annotations [27],

---

*Equal Contribution
†Correspondence Author

Forward Mapping (UV)
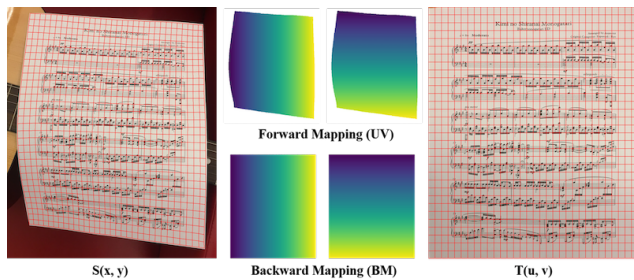
S(x, y)   **Backward Mapping (BM)**   **T(u, v)**

Figure 2. $S(x, y) = T(u, v)$: The equal sign means that pixel values of the corresponding position in different images are equal. Grids belonging to different images correspond to each other. $S \rightarrow T$ is backward mapping, while $T \rightarrow S$ is forward mapping.

the latent 3D shapes of documents [10], high-level semantics [28], and so on [11, 39]. These deep-learning based methods define the document image dewarping problem as a task of learning a 2-dimensional deformation field, which can move pixels from the original image S (Source) to the geometrically rectified image T (Target) as shown in Fig. 2. From the perspective of image quality, those deep-learning approaches significantly improved the dewarping accuracy. However, due to the low-frequency characteristic of neural networks, those approaches remain an issue of readability in the text regions of the output images.

In this paper, we revisit the deep-learning based approaches for document image dewarping and address the issue of readability by grid regularization. Our proposed approach takes advantage of the inherent pattern of forward 2-dimensional deformation field[1] and uses deep learning to detect some geometric information of document images, such as boundaries and text lines, to complete the task of dewarping from a single document image. More precisely, we first obtain boundary points and text lines through key-point detection and semantic segmentation respectively. Then, we estimate the vertical boundary of the text lines using their geometric properties. After that, this geometric information are discretized into 2-dimensional deformation field constraints, with which image deformation energy is minimized by integrating a grid regularization term. Finally, the rectified image is reconstructed, which conforms to our geometric prior characteristics (Fig. 1).

Our contributions in this paper are as follows:

- We present a brand-new framework for document image rectification. The framework employs document boundaries and text lines to construct deformation field with a grid regularization term which is based on the traditional deformation model.

- We systematically analyze the relative merits of reg-

[1]Our work relies on the assumption in the most common situations that the text lines and document boundaries are either horizontal or vertical in the rectified images.

ular terms used in image rectification approaches and come up with a novel grid generation method that better optimizes the quality of the deformation gird.

- The proposed approach achieves state-of-the-art performance on the DocUNet benchmark, demonstrating the effectiveness of the grid regularization term.

2. Related Works

Parametric model based approaches. consider a document image as a simple parametric surface and estimate the corresponding parameters by relying on the detected geometric features in the image. Cylindrical model [6,8,19,23], coons mesh [13] and more complicated developable surface [3,16,22] have been always chosen as the base models. The involved geometric features include 2D curves such as text lines [26,34], document boundary [7], as well as 3D curves obtained by structured light cameras [2,31]. Recently, Kim and Kil *et al.* [17] [18] combine parameter estimation with the camera model that has attracted much attention. While with good performance on document images with simple geometric distortions, these parametric models are too easy to tackle complex geometric distortions appearing in real-world applications.

3D reconstruction based methods. often have two stages: estimating the 3D shape of the document image and predicting a 2D deformation field. Many ways can be used to obtain a 3D point-cloud surface [42] such as with depth camera [37,43], binocular camera, laser scanner [43], multi view images [42] and even with text lines [35]. Subsequently, a 3D surface can be reconstructed from the point clouds and the corresponding 2D deformation field [4] is then recovered by retrieving the surface parameters [29,30]. Obviously, the calculation of the 2D deformation field in this kind of methods heavily depends on the estimation accuracy of the 3D surfaces, *e.g.*, the continuity of the 3D surface and the accuracy of bending. In real-world applications, these methods often quickly lose their efficiency when the lighting conditions and backgrounds are not well-designed.

Deep-learning based approaches. on document image dewarping can be dated back to the work in [27], which approached the problem by learning the 2D deformation field with a stacked U-Net. According to the paradigm of 3D reconstruction, Das *et al.* [10] add 3D shape information into the network to get a better estimation of the deformation field. While Markovitz *et al.* [28] use the constraint that the text line is perpendicular to the text line boundary in 3D shape and texture mapping estimation of DewarpNet [10] to predict the deformation field. Xie *et al.* [39] improve forward optical flow estimation by using the similarity of adjacent optical flow. More recently [40], they further proposed to use Encode structure to estimate fewer points with

the Laplace grid constraint and obtain a better deformation field by traditional interpolation algorithm. Li *et al.* [21] predict a series of small parts of the accurate deformation field by dividing the deformation field and get global result by integral constraints. Das *et al.* [11] integrate this thought into DewarpNet [10], resulting in an end-to-end model. Although these approaches [14, 25, 32] are designed to rectify the whole image, they tend to ignore the detailed text content which greatly affects the readability of the rectified document.

3. A Revisit to Document Image Dewarping

Let $S : \Omega_S \mapsto \mathbb{R}^3$ be a RGB document image suffered from some geometric distortions on the image domain $\Omega_S \subset \mathbb{R}^2$ (*i.e.*, image grid), the ultimate goal of document image dewarping is to pursue a geometric transformation

$$\phi : \Omega_S \mapsto \Omega_T$$

for reconstructing a new RGB image $T : \Omega_T \mapsto \mathbb{R}^3$ such that there is no geometric distortion on the image domain $\Omega_T \subset \mathbb{R}^2$, meaning that Ω_T is flattened.

As being reviewed in Section 2, traditional methods are apt to recover such geometric transformation ϕ either by relying on some paramedic models with geometric constraints, or by using to calculate the 2D deformation field with the help of 3D surface information reconstructed from the document images. While being simple, these methods often lose their efficiency when the geometric distortions appeared in the source image S are complex, *e.g.* the cases discussion in this paper.

Instead of estimating ϕ from a single document image, and benefiting from the strong capability of neural networks to approximate nonlinear functions, deep-learning based methods propose to learn such geometric transformation ϕ from a large set of annotated document images with their corresponding rectified version being given.

In what follows, we revisit the problem of document image dewarping, including both traditional methods and deep learning based ones, from the point view of grid regularization. First of all, we recall the traditional interpolation model [36], an elementary procedure that is widely used in DocDig.

3.1. Transfinite Interpolation

Given the four boundaries of a document image, the transfinite interpolation model [36] can dewarp the document image by using parametric curves.

Denote $\mathbf{c}_1(t), \mathbf{c}_2(t), \mathbf{c}_3(t), \mathbf{c}_4(t)$ as the four curves of boundaries parameterized by t, and $\mathbf{c}(u, v)$ being the grid interpolation from the distorted domain Ω_S to the dewarping Ω_T with u, v are the surface parameters, then the trans-

Table 1. Deep learning methods concerning regularization term.

Method	Regularization Term	MS-SSIM ↑	LD ↓
DocUNet [27]	None	0.4389	10.90
DewarpNet [10]	Checkerboard Reconstruction	0.4692	8.98
Flow Estimation [39]	Similarity of Adjacent Flow	0.4361	**8.50**
Xie *et al.* [40]	Laplacian Mesh	0.4769	9.03
Li *et al.* [21]	Image Splitting	-	-
Das *et al.* [11]	Image Splitting and End-to-End	**0.4879**	9.23

Table 2. Traditional Interpolation with Boundary: The B in parentheses represents the boundary of the deformation field. We adopt DocUNet [27] and DewarpNet [10] pretraining model to obtain the results in the table. At the same time, we utilize traditional interpolation algorithms (TFI, TPS) to obtain other results by using the boundary points of the deformation field from deep models.

Method	CER(std) ↓	ED ↓	MS-SSIM ↑	LD ↓
DocUNet [27]	0.4872 (0.182)	2051.84	**0.4332**	12.59
DocUNet(B)+TFI	0.4537 (0.170)	1852.66	0.4240	12.85
DocUNet(B)+TPS	**0.3861 (0.177)**	**1613.36**	0.4133	**11.92**
DewarpNet [10]	**0.3097 (0.193)**	**1360.51**	0.4340	10.08
DewarpNet(B) +TFI	0.3546 (0.203)	1488.25	**0.4374**	10.02
DewarpNet(B) +TPS	0.3349 (0.194)	1488.00	0.4301	**9.70**

finite interpolation reads,

$$\mathbf{c}(u,v) = (1-u, u) \begin{pmatrix} \mathbf{c}_4(v) \\ \mathbf{c}_2(v) \end{pmatrix} + (1-v, v) \begin{pmatrix} \mathbf{c}_1(u) \\ \mathbf{c}_3(u) \end{pmatrix}$$
$$- (1-u, u) \begin{pmatrix} \mathbf{c}_1(0) & \mathbf{c}_2(0) \\ \mathbf{c}_3(1) & \mathbf{c}_4(1) \end{pmatrix} \begin{pmatrix} 1-v \\ v \end{pmatrix}. \tag{1}$$

The first two terms correspond to ruled surfaces and the last is a correction term which is a projection transformation. Ideally, the output grid points in Ω_T are uniform.

3.2. Significance of Uniformly Deformation Grid

Actually, there are strong geometric priors for document image dewarping. For instance, the pixel coordinates in the rectified image should be uniformly distributed on the grid. However, the deep-learning based methods heavily depend on the representation ability of the deep network and lack an efficient way to integrate such geometric constraints, which often results in a nonuniform output grid.

Table 1 summarizes the methods that integrate grid regularization into the document image dewarping problem since DocUNet [27]. One can see that the use of grid regularization has brought significant improvements to the problem in various methods. This motivates us to define a simple task to verify the validity of the regularization term.

We use DocUNet [27] and DewarpNet [10] as baselines and compare two traditional interpolation algorithms, *i.e.*, TransFinite Interpolation (TFI) [15] and Thin Plate Spline Interpolation (TPS) [1]. The results are reported in Table 2. For DocUNet [27], which has no regularization constraint, a simple grid regularization method can bring significant improvement, saying that the *Character Error Rate* (CER) is decreased by 10.1%. But for DewarpNet [10], which has al-

ready been constrained by grid regularization, it is difficult for us to obtain good results based on the existing boundary conditions. Therefore, it is of great interest to design a model that can be compatible with more geometric information and enable us to integrate grid regularization constraints.

3.3. A Unified Document Image Dewarping Model

To motivate the following studies, we first re-examine the dewarping function from the perspective of energy minimization. We approximate the solution by minimizing the transformation energy as in Eqn. (2).

$$\varepsilon = \varepsilon_\phi + \lambda \varepsilon_d,$$
$$\phi : (x, y) \mapsto (u, v), \qquad (2)$$
$$(x, y) \in \Omega_S, (u, v) \in \Omega_T.$$

The ε represents the total energy of the dewarping system; ϕ represents the deformation which transforms the source image S into the target image T, that is $\phi(x, y) = (u, v)$, with $(x, y) \in \Omega_S$ being the coordinate point in the domain of S, $(u, v) \in \Omega_T$ being the corresponding coordinate point in the domain of T. ε_ϕ, ε_d represent the data penalty energy (usually points displacement) and surface distortion energy, respectively. λ is a hyper parameter to balance the energies between the data penalty and the distortion. The solution with the smallest total energy is the image deformation we expected.

We can use this uniform image dewarping model to summarize some previous methods including both traditional methods and deep learning methods.

Let $\mathbf{p}$ and $\mathbf{q}$ represent the source coordinate points and the target coordinate points, respectively. ε_ϕ is the control points displacement loss according to Eqn. (3).

$$\varepsilon_\phi = \sum_{i=1}^{N} \|\phi(\mathbf{p_i}) - \mathbf{q_i}\|_2^2. \qquad (3)$$

For the traditional methods, if taking $\varepsilon_d = \iint_{\mathbb{R}^2} (\phi_{xx}^2 + \phi_{yy}^2) dx dy$ and $N = 4$, we will get the projection transformation. And the transfinite interpolation (TFI) also fits this form, which regards boundary points as control points and uses the same ε_d as the projection transformation. For the thin plate spline (TPS) interpolation, ε_ϕ characterize the displacement of boundary points while ε_d is an integral to characterize the degree of deformation, as in

$$\varepsilon_d = \iint_{\mathbb{R}^2} (\frac{\partial^2 \phi}{\partial x^2})^2 + 2(\frac{\partial^2 \phi}{\partial x \partial y})^2 + (\frac{\partial^2 \phi}{\partial y^2})^2 dx dy. \qquad (4)$$

In the deep-learning based methods, we add all points to points displacement energy ε_ϕ. As for the second item, DocUNet [27] has $\lambda = 0$. While DewarpNet [10] has a regularization term with binary classification using checkerboard. The flow estimation [39] method has a constraint on local adjacency flow. And Xie *et al.* [40] achieves accurate grid prediction by reducing the number (N) of control points.

4. The Proposed Method

Similar to DocUNet [27], we approach the document image dewarping problem by calculating a forward 2D deformation field. In our model (Fig. 3), we first detect geometric elements such as the boundaries of the document and text lines in the document images and discretize them into points. Next, we minimize an objective function incorporating the grid regularization to obtain an optimal deformation field. Compared with the previous methods, our model utilizes explicit geometry priors to dewarp the distortions in document images.

4.1. Grid Regularization with Text Lines

In this subsection, we elaborate on how to use the unified document image deformation model to generate a deformation grid for our problem. We discretize the geometric conditions of lines into the energy form of points and use a regularization term to constrain the deformation grid.

Text lines and vertical lines. We use a common semantic segmentation network, *i.e.* UNet, to extract text lines from document images, and obtain the boundaries of text lines by Algorithm 1.

Algorithm 1 Detecting the Boundary of Text Lines

Input:
 Text lines of the document image;
 Hyper-parameter: w = 15 pixels, h = 15 pixels, $\theta =$ arctan(0.45);
Output:
 The boundaries of text lines;
1: Let A is the set of left endpoints $d(x, y)$ of text lines, and B is the set of right endpoints.
2: Calculate the vertical direction g of each endpoint, which is determined by the average gradient of the three adjacent control points to itself.
3: For each $d(x, y) \in A$, we search for another point $e \in A$ which is nearest to point d in the region of $[x - w : x + w, y - h : y]$ and the line gradient between d and e with the condition that g is less than θ. Then connect d and e.
4: We do the same thing for B, the connecting lines form the connected component M_1.
5: Change the search area to $[x - w : x + w, y : y + h]$ to obtain the connected component M_2.
6: The final result of $M_1 \cap M_2$ which can effectively avoid some error situations.

UV patterns in source image. As illustrated in Fig. 4, from the visualizations of deformed grids, we find that the position of the boundaries has some characteristic in the forward mapping: Points on the left boundary have a 0 value, and points on the right boundary has a value of 1 in the U map. And the position values increase uniformly from the
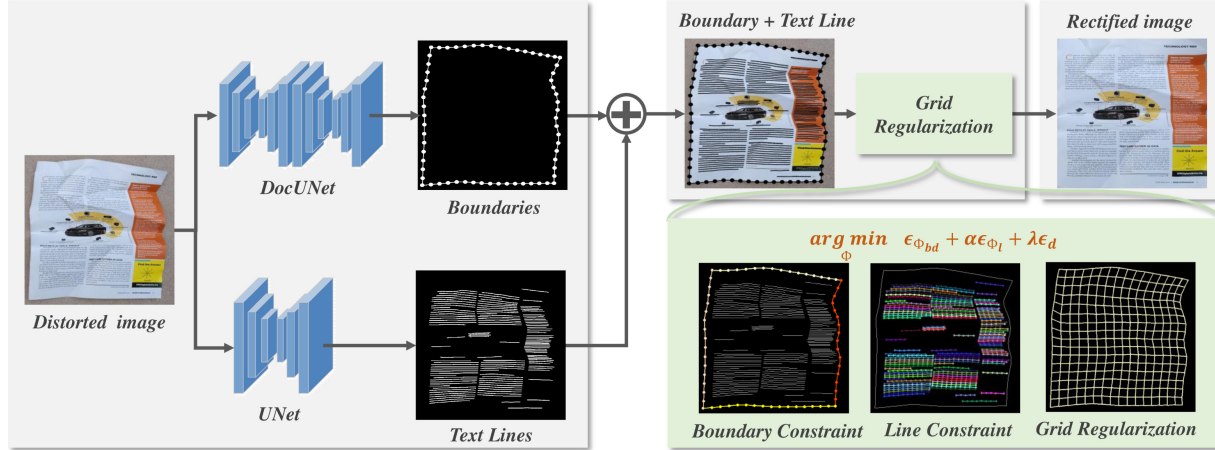
Figure 3. The pipeline of the proposed method. Taking a document image as input, DocUNet detects the backward mapping(BM) of document image boundaries by regression and UNet detects text lines by segmentation. The obtained geometric elements are discretized into points and fed into the grid regularization module. The proposed regularization module takes the boundary constraint, text line constraint and the grid regularization term as optimization conditions to calculate a uniform forward map (UV map).
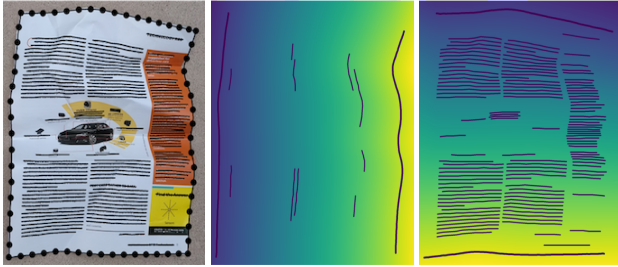

Figure 4. The UV patterns that can be used to develop some equality constraints on UV map.

left to the right in the U map. Being the same as U map, points in the V map have the value of 0's or 1's for the top and the bottom boundary, respectively. Such rules have practical implications that we can make the boundary in S go where it should be in T. The points on the same vertical line share the same u value, while the points on the same text line have the same v value, which can guide the lines alongside the text line to go horizontal and the lines perpendicular to the text line to go vertical in T. These rules can be used to develop some equality constraints on the UV map.

Establishment of optimization problem. We first discretize lines into points. Therefore, we have boundary points, text line points and vertical line points. For convenience, we split $\phi(x, y)$ into two parts:

$$\forall (x,y) \in \Omega_S, \ \phi(x,y) = (\phi_1(x,y), \ \phi_2(x,y)) = (u,v).$$

The energy of the points on the document image boundary have the form in Eqn. (5).

$$\varepsilon_{\phi_{bd}} = \sum_{k=1}^{K} \|\phi_i(x_k, y_k) - j\|_2^2, \qquad (5)$$

where K is the number of points in every boundary, with $bd \in \{top, left, bottom, right\}$. If we set $(i, j) = (0, 0)$, then $\varepsilon_{\phi_{bd}}$ implies that the points on the left boundary of the document image in S need to go to the left boundary in T. $(i, j) = (0, 1), (1, 0), (1, 1)$ represent right boundary, top boundary and bottom boundary, respectively.

The points on the same text line in S have the following energy form in Eqn. (6).

$$\varepsilon_{\phi_l} = \sum_{k=1}^{J-1} \|\phi_i(x_k, y_k) - \phi_i(x_{k+1}, y_{k+1})\|_2^2, \qquad (6)$$

with (x_k, y_k) and $\phi_i(x_{k+1}, y_{k+1})$ being two adjacent points on the same line. J is the number of points in a particular text line. The same thing for the vertical line. A horizontal text line has $i = 1$, and a vertical line has $i = 0$.

Inspired by regularization terms in TFI and TPS, we define our regularization terms by Eqn. (7).

$$\varepsilon_d = \iint_{\mathbb{R}^2} \left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2}\right)^2 + \beta \left(\frac{\partial^2 \phi}{\partial x \partial y}\right)^2 dx dy. \qquad (7)$$

In order to make the optimization problem be compatible with the interpolation of dense control points, we merge ϕ_{xx} and ϕ_{yy} that guarantee the smoothness of the grid. ϕ_{xy} keeps the shape of the grid.

So far, we can write down our optimization problem in Eqn. (8), by solving which we can obtain the grid deformation ϕ from S to T.

$$\arg \min_{\phi} \sum_{k=1}^{4} \varepsilon_{\phi_{bd_k}} + \alpha \sum_{k=1}^{N_1} \varepsilon_{\phi_{l_k}} + \lambda \varepsilon_d. \qquad (8)$$

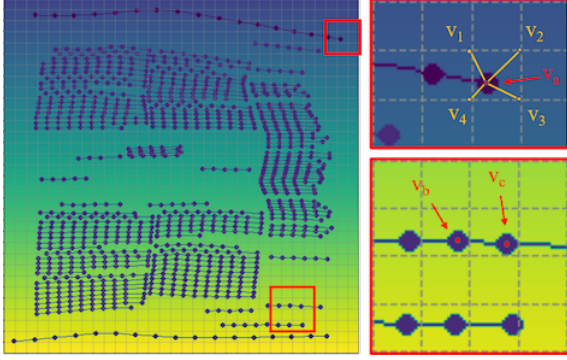where N_1 is the number of lines, and α, λ are parameters to balance different energy terms.

Figure 5. Calculation of V map: For every point a in V map: $v_a = \sum_{i=1}^{4} w_i \cdot v_i$, with w_i being the coefficient of bilinear interpolation assigned to the four surrounding grid points v_i. For point a in the top boundary, we have $v_a = 0$. For point b, c in the same text line, we have $v_b - v_c = 0$.

Discretizing the optimization problem. To achieve a discrete solution of the transformation ϕ, we need to discretize the data penalty energy and surface distortion energy. For a point (x, y) on the boundary, $\phi(x, y)$ could be discrete to the nearest 4 grid points $\phi(x_i, y_i)$ by Eqn. (9).

$$\phi(x, y) = \sum_{i=1}^{4} w_i \cdot \phi(x_i, y_i), \tag{9}$$

where w_i is a bilinear interpolation coefficient, which has been used to apply known conditions to unknown grid points. The details of calculation are shown[2] in Fig. 5. This form can also be applied to lines.

The distortion energy ε_d can be discretized by,

$$\sum_{i,j} \Big(\phi[i+1, j] + \phi[i-1, j] + \phi[i, j+1] + \phi[i, j-1] - 4\phi[i, j] \Big)^2$$
$$+ \beta \sum_{i,j} \Big(\phi[i+1, j+1] - \phi[i+1, j] - \phi[i, j+1] + \phi[i, j] \Big)^2.$$

4.2. Implementation Details

Computing the document boundary constraints. We obtain the document boundary by using DocUNet trained on the Doc3D dataset [10]. As we focus on boundary information, we add some preprocessing to the boundary, such as color changes and flip operations. The image size for training is 128×128. And the train loss for DocUNet is defined as L_1 distance between the predicted boundary, *i.e.*, bd_{pred} with $bd \in \{top, bottom, left, right\}$ and the ground truth boundary of the deformation field bd_{gt}.

Computing text-lines constraints. We train a UNet to detect text-line masks in document images with the help of

---

[2]Fig. 2 shows the *groundtruth UV maps* while Fig. 5 illustrates the *final UV predictions* that have the minimal deformation energy, therefore the background values are empty in Fig. 2.

PubLayNet [45] and the Scanned Script Dataset [44]. The image size for training is 512×512. The UV labels from the Doc3D dataset [10] are used to warp the image and text lines masks. Since the masks of the text lines are always much less than the background, we use an L_2 loss weighted by the pixel proportions [41] to train the UNet model.

Grid regularization. We discrete the line segments into points on a 512×512 image grid. The interval between points on the same text line is 16 pixels and the interval between points on the same vertical line is 10 pixels. By virtue of the geometry of these discrete points, one can redefine the data penalty term and image distortion term in the image distortion model accordingly. We set the hyper-parameter $n = 128, \alpha = 10, \lambda = 2, \beta = 20$ and we solve the optimization problem using Alternating Direction Method of Multipliers (ADMM) [12] for Quadratic Programming (QP). When solving this optimization problem, due to the particularity of grid coordinates, we can optimize ϕ_1 and ϕ_2 separately and change the problem from a $n \times n \times 2$ dimension problem to two $n \times n$ dimension problems.

Post-processing. Once the forward map is obtained, we firstly generate the backward map (BM: UV $\mapsto [0, 1] \times [0, 1]$, LinearNDinterpolator) and then upsample the BM by a bilinear interpolation operation to obtain the high-resolution BMs. Then, for each pixel in the obtained BM, we sample the corresponding RGB value from the input image to yield the final results.

5. Experiments and Analysis

5.1. Datasets and Evaluation Metrics

Dataset. For experiments, we use the DocUNet [27] benchmark which has 130 images from real-world scenes and 50 images with text labels [9] which can help us analyze OCR performance.

CER and ED: We use CER (Character Error Rate) and ED (Edit Distance) [20] to evaluate a dewarping method. Specifically, for a rectified document image, CER computes the ratio of the unexpected deletions, insertions and substitutions among the reference string, while ED measures the dissimilarity between the OCR results from the rectified image and the ground truth labels. For ED metric, we use PyTesseract (v0.3.8) [33] for computation.

MS-SSIM and LD: For the rectified document images, we also leverage the image-based metrics of MS-SSIM (*i.e.*, multi-scale SSIM) [38] and the Local Distortion (LD) [24] metric to evaluate the results from global and local perspectives. In the implementation, we use the evaluation code provided by DocUNet for computation.

5.2. Comparison with the State of the Arts

We quantitatively and qualitatively compare our method with the S.O.T.A. deep learning methods [10,11,21,27,40].

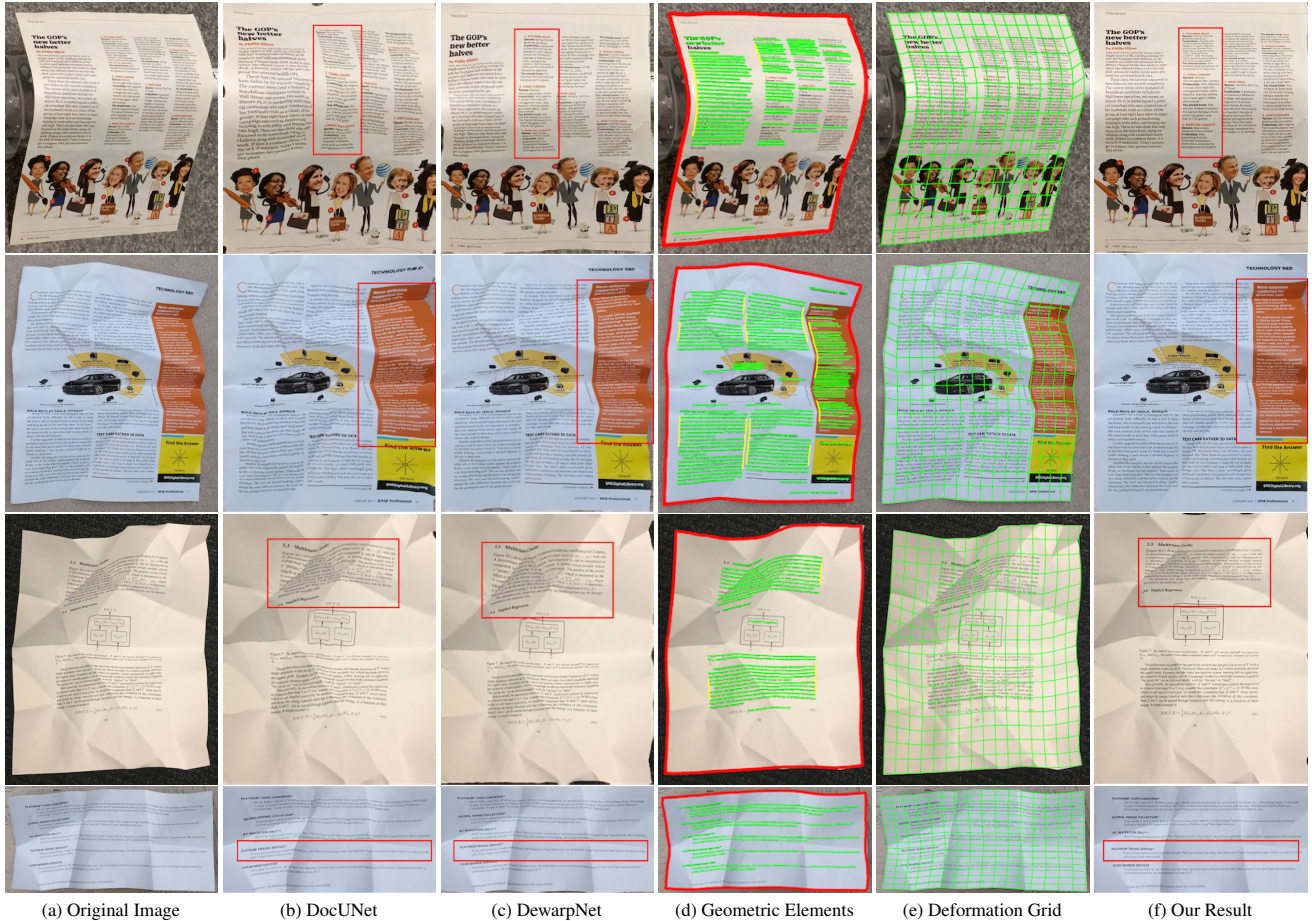| (a) Original Image | (b) DocUNet | (c) DewarpNet | (d) Geometric Elements | (e) Deformation Grid | (f) Our Result |

Figure 6. Qualitative comparison with previous methods: Our method uses text lines and vertical lines to guide the generation of deformation grid, and the rectified images are consistent with our expectations for Geometric Elements in document images. The green lines in the image of Geometric Elements are text lines and the yellow lines are the vertical lines extracted by our method.

Table 3. Quantitative comparsion of the proposed and previous methods on DocUNet benchmark. Standard deviation is reported in the parentheses. "↑" indicates the higher the better and "↓" means the opposite.

Method	$CER(std) \downarrow$	$ED \downarrow$	$MS\text{-}SSIM \uparrow$	$LD \downarrow$
DocUNet [27]	0.3955 (0.272)	1684.34	0.4389	10.90
DewarpNet [10]	0.3136 (0.248)	1288.60	0.4692	8.98
Flow Estimation [39]	0.4472 (0.274)	2000.04	0.4361	**8.50**
Xie *et al.* [40]	-	-	0.4769	9.03
Das *et al.* [11]	0.3001 (0.14)	-	0.4879	9.23
Ours	**0.2068 (0.141)**	**896.48**	**0.4922**	9.36

In those approaches, DocUNet [27] did not use any grid regularization while DewarpNet [10] using the checkerboard reconstruction term for dewarping. For the flow-based approach, Xie *et al.* [40] utilize Laplacian grid to achieve a better performance. We also compared our method with the splitting-based approaches [11, 21].

Quantitative Comparison. As reported in Table 3, our proposed approach achieves the best performance for the metrics of CER, ED and MS-SSIM. Benefiting from the regularization formulation with text lines and the bound-aries, our method reduces the CER by 9 points and ED by 30%. For the image-based metrics of MS-SSIM and LD, our method obtains better structural similarity while maintaining a comparable result of the local distortion.

Qualitative Comparison. Fig. 6 shows the qualitative comparison with the prior arts. It is illustrated that our proposed method handles severe distortions very well. For the interior of the rectified images, our dewarping results are more smooth than DocUNet and DewarpNet as we did not use the predicted the deformation field in the interior regions. Alternatively, we obtain the deformation field in the internal regions via optimization with our proposed regularization term. Furthermore, we show the local comparison between our approach and DewarpNet in Fig. 7.

5.3. Ablation Study

We will analyze the feasibility of using a grid regularization optimizer to replace the traditional interpolation algorithm and the influence of the geometric information on the grid regularization optimizer.
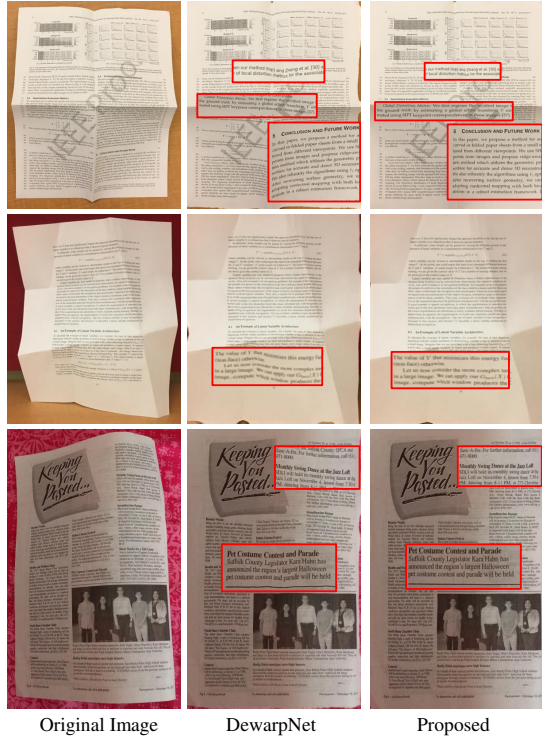
| Original Image | DewarpNet | Proposed |

Figure 7. Local comparison of the proposed method with Dewarp-Net. The document image obtained by our proposed method has better readability. The red boxes show the enlarged details.

Table 4. Ablation experiment on different the grid regularization terms. We replace the proposed grid regularization with the traditional interpolation algorithm (TFI, TPS) for comparison. In the case of the same boundary control points, our proposed method can get slightly better results than the traditional interpolation algorithm in the image quality evaluation index.

Method	CER(std) ↓	ED ↓	MS-SSIM ↑	LD ↓
DocUNet	0.3955 (0.272)	1684.34	0.4389	10.90
Boundary+TFI	0.3379 (0.165)	**1406.32**	0.4821	9.74
Boundary+TPS	**0.3340 (0.178)**	1457.30	0.4830	**9.42**
Boundary+GR	0.3511 (0.183)	1513.04	**0.4833**	**9.42**

Table 5. Ablation studies on the decision choices of using different geometric elements of boundaries, text lines and vertical lines (V).

Method	CER(std) ↓	ED ↓	MS-SSIM ↑	LD ↓
DocUNet	0.3955 (0.272)	1684.34	0.4389	10.90
Boundary	0.3511 (0.183)	1513.04	0.4833	9.42
Boundary+Text Line	0.2081 (0.142)	908.84	0.4907	9.44
Boundary+Text Line + V	**0.2068 (0.141)**	**896.48**	**0.4922**	**9.36**

Grid Regularization *vs*. Traditional Interpolation. In this ablation study, we use a grid regularization method to replace the traditional interpolation algorithm. All methods in Table 4 use the same boundary points generated by our method. It demonstrates that our grid regularization optimizer can completely replace the traditional interpolation algorithm. In fact, ϕ obtained by our method can be re-

garded as two ruled surfaces of TFI with some distortion in accordance with the geometric structure of the text. And because we regard the grid regularization method as an optimization problem, the scalability is better than the traditional interpolation algorithm, which facilitates us to add geometric information into the subsequent algorithm.
The Influence of Different Geometric Information. In this study, we show the influence of boundaries, text lines and vertical lines for the final performance. As shown in Table 5, all those geometric elements are positive for document image dewarping.

5.4. Limitations

Although we have obtained inspiring results on the Do-cUNet benchmark dataset, there are still some issues with our approach. Firstly, due to the computational complexity of the optimization problem, the size of the deformation grid unit is currently set to be 128×128, which somewhat limits the density of the discrete geometric information and makes it difficult to continuously pick points in an extremely small area. This might be the main reason that the rectified result in the third column of Fig. 6 (f) has some parts moving toward the upper boundary. In order to get more precise rectified results, one can refine the grid by further pursuing some fast optimizers for the energy minimization task. Secondly, the current version of our method is not end-to-end, thus we may need to balance the different energy terms in the dewarping process. Thus, one future direction is to develop an end-to-end trainable version of our method to automatically learn all the parameters.

6. Conclusion

This paper studies the problem of document image dewarping. By revisiting the deep-learning based dewarping approaches of document images, we address the remained challenge on the readability of the rectified document images in a geometric perspective with the text lines and image boundaries. With the learned geometric elements available, we design a grid regularization term on the deformation grid to estimate the 2D deformation field by solving an optimization problem. In our experiments, we demonstrate the effectiveness of our proposed approach with a new state-of-the-art performance on the DocUNet benchmark obtained.

References

[1] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE PAMI*, 11(6):567–585, 1989.

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

[3] Michael S Brown and W Brent Seales. Image restoration of arbitrarily warped documents. *IEEE PAMI*, 26(10):1295–1306, 2004.

[4] Michael S Brown, Mingxuan Sun, Ruigang Yang, Lin Yun, and W Brent Seales. Restoring 2d content from distorted documents. *IEEE PAMI*, 29(11):1904–1916, 2007.

[5] Michael S Brown and Y-C Tsoi. Geometric and shading correction for images of printed materials using boundary. *IEEE TIP*, 15(6):1544–1554, 2006.

[6] Huaigu Cao, Xiaoqing Ding, and Changsong Liu. A cylindrical surface model to rectify the bound document image. In *ICCV*, pages 228–233, 2003.

[7] Huaigu Cao, Xiaoqing Ding, and Changsong Liu. Rectifying the bound document image captured by the camera: A model based approach. In *ICDAR*, pages 71–75, 2003.

[8] Frédéric Courteille, Alain Crouzil, Jean-Denis Durou, and Pierre Gurdjos. Shape from shading for the digitization of curved documents. *Machine Vision and Applications*, 18(5):301–316, 2007.

[9] Sagnik Das and Ke Ma. Text labels of doc3d benchmark, https://github.com/cvlab-stonybrook/DewarpNet, 2019.

[10] Sagnik Das, Ke Ma, Zhixin Shu, Dimitris Samaras, and Roy Shilkrot. Dewarpnet: Single-image document unwarping with stacked 3d and 2d regression networks. October 2019.

[11] Sagnik Das, Kunwar Yashraj Singh, Jon Wu, Erhan Bas, Vijay Mahadevan, Rahul Bhotika, and Dimitris Samaras. End-to-end piece-wise unwarping of document images. In *ICCV*, pages 4268–4277, 2021.

[12] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *JMLR*, 17(83):1–5, 2016.

[13] Gerald Farin and Dianne Hansford. Discrete coons patches. *CAGD*, 16(7):691–700, 1999.

[14] Hao Feng, Yuechen Wang, Wengang Zhou, Jiajun Deng, and Houqiang Li. Doctr: Document image transformer for geometric unwarping and illumination correction. *arXiv:2110.12942*, 2021.

[15] William J Gordon and Charles A Hall. Transfinite element methods: blending-function interpolation over arbitrary curved element domains. *Numerische Mathematik*, 21(2):109–129, 1973.

[16] Nail Gumerov, Ali Zandifar, Ramani Duraiswami, and Larry S Davis. Structure of applicable surfaces from single views. In *ECCV*, pages 482–496. Springer, 2004.

[17] Taeho Kil, Wonkyo Seo, Hyung Il Koo, and Nam Ik Cho. Robust document image dewarping method using text-lines and line segments. In *ICDAR*, volume 1, pages 865–870, 2017.

[18] Beom Su Kim, Hyung Il Koo, and Nam Ik Cho. Document dewarping via text-line based optimization. *Pattern Recognition*, 48(11):3600–3614, 2015.

[19] Hyung Il Koo, Jinho Kim, and Nam Ik Cho. Composition of a dewarped and enhanced document image from two view images. *IEEE TIP*, 18(7):1551–1562, 2009.

[20] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.

[21] Xiaoyu Li, Bo Zhang, Jing Liao, and Pedro V Sander. Document rectification and illumination correction using a patch-based cnn. *ACM TOG*, 38(6):1–11, 2019.

[22] Jian Liang, Daniel DeMenthon, and David Doermann. Flattening curved documents in images. In *CVPR*, volume 2, pages 338–345, 2005.

[23] Jian Liang, Daniel DeMenthon, and David Doermann. Geometric rectification of camera-captured document images. *IEEE PAMI*, 30(4):591–605, 2008.

[24] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE PAMI*, 33(5):978–994, 2010.

[25] Xiyan Liu, Gaofeng Meng, Bin Fan, Shiming Xiang, and Chunhong Pan. Geometric rectification of document images using adversarial gated unwarping network. *Pattern Recognition*, 108:107576, 2020.

[26] Shijian Lu and Chew Lim Tan. Document flattening through grid modeling and regularization. In *ICPR*, volume 1, pages 971–974, 2006.

[27] Ke Ma, Zhixin Shu, Xue Bai, Jue Wang, and Dimitris Samaras. Docunet: Document image unwarping via a stacked u-net. In *CVPR*, June 2018.

[28] Amir Markovitz, Inbal Lavi, Or Perel, Shai Mazor, and Roee Litman. Can you read me now? content aware rectification using angle supervision. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, pages 208–223, Cham, 2020. Springer International Publishing.

[29] Gaofeng Meng, Chunhong Pan, Shiming Xiang, Jiangyong Duan, and Nanning Zheng. Metric rectification of curved document images. *IEEE PAMI*, 34(4):707–722, 2011.

[30] Gaofeng Meng, Yuanqi Su, Ying Wu, Shiming Xiang, and Chunhong Pan. Exploiting vector fields for geometric rectification of distorted document images. In *ECCV*, pages 172–187, 2018.

[31] Gaofeng Meng, Ying Wang, Shenquan Qu, Shiming Xiang, and Chunhong Pan. Active flattening of curved document images via two structured beams. In *CVPR*, pages 3890–3897, 2014.

[32] Vijaya Kumar Bajjer Ramanna, Syed Saqib Bukhari, and Andreas Dengel. Document image dewarping using deep learning. In *ICPRAM*, 2019.

[33] Ray Smith. An overview of the tesseract ocr engine. In *ICDAR*, volume 2, pages 629–633, 2007.

[34] Nikolaos Stamatopoulos, Basilis Gatos, Ioannis Pratikakis, and Stavros J Perantonis. Goal-oriented rectification of camera-based document images. *IEEE TIP*, 20(4):910–920, 2010.

[35] Yuandong Tian and Srinivasa G Narasimhan. Rectification and 3d reconstruction of curved document images. In *CVPR*, pages 377–384, 2011.

[36] Yau-Chat Tsoi and Michael S Brown. Multi-view document rectification using boundary. In *CVPR*, pages 1–8, 2007.

[37] Adrian Ulges, Christoph H Lampert, and Thomas Breuel. Document capture using stereo vision. In *Proceedings of ACM Symposium on Document Engineering*, pages 198–200, 2004.

[38] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.

[39] Guo-Wang Xie, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. Dewarping document image by displacement flow estimation with fully convolutional network. In *International Workshop on Document Analysis Systems*, pages 131–144. Springer, 2020.

[40] Guo-Wang Xie, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. Document dewarping with control points. In *ICDAR*, pages 466–480. Springer, 2021.

[41] Zhucun Xue, Nan Xue, Gui-Song Xia, and Weiming Shen. Learning to calibrate straight lines for fisheye image rectification. In *CVPR*, pages 1643–1651, 2019.

[42] Shaodi You, Yasuyuki Matsushita, Sudipta Sinha, Yusuke Bou, and Katsushi Ikeuchi. Multiview rectification of folded documents. *IEEE PAMI*, 40(2):505–511, 2017.

[43] Li Zhang, Yu Zhang, and Chew Tan. An improved physically-based method for geometric restoration of distorted document images. *IEEE PAMI*, 30(4):728–734, 2008.

[44] Niansong Zhang and Patrick Yang. Icdar robust reading challenge on scanned receipts ocr and information extraction, https://github.com/zzzDavid/ICDAR-2019-SROIE, 2019.

[45] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largest dataset ever for document layout analysis. In *ICDAR*, pages 1015–1022, 2019.