DewarpNet: Single-Image Document Unwarping With Stacked 3D and 2D Regression Networks

Sagnik Das* Ke Ma* Zhixin Shu Dimitris Samaras Roy Shilkrot
Stony Brook University
{sadas, kemma, zhshu, samaras, roys}@cs.stonybrook.edu

Abstract

Capturing document images with hand-held devices in unstructured environments is a common practice nowadays. However, "casual" photos of documents are usually unsuitable for automatic information extraction, mainly due to physical distortion of the document paper, as well as various camera positions and illumination conditions. In this work, we propose DewarpNet, a deep-learning approach for document image unwarping from a single image. Our insight is that the 3D geometry of the document not only determines the warping of its texture but also causes the illumination effects. Therefore, our novelty resides on the explicit modeling of 3D shape for document paper in an end-to-end pipeline. Also, we contribute the largest and most comprehensive dataset for document image unwarping to date – Doc3D. This dataset features multiple ground-truth annotations, including 3D shape, surface normals, UV map, albedo image, etc. Training with Doc3D, we demonstrate state-of-the-art performance for DewarpNet with extensive qualitative and quantitative evaluations. Our network also significantly improves OCR performance on captured document images, decreasing character error rate by 42% on average. Both the code and the dataset are released [1].

1. Introduction

Paper documents carry valuable information and serve an essential role in our daily work and life. Digitized documents can be archived, retrieved, and shared in a convenient, safe, and efficient manner. With the increasing popularity of portable cameras and smartphones, document digitization becomes more accessible to users through picture taking. Once captured, the document images can be converted into electronic formats, for example, a PDF file, for further processing, exchange, information extraction, and content analysis. While capturing images, it is desirable to preserve the information on the document with the best possible accuracy – with a minimal difference from a flatbed-scanned version. However, casual photos captured with mobile devices often suffer from different levels of distortions due to uncontrollable factors such as physical deformation of the paper, varying camera positions, and unconstrained illumination conditions. As a result, these raw images are often unsuitable for automatic information extraction and content analysis.

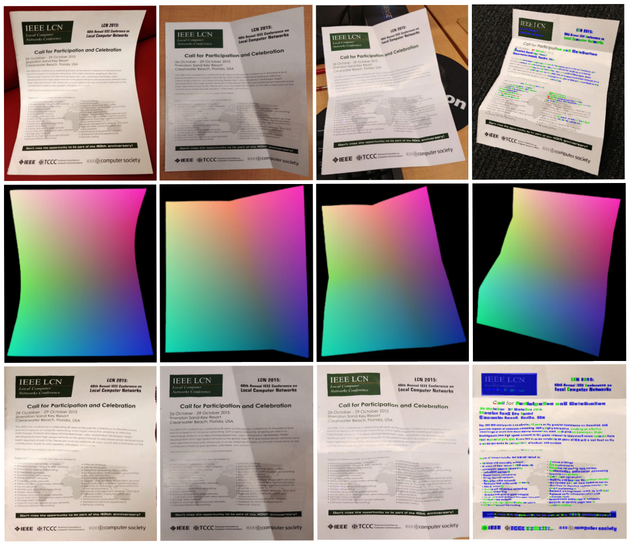
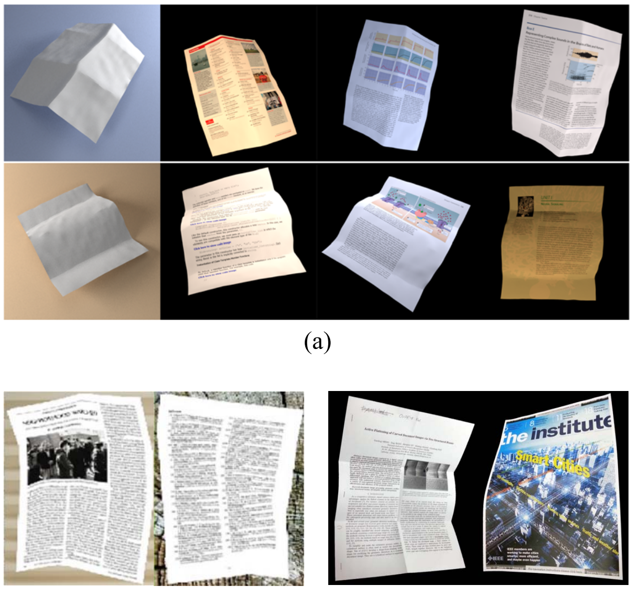


Figure 1. **Document image unwarping.** Top row: input images. Middle row: predicted 3D coordinate maps. Bottom row: predicted unwarped images. Columns from left to right: 1) curled, 2) one-fold, 3) two-fold, 4) multiple-fold with OCR confidence highlights in Red (low) to Blue (high).

Previous literature has studied the document-unwarping problem using various approaches. Traditional approaches [26, 46] usually rely on the geometric properties of the paper to recover the unwarping. These methods first estimate the 3D shape of the paper, represented by either some parametric shape representations [9, 47] or some non-parametric shape representations [35, 45]. After that, they compute the

*indicates equal contribution.

[1] https://www.cs.stonybrook.edu/~cvl/dewarpnet.html



(a)

(b)

(c)

Figure 2. **Comparison of different datasets.** (a) shows the images from our Doc3D dataset. We show 6 images rendered from 2 meshes here. Each mesh can be rendered with various textures and illumination conditions. (b) are the synthetic training images copied from [23]. (c) are the real world test images in [45] .

flattened image from the warped image and the estimated shape using optimization techniques. A common drawback of these methods is that they are usually computationally expensive and slow due to the optimization process. Recent work by Ma et al. [23] proposed a deep learning system that directly regresses the unwarping operation from the deformed document image. Their method significantly improved the speed of document unwarping system. However, their method did not follow the 3D geometric properties of the paper warping – training data was created with a set of 2D deformations – and therefore often generate unrealistic results in testing.

Paper folds happen in 3D: papers with different textures but the same 3D shape can be unwarped with the same deformation field. Hence, 3D shape is arguably the most critical cue for recovering the unwarped paper. Based on this idea, we propose DewarpNet, a novel data-driven unwarping framework that utilizes an explicit 3D shape representation for learning the unwarping operation. DewarpNet works in two-stages with two sub-networks: i) The "shape network" consumes an image of a deformed document and outputs a 3D-coordinate map which has shown to be sufficient for the unwarping task[45]. ii) The "texture mapping network" *backward maps* the deformed document image to a flattened document image. We train both sub-networks jointly with regression losses on the intermediate 3D shape and final unwarping result (Fig. 1). After that, we provide a "refinement network" that removes the shading effect from the rectified image, further improving the perceptual quality of the result.

To enable the training of this unwarping network with explicit intermediate 3D representation, we create the Doc3D dataset – the largest and most comprehensive dataset for document image unwarping to date. We collect Doc3D in a hybrid manner, combining (1) captured 3D shapes (meshes) from naturally warped papers with (2) photorealistic rendering of an extensive collection of document content. Each data point comes with rich annotations, including 3D coordinate maps, surface normals, UV texture maps, and albedo maps. In total, Doc3D contains approximately 100,000 richly annotated photorealistic images.

We summarize our contributions as follows:

First, we contribute the Doc3D dataset. To the best of our knowledge, this is the first and largest document image dataset with multiple ground-truth annotations in both 3D and 2D domain.

Second, we propose DewarpNet, a novel end-to-end deep learning architecture for document unwarping. This network enables high-quality document image unwarping in real-time.

Third, trained with the rich annotations in the Doc3D dataset, DewarpNet shows superior performance compared to recent state-of-the-art [23]. Evaluating with perceptual similarity to real document scans, we improve the Multi-Scale Structural Similarity (MS-SSIM) by 15% and reduce the Local Distortion by 36%. Furthermore, we demonstrate the practical significance of our method by a 42% decrease in OCR character error rate.

2. Previous Work

Based on how deformation is modeled, the two groups of prior work on document unwarping are: parametric shape-based models and non-parametric shape-based models:

Parametric shape-based methods assume that document deformation is represented by low dimensional parametric models and the parameters of these models can be inferred using visual cues. Cylindrical surfaces are the most prevalent parametric models [8, 16, 19, 26, 41, 46]. Other models include Non-Uniform Rational B-Splines (NURBS) [10, 44], piece-wise Natural Cubic Splines (NCS) [36], Coon patches [9], etc. Visual cues used for estimating model parameters include text lines [25], document boundaries [5], or laser beams from an external device [27]. Shafait and Breuel [33] reported several parametric shape based methods on a small dataset with only perspective and curl distortions. However, it is difficult for such low dimensional models to model complex surface deformations.

Non-parametric shape-based methods, in contrast, do not rely on low-dimensional parametric models. Such methods usually assume a mesh representation for the de-

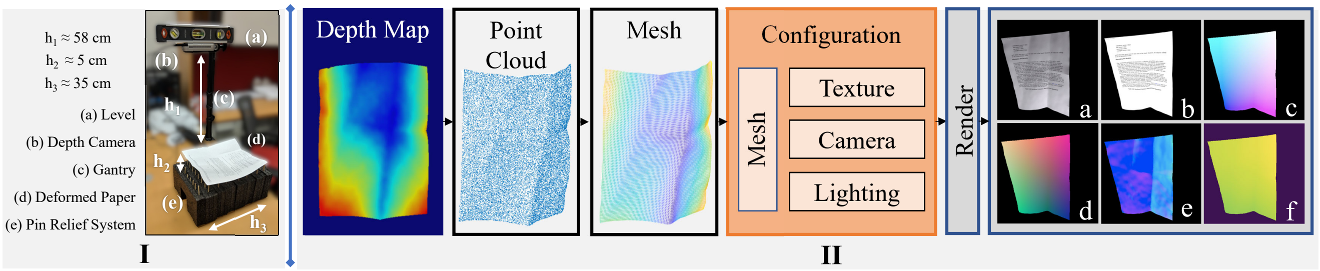


Figure 3. **Data collection pipeline.** I. Workstation. A leveled depth camera mounted on a gantry captures the deformed document. A pin relief system precisely controls the warping. II. Data processing. We turned the depth map into a point cloud to reconstructed a mesh. With multiple rendering configurations, we rendered (a) images, (b) albedo maps, (c) UV maps, (d) 3D coordinate maps, (e) Surface normals, (f) depth maps

formed document paper, and directly estimate the position of each vertex on the mesh. Approaches used to estimate the vertex positions, include reference images [29], text lines [21, 35, 39], and Convolutional Neural Networks (CNNs) [30]. Many approaches reconstruct the mesh from estimated or captured 3D paper shape information. Notable examples are point clouds estimated from stereo vision [38], multi-view images [45], structured light [4], laser range scanners [47], etc. There is also work on directly using texture information for this task [11, 24, 43]. However, resorting to external devices or multi-view images makes the methods less practical. Local text line features cannot handle documents that mix text with figures. Moreover, these methods often involve complicated and time-consuming optimization. Recently, Ma et al. [23] proposed "DocUNet", which is the first data-driven method to tackle document unwarping with deep learning. Compared to prior approaches, DocUNet is faster during inference but does not always perform well on real-world images, mainly because the synthetic training dataset only used 2D deformations.

3. The Doc3D Dataset

We created the Doc3D dataset in a hybrid manner, using both real document data and rendering software. We first captured the 3D shape (mesh) of naturally deformed *real* document paper. After that, we rendered the images with *real* document texture in Blender [1] using path tracing [40]. We used diverse camera positions and varying illumination conditions in rendering.

A significant benefit of our approach is that the dataset is created in large scale with photorealistic rendering. Meanwhile, our method generates multiple types of pixel-wise document image ground truth, including 3D coordinate maps, albedo maps, normals, depth maps, and UV maps. Such image formation variations are useful for our task, but usually harder to obtain in real-world acquisition scenarios.

Compared with the dataset in [23] where 3D deformation was modeled in 2D only [28], our dataset simulates document deformation in a physically-grounded manner. Thus, it is reasonable to expect that deep-learning models trained on our dataset will generalize better when testing on real-world images, compared to models trained on the dataset of [23]. We visually compare dataset samples in Fig. 2.

3.1. Capturing Deformed Document 3D Shape

3D point cloud capture. Our workstation (Fig. 3 (I)) for deformed document shape capture consists of a tabletop, a gantry, a depth camera, and a relief stand. The gantry holds the depth camera level, facing towards the tabletop, at the height of 58 cm. At this height, the depth camera captures the whole document while still preserving deformation details. The relief stand has 64 individually controlled pins, raising the height of the document to isolate it from the tabletop. The height differences make it easier to extract the document from the background in the depth map. The stand simulates complex resting surfaces for the document and also supports the deformed document to maintain curls or creases.

We used a calibrated Intel RealSense D415 depth camera to capture the depth map. Assuming no occlusion, the point cloud of the document was obtained via $X^{(3D)} = K^{-1}[i, j, d_{ij}]^T$, where d_{ij} is the depth value at the pixel position i, j in the depth map. The intrinsic matrix K was read from the camera. We averaged 6 frames to reduce zero-mean noise, and applied Moving Least Squares (MLS) [32] with a Gaussian kernel to smooth the point cloud.

Mesh creation. We extracted a mesh from the captured point cloud using the ball pivoting algorithm [3]. The mesh has ~130,000 vertices and 270,000 faces covering all vertices. We then subsampled each mesh to a 100 × 100 uniform mesh grid to facilitate mesh augmentation, alignment, and rendering. Due to the accuracy limits of our inexpensive sensor, even a higher resolution mesh grid cannot provide finer details like subtle creases. Each vertex has a UV position, to indicate texture coordinates, used for texture mapping in the rendering step. Assigning $(u, v) = \{(0,0), (0,1), (1,0), (1,1)\}$ to the 4 corner vertices of the

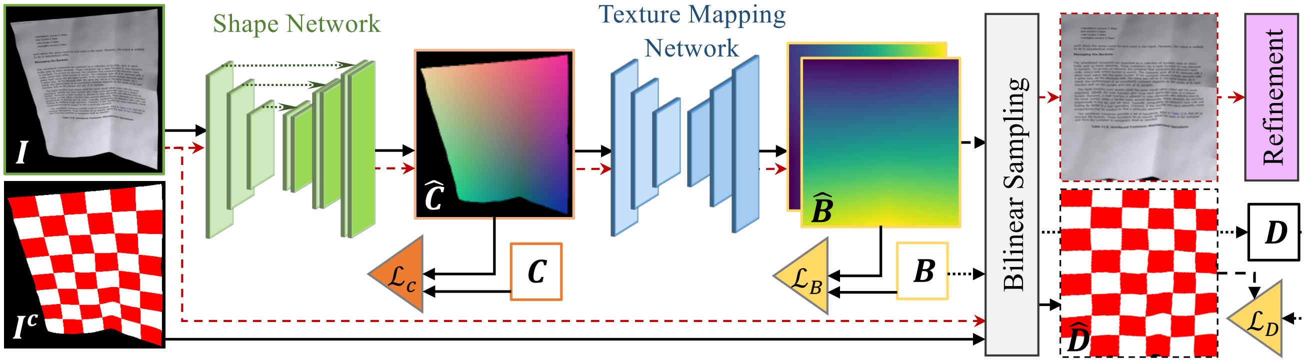


Figure 4. **DewarpNet Framework.** $\mathbf{I}$ is the input deformed document image. $\mathbf{I^c}$ is the $\mathbf{I}$ in checkerboard pattern texture. Training Flow is in black lines. The two black dashed lines refer to the predicted ($\hat{\mathbf{D}}$) and ground-truth ($\mathbf{D}$) unwarped reconstruction patterns. Testing flow is in red dashed lines. Triangles denote the losses (see Sec. 4.2 for details). $\mathbf{C}$ and $\mathbf{B}$ are the ground-truth for the 3D coordinates and the backward mapping respectively.

mesh, we interpolated UV values for all vertices [37].

Mesh augmentation and alignment. To further exploit each mesh, we first flipped the mesh along the x, y, z axes respectively resulting in 8 meshes, as well as randomly cropped out 4 small meshes ranging from 65×65 to 95×95 vertices in different aspect ratios. We interpolated all meshes to the same resolution of 100×100. These additional meshes significantly increased the diversity of the dataset. All meshes were aligned to a template mesh by solving an absolute orientation problem [13] to unify scale, rotation, and translation. This step ensured that one unique deformation had one unique 3D coordinate representation. In total, we generated 40,000 different meshes.

3.2. Document Image Rendering

Configuration. To increase the diversity of the dataset, we altered the configurations of camera, lighting, and texture in the rendering process. For each image, the camera was randomly placed on a spherical cap, with an "up" direction in $[-30°, 30°]$ range. The camera direction was constrained within a small area around the virtual world origin. We rendered 70% of the images using lighting environments randomly sampled from the 2100 environment maps in the Laval Indoor HDR dataset [12]. We also rendered 30% of the images under simple lighting conditions using a randomly sampled point light. The textures on the mesh were obtained from real-world document images. We collected 7,200 images of academic papers, magazines, posters, books, etc., containing a mix of text and figures in multiple layouts.

Rich annotations. For each image, we generated the 3D coordinate map, depth map, normals, UV map, and albedo map. In Sec. 4, we show how we incorporate these ground truth images into our network.

4. DewarpNet

4.1. Network Architecture

DewarpNet, as shown in Fig. 4, consists of two subnetworks for learning unwarping: the *shape network* and the *texture mapping network*. Additionally, we propose a post-processing refinement module for illumination effect adjustment that visually improves the unwarped images.

DewarpNet takes as input an image of a deformed document $\mathbf{I} \in \mathbb{R}^{h \times w \times 3}$ and predicts a backward mapping $\mathbf{B} \in \mathbb{R}^{h \times w \times 2}$ (h and w are height and width). The mapping $\mathbf{B}$ is a flow field representing an image deformation: each pixel (x, y) in $\mathbf{B}$ represents a pixel position in the input image $\mathbf{I}$. We use bilinear sampling to sample the pixel value in $\mathbf{I}$ to generate the final unwarped document image $\mathbf{D} \in \mathbb{R}^{h \times w \times 3}$.

Shape Network. DewarpNet first regresses the 3D shape of the input document image. We formulate this regression task as an image-to-image translation problem: given an input image $\mathbf{I}$, the shape network translates each pixel of $\mathbf{I}$ into the 3D coordinate map, $\mathbf{C} \in \mathbb{R}^{h \times w \times 3}$, where each pixel value (X, Y, Z) corresponds to 3D coordinates of the document shape, as shown in Fig. 4. We use a U-Net [31] style encoder-decoder architecture with skip connections in the shape network.

Texture Mapping Network. The texture mapping network takes the 3D coordinate map $\mathbf{C}$ as input and outputs the backward mapping $\mathbf{B}$. In the texture mapping network, we use an encoder-decoder architecture with multiple DenseNet [14] blocks. This task is a coordinate transformation from 3D coordinates in $\mathbf{C}$ to texture coordinates in $\mathbf{B}$. We apply *Coordinate Convolution* (CoordConv) in the texture mapping network since it was shown to improve the generalization ability of the network for coordinate transformation tasks [18, 22]. Our experiment shows the effectiveness of this technique in Sec. 5.5.

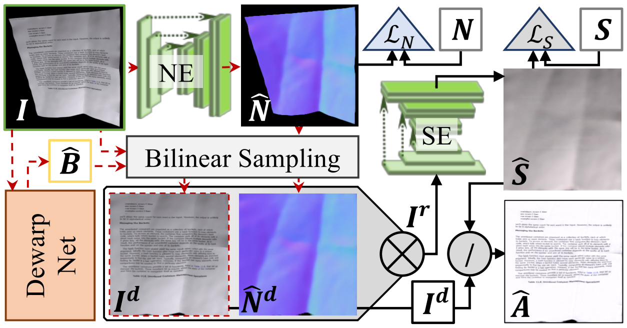


Figure 5. **Proposed Refinement Network.** $\hat{\mathbf{B}}$ is the predicted backward mapping. NE is the normal estimation network and $\hat{\mathbf{N}}$ are the predicted normals. $\mathbf{I}^d$ and $\hat{\mathbf{N}}^d$ are unwarped $\mathbf{I}$ and $\hat{\mathbf{N}}$ based on $\hat{\mathbf{B}}$. SE is the shading estimation network and $\hat{\mathbf{S}}$ is the predicted shading map. $\mathbf{I}^r$ is the concatenation ($\bigotimes$) of $\mathbf{I}^d$ and $\hat{\mathbf{N}}^d$. $\hat{\mathbf{A}}$ is the final shading free output and "/" represents element-wise division operator. Red dashed arrows signify the inference path.

Refinement Network. The refinement network serves as a post-processing component of our system to adjust for illumination effects in the rectified image. This network not only enhances perceptual quality of the results, but also improves OCR performance (Sec. 5.4). We leverage additional ground-truth information (i.e., surface normals and albedo maps) in the Doc3D dataset to train the refinement network. The refinement network has two U-Net [31] style encoder-decoders as shown in Fig. 5: one is used to predict the surface normals $\mathbf{N} \in \mathbb{R}^{h\times w\times 3}$ given the input image $\mathbf{I}$; the other takes $\mathbf{I}$ and the corresponding $\mathbf{N}$ as input and estimates a shading map $\mathbf{S} \in \mathbb{R}^{h\times w\times 3}$. $\mathbf{S}$ describes shading intensity and color. Then we recover the shading free image $\mathbf{A}$ based on an intrinsic image decomposition [2]: $\mathbf{I} = \mathbf{A} \odot \mathbf{S}$, where $\odot$ is the Hadamard product operator. More details are discussed in the supplementary material.

4.2. Training Loss Functions

The training process has two phases. In the first phase, the shape network and the texture mapping network are trained separately for initialization. In the second phase, the two sub-networks are trained jointly to improve the unwarping result. For convenience, we denote a predicted variable as $\hat{\mathbf{X}}$, and its ground-truth as $\mathbf{X}$. The shape network optimizes the loss function ($\mathcal{L}_C$) in Eq. 1 on the 3D coordinate map $\mathbf{C}$ defined in Sec. 4.1:

$$\mathcal{L}_C = \|\hat{\mathbf{C}} - \mathbf{C}\|_1 + \lambda \|\nabla\hat{\mathbf{C}} - \nabla\mathbf{C}\|_1 \tag{1}$$

where $\nabla\mathbf{C} = \|(\nabla_x\mathbf{C}, \nabla_y\mathbf{C})\|_2$, $\nabla_x\mathbf{C}$ and $\nabla_y\mathbf{C}$ are the horizontal and vertical image gradients of $\mathbf{C}$, and λ controls the gradient term's influence. The image gradient helps learn high-frequency details such as ridges and valleys of $\mathbf{C}$.

The texture mapping network is trained to minimize $\mathcal{L}_T$ in Eq. 2. This loss is defined as a linear combination of the loss term $\mathcal{L}_B$ on the predicted backward mapping $\hat{\mathbf{B}}$ and the loss term $\mathcal{L}_D$ on the predicted unwarped image $\hat{\mathbf{D}}$:

$$\mathcal{L}_T = \gamma \underbrace{\|\mathbf{B} - \hat{\mathbf{B}}\|_1}_{\mathcal{L}_B} + \delta \underbrace{\|\mathbf{D} - \hat{\mathbf{D}}\|_2}_{\mathcal{L}_D} \tag{2}$$

where γ and δ are weights associated to $\mathcal{L}_B$ and $\mathcal{L}_D$.

$\mathcal{L}_D$ is the reconstruction loss for the unwarped image. $\mathcal{L}_B$ is the regression loss of the absolute pixel coordinates. We optimize both $\mathcal{L}_B$ and $\mathcal{L}_D$ to improve unwarping results (see Sec. 5.5 for ablations).

During training, for each input image $\mathbf{I}$, we apply the corresponding ground truth deformation to a regular checkerboard pattern image $\mathbf{D}$, obtaining a checkerboard image $\mathbf{I^c}$. We use the predicted backward mapping $\hat{\mathbf{B}}$ to unwarp $\mathbf{I^c}$, obtaining the unwarped checkerboard image $\hat{\mathbf{D}}$ to calculate $\mathcal{L}_D$. The goal of checkerboard texture is to encourage the consistency of $\mathcal{L}_D$ across various input images regardless of the document texture. In other words, two images with identical deformations should unwarp in the same way irrespective of their content, which implies the same $\mathcal{L}_D$. Note that $\mathbf{I^c}$ is only used in training.

In the second phase, the shape and texture mapping networks are trained simultaneously in an end-to-end manner. Such joint optimization enables the backward mapping loss to compensate for imperfections in the shape network. The objective function $\mathcal{L}$ for end-to-end training (Eq. 3) is a weighted linear combination of $\mathcal{L}_C$ (3D coordinates) and $\mathcal{L}_T$ (texture map).

$$\mathcal{L} = \alpha\mathcal{L}_C + \beta\mathcal{L}_T \tag{3}$$

For the shading removal refinement task we use $\ell 1$ loss on $\mathbf{S}$ and $\hat{\mathbf{S}}$: $\mathcal{L}_S = \|\mathbf{S} - \hat{\mathbf{S}}\|_1$.

4.3. Training Details

We train our models on the Doc3D dataset of 100,000 images, splitting into training and validation sets such that they have no meshes in common. In the first phase of initilization training, the texture mapping network takes the ground truth 3D coordinate map $\mathbf{C}$ as input. Later, in the second phase of joint training, each sub-network is initialized with the best separately trained models. The input to the texture mapping network is the predicted 3D coordinate map $\hat{\mathbf{C}}$ from the shape network. $\hat{\mathbf{B}}$ ranges in $[-1, 1]$ whereas $\hat{\mathbf{C}}$ ranges in $[0, 1]$.

We apply multiple ways of data augmentation: We replace the background of our training data with images from the Describable Texture Dataset (DTD) [7] and the KTH2b-tips dataset [6] actively during training. The intensity and color of each training image are also randomly jittered.

Hyperparameters. Initially, we set $\lambda = 0.2$ (Eq. 1) then increase by 0.2 after every 50 epochs up to 1.0. We found that $\gamma = 10.0$ and $\delta = 0.5$ (Eq. 2) provide adequate

Class	Deformation Type
(a) Perspective	Flat, with perspective warping.
(b) Curled	Curved, without creases.
(c) One-Fold	One significant crease is visible.
(d) Multi-Fold	Multiple creases on the page.
(e) Random-Easy	Random folds and some crumples.
(f) Random-Hard	Hard crumples, irregular folding.

Table 1. Classification of samples in Doc3D.

reconstruction quality. For joint training we used $\alpha = \beta = 0.5$ (Eq. 3). We use the Adam solver [15] with a batch size of 40, and weight decay of 5×10^{-4}. The learning rate is initially set at 1×10^{-4}, and reduced by a factor of 0.5 if the loss does not reduce for 5 epochs.

5. Experiments

We evaluate our method with multiple experiments on the 130-image benchmark from [23], and also show qualitative results on real images from [45]. As a baseline, we train the DocUNet [23] unwarping method on our new Doc3D dataset. Furthermore, we evaluate OCR performance of our method from a document analysis perspective. Finally, we provide a detailed ablation study to show how the use of the Coordinate Convolutions [22], and the loss $\mathcal{L}_D$ affect unwarping performance. Qualitative evaluations are shown in Fig. 7.

5.1. Experimental Setup

Benchmark. For quantitative evaluation, we classify the 130-image benchmark [23] into six classes indicating six different levels of deformation complexity (see Table 1). The benchmark dataset contains various kinds of documents, including images, graphics, and multi-lingual text.

Evaluation Metrics. We use two different evaluation schemes based on (a) Image similarity and (b) Optical Character Recognition (OCR) performance.

We use two image similarity metrics: Multi-Scale Structural Similarity (MS-SSIM) [42] and Local Distortion (LD) [45], as quantitative evaluation criteria, following [23]. SSIM computes the similarity of the mean pixel value and variance within each image patch and averages over all the patches in an image. MS-SSIM applies SSIM at multiple scales using a Gaussian pyramid, better suited for the evaluation of global similarity between the result and ground-truth. LD computes a dense SIFT flow [20] from the unwarped document to the corresponding document scan, thus focusing on the rectification of local details. The parameters of LD are set to the default values of the implementation provided by [23]. For a fair comparison, all the unwarped output and target flatbed-scanned images are resized to a 598400 pixel area, as recommended in [23].

OCR accuracy is calculated in terms of Character Error Rate (CER). CER is evaluated by calculating the Edit Distance (ED) [17] between the *reference* and *recognized* text. ED is the total number of substitutions (s), insertions (i) and deletions (d) to obtain the *reference* text, given the *recognized* text. CER $= (s+i+d)/N$, where N is the number of characters in the *reference* text, which is obtained from the flatbed scanned document images.

5.2. DocUNet on Doc3D

We present a baseline validation of the proposed Doc3D dataset by training the network architecture in DocUNet [23] on our dataset – Doc3D. DocUNet is a 3D-agnostic model. The architecture consists of two stacked UNets. DocUNet takes a 2D image as input and outputs a forward mapping (each pixel represents the coordinates in the **texture image**). The supervisory signal is solely based on the ground truth forward mapping. Unlike the proposed DewarpNet which can directly output the unwarped image, DocUNet needs several post-processing steps to convert the forward mapping to the backward mapping (each pixel represents the coordinates in the **warped input image**) and then sample the input image to get the unwarped result.

Results in Table 2 show significant improvement when we train DocUNet on Doc3D instead of the 2D synthetic dataset from [23]. The significant reduction of LD (14.08 to 10.85) signals a better local detail rectification. This improvement is the result of both (1) the DewarpNet architecture and (2) training with a more physically grounded Doc3D dataset, compared to the 2D synthetic dataset in [23].

5.3. Test DewarpNet on the DocUNet Benchmark

We evaluate both DewarpNet and DewarpNet(*ref*) (i.e., DewarpNet augmented with the post-processing refinement network) on the DocUNet Benchmark dataset. We provide comparisons on both (1) the overall benchmark dataset (Table 2) and (2) each class in the benchmark (Fig. 6). The latter provides detailed insight into the improvements of our approach over previous methods. From class (a) to (e), our model consistently improves MM-SSIM and LD over the previous state-of-the-art. In the most challenging class (f), where the images usually exhibit multiple crumples and random deformations, our method achieves comparable and slightly better results.

Time Efficiency of DewarpNet. Our model takes 32ms on average to process a 4K resolution image. Compared to DocUNet [23] this represents *a 125x speed up*. DewarpNet directly outputs the unwarped image whereas DocUNet requires an expensive separate post-processing step.

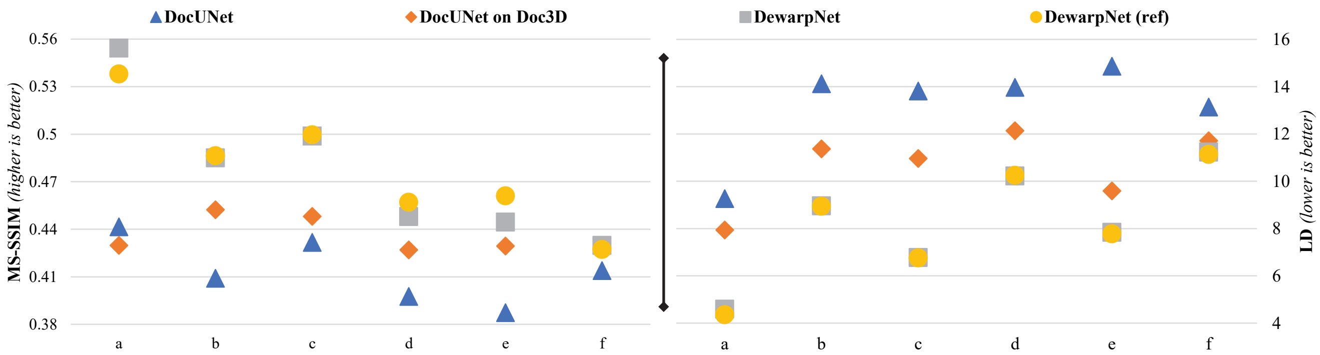


Figure 6. **Comparison of different methods on deformation classes.** We evaluate the results on: i) MS-SSIM (*higher is better*) and ii) LD (*lower is better*); Labels on x-axis correspond to the deformation classes (a)-(f) (as defined in Sec. 5.1).

Method	**MS-SSIM** ↑	**LD** ↓
DocUNet	0.41	14.08
DocUNet on Doc3D	0.4389	10.90
DewarpNet	0.4692	8.98
DewarpNet (*ref*)	**0.4735**	**8.95**

Table 2. Comparison of DewarpNet and DocUNet variants on the DocUNet benchmark, DewarpNet (*ref*) is DewarpNet combined with the refinement network.

Method	**ED** ↓	**CER** (*std*) ↓
Original Warped Image	2558.36	0.6178 (0.295)
DocUNet	1975.86	0.4656 (0.263)
DocUNet on Doc3D	1684.34	0.3955 (0.272)
DewarpNet	1288.60	0.3136 (0.248)
DewarpNet (*ref*)	**1114.40**	**0.2692** (0.234)

Table 3. OCR comparison between all methods.

Texture Mapping Net.	$\ell 2$ on $\hat{\mathbf{B}}$	SSIM on $\hat{\mathbf{D}}$
w/o CoordConv	4.73×10^{-5}	0.9260
CoordConv	$\mathbf{3.99 \times 10^{-5}}$	**0.9281**
$\mathcal{L}_B$	1.40×10^{-4}	0.8539
$\mathcal{L}_B + \mathcal{L}_D$	$\mathbf{3.99 \times 10^{-5}}$	**0.9281**

Table 4. Effects of CoordConv and using the $\mathcal{L}_D$ in the Texture Mapping Net

5.4. OCR Evaluation

We use PyTesseract (v0.2.6) [34] as the OCR engine to evaluate the utility of our work on text recognition from images. The text ground-truth (*reference*) is generated from 25 images from DocUNet [23]. In all these images, more than 90% of the content is text. The supplementary material contains some samples from our OCR test-set. OCR performance comparison, presented in Table 3, shows our method outperforms [23] with a large margin in all metrics. In particular, DewarpNet reduces CER by 33% compared to DocUNet, and the refinement network gives a reduction of 42%.

5.5. Ablation Studies

Coordinate Convolution (CoordConv). We investigate the effects of CoordConv on texture mapping network performance. The experiment (Table 4) on Doc3D validation set demonstrates that using CoordConv leads to a 16% $\ell 2$-error reduction on $\hat{\mathbf{B}}$ and a slight improvement of SSIM on $\hat{\mathbf{D}}$ from 0.9260 to 0.9281.

Loss $\mathcal{L}_D$. The texture mapping network benefits greatly from using $\mathcal{L}_D$ (unwarped visual quality loss). As shown in Table 4 compared to using the absolute pixel coordinate loss $\mathcal{L}_B$ only, using $\mathcal{L}_B + \mathcal{L}_D$ significantly reduces the $\ell 2$ error on $\hat{\mathbf{B}}$ by 71% and improve the SSIM on $\hat{\mathbf{D}}$ by 9%.

5.6. Qualitative Evaluation

For qualitative evaluation, we compare DewarpNet with DocUNet in Fig. 7 and You et al. [45] in Fig. 8. The method by [45] utilizes multi-view images to unwarp a deformed document. Even with a single image, DewarpNet shows competitive unwarping results.

Additionally, we show that the proposed method is robust to illumination variation and camera viewpoint changes in Fig. 9. To evaluate the illumination robustness, we test on multiple images with a fixed camera viewpoint but different directional lighting from front, back, left, right of the document, and environment lighting. We also test DewarpNet robustness to multiple camera viewpoints, on a sequence of multi-view images provided by [45]. Results show that DewarpNet yields almost the same unwarped image in all cases.

6. Conclusions and Future Work

In this work, we present DewarpNet, a novel deep learning architecture for document paper unwarping. Our

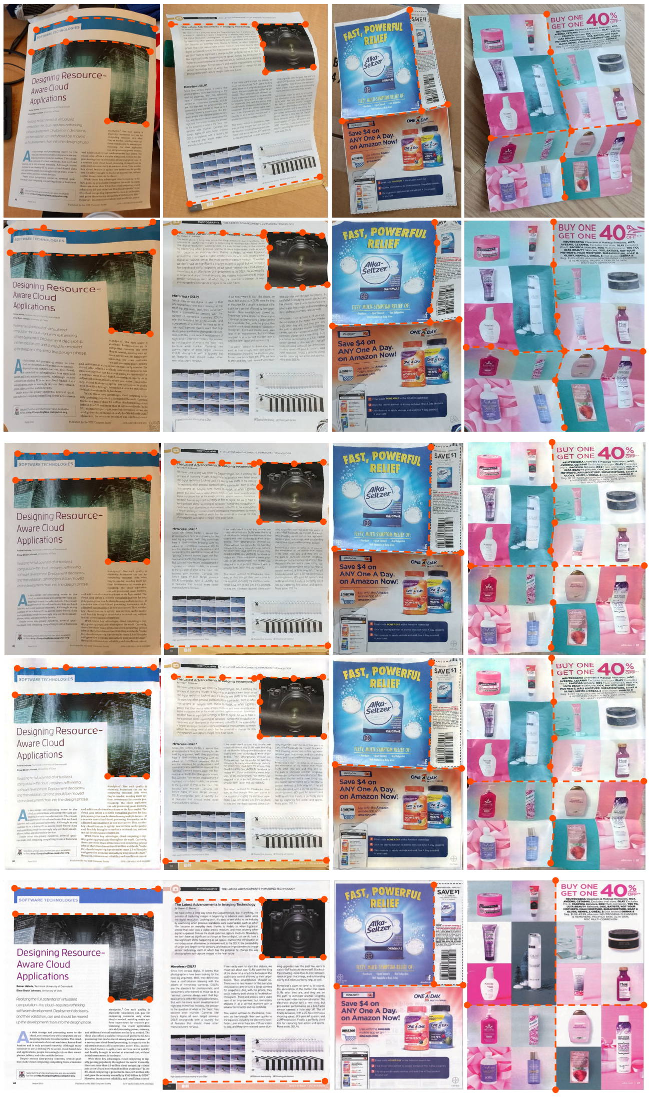


Figure 7. **Qualitative comparison of DewarpNet results on DocUNet** [23]. Row 1: Original warped images, Row 2: Results of [23], Row 3: Results of DewarpNet, Row 4: Results of DewarpNet after Shading Removal, Row 5: Flatbed scanned images. Red overlay markings show structural deformation.

method is robust to document content, lighting, shading, or background. Through the explicit modeling of 3D shape, DewarpNet shows superior performance over previous state-of-the-art. Additionally, we contribute the Doc3D dataset – the largest and most comprehensive dataset for document image unwarping, which comes with multiple 2D and 3D ground truth annotations.

Some limitations exist in our work: First, the inexpensive depth sensor cannot capture fine details of deformation like subtle creases on a paper crumple. Thus our data lacks samples with highly complex paper crumple. In future work, we plan to construct a dataset with better details and more complex structures. Second, DewarpNet is relatively sensitive to occlusion: results degrade when parts of the imaged document are occluded. In future work, we plan to address this difficulty via data augmentation and adversarial training.

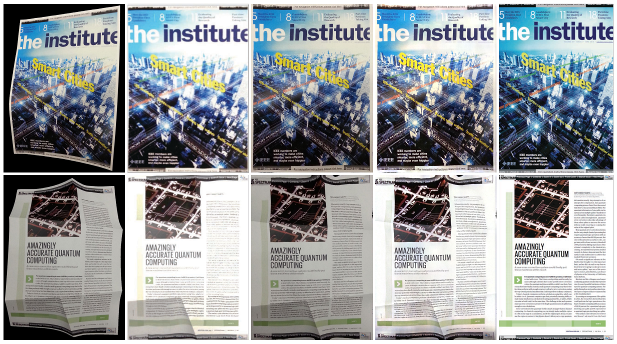


Figure 8. **Comparison with You et. al.** [45]. Columns from left to right: 1) Original warped images, 2) Results from [45], 3) DewarpNet, 4) DewarpNet results after shading removal, 5) Flatbed scanned images.

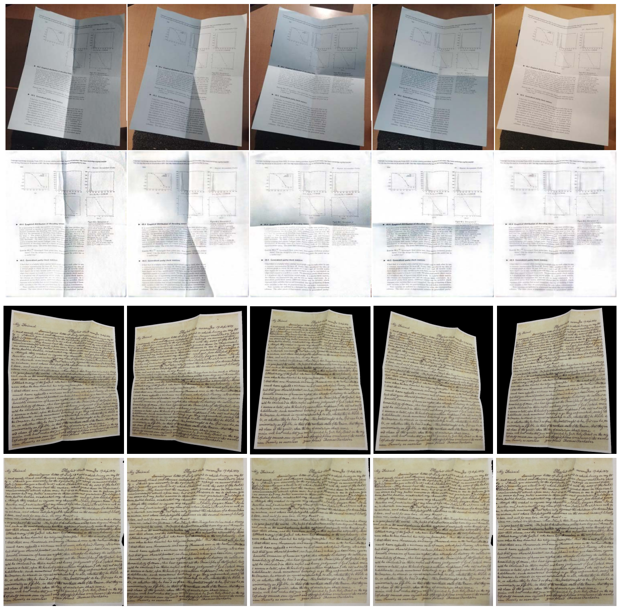


Figure 9. **DewarpNet robustness.** Top two rows : Robustness to lighting (results shown are after refinement step): Columns 1-4: Directional light on different sides of the document, i.e. right, left, top, bottom. Column 5: Environment light. Although the refinement network handles shading quite well, it is unable to remove the hard shadows. Bottom two rows: Robustness to camera viewpoint.

Acknowledgements. This research was supported by the SAMSUNG Global Research Outreach (GRO) Program, NSF grants CNS-1718014, IIS-1763981, a gift from the Nvidia Corporation, a gift from Adobe Research, the Partner University Fund and the SUNY2020 ITSC.

References

[1] Blender - a 3D modelling and rendering package. 3

[2] Harry Barrow, J Tenenbaum, A Hanson, and E Riseman. Recovering intrinsic scene characteristics. *Comput. Vis. Syst*, 2:3–26, 1978. 5

[3] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999. 3

[4] Michael S Brown and W Brent Seales. Document restoration using 3D shape: A general deskewing algorithm for arbitrarily warped documents. In *Proc. ICCV*. IEEE, 2001. 3

[5] Huaigu Cao, Xiaoqing Ding, and Changsong Liu. A cylindrical surface model to rectify the bound document image. In *Proc. ICCV*. IEEE, 2003. 2

[6] Barbara Caputo, Eric Hayman, and P Mallikarjuna. Class-specific material categorisation. In *Proc. ICCV*. IEEE, 2005. 5

[7] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proc. CVPR*. IEEE, 2014. 5

[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. 2

[9] Sagnik Das, Gaurav Mishra, Akshay Sudharshana, and Roy Shilkrot. The Common Fold: Utilizing the Four-Fold to Dewarp Printed Documents from a Single Image. In *Proceedings of the 2017 ACM Symposium on Document Engineering*, DocEng '17, pages 125–128. ACM, 2017. 1, 2

[10] Hironori Ezaki, Seiichi Uchida, Akira Asano, and Hiroaki Sakoe. Dewarping of document image by global optimization. In *Proc. ICDAR*. IEEE, 2005. 2

[11] David A Forsyth. Shape from texture and integrability. In *Proc. ICCV*. IEEE, 2001. 3

[12] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to Predict Indoor Illumination from a Single Image. *ACM Transactions on Graphics (SIGGRAPH Asia)*, 9(4), 2017. 4

[13] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A*, 4(4):629–642, Apr 1987. 4

[14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proc. CVPR*, number 2, 2017. 4

[15] D Kinga and J Ba Adam. A method for stochastic optimization. 2015. 6

[16] Hyung Il Koo, Jinho Kim, and Nam Ik Cho. Composition of a dewarped and enhanced document image from two view images. *IEEE Transactions on Image Processing*, 18(7):1551–1562, 2009. 2

[17] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, Feb 1966. 6

[18] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image. In *Proc. ECCV*, 2018. 4

[19] Jian Liang, Daniel DeMenthon, and David Doermann. Geometric rectification of camera-captured document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):591–605, 2008. 2

[20] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):978–994, 2011. 6

[21] Changsong Liu, Yu Zhang, Baokang Wang, and Xiaoqing Ding. Restoring camera-captured distorted document images. *International Journal on Document Analysis and Recognition*, 18(2):111–124, 2015. 3

[22] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *arXiv preprint arXiv:1807.03247*, 2018. 4, 6

[23] Ke Ma, Zhixin Shu, Xue Bai, Jue Wang, and Dimitris Samaras. DocUNet: Document Image Unwarping via A Stacked U-Net. In *Proc. CVPR*. IEEE, 2018. 2, 3, 6, 7, 8

[24] Jitendra Malik and Ruth Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *International Journal of Computer Vision*, 23(2):149–168, 1997. 3

[25] Gaofeng Meng, Chunhong Pan, Shiming Xiang, and Jiangyong Duan. Metric rectification of curved document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):707–722, 2012. 2

[26] Gaofeng Meng, Yuanqi Su, Ying Wu, Shiming Xiang, and Chunhong Pan. Exploiting Vector Fields for Geometric Rectification of Distorted Document Images. In *Proc. ECCV*, 2018. 1, 2

[27] Gaofeng Meng, Ying Wang, Shenquan Qu, Shiming Xiang, and Chunhong Pan. Active flattening of curved document images via two structured beams. In *Proc. CVPR*. IEEE, 2014. 2

[28] Rahul Narain, Tobias Pfaff, and James F. O'Brien. Folding and Crumpling Adaptive Sheets. *ACM Transactions on Graphics (TOG)*, 32(4):51:1–51:8, 2013. 3

[29] Jonas Östlund, Aydin Varol, Dat Tien Ngo, and Pascal Fua. Laplacian meshes for monocular 3D shape recovery. In *Proc. ECCV*. Springer, 2012. 3

[30] Albert Pumarola, Antonio Agudo, Lorenzo Porzi, Alberto Sanfeliu, Vincent Lepetit, and Francesc Moreno-Noguer. Geometry-Aware Network for Non-Rigid Shape Prediction from a Single View. In *Proc. CVPR*. IEEE, 2018. 3

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. MICCAI*. Springer, 2015. 4, 5

[32] Scott Schaefer, Travis McPhail, and Joe Warren. Image deformation using moving least squares. *ACM transactions on graphics (TOG)*, 25(3):533–540, 2006. 3

[33] Faisal Shafait and Thomas M Breuel. Document image dewarping contest. In *Workshop on Camera-Based Document Analysis and Recognition*, 2007. 2

[34] R. Smith. An Overview of the Tesseract OCR Engine. In *Proc. ICDAR*. IEEE, 2007. 7

[35] Yuandong Tian and Srinivasa G Narasimhan. Rectification and 3D reconstruction of curved document images. In *Proc. CVPR*. IEEE, 2011. 1, 3

[36] Yau-Chat Tsoi and Michael S Brown. Multi-view document rectification using boundary. In *Proc. CVPR*. IEEE, 2007. 2

[37] Yochay Tzur and Ayellet Tal. FlexiStickers: Photogrammetric texture mapping using casual images. In *Proc. ACM SIGGRAPH*. ACM, 2009. 4

[38] Adrian Ulges, Christoph H. Lampert, and Thomas Breuel. Document Capture Using Stereo Vision. In *Proceedings of the 2004 ACM Symposium on Document Engineering*, DocEng '04, pages 198–200. ACM, 2004. 3

[39] Adrian Ulges, Christoph H Lampert, and Thomas M Breuel. Document image dewarping using robust estimation of curled text lines. In *Proc. ICDAR*. IEEE, 2005. 3

[40] Eric Veach and Leonidas J. Guibas. Metropolis Light Transport. In *Proc. ACM SIGGRAPH*, 1997. 3

[41] Toshikazu Wada, Hiroyuki Ukida, and Takashi Matsuyama. Shape from shading with interreflections under a proximal light source: Distortion-free copying of an unfolded book. *International Journal of Computer Vision*, 24(2):125–135, 1997. 2

[42] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*. IEEE, 2003. 6

[43] Andrew P Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17(1-3):17–45, 1981. 3

[44] Atsushi Yamashita, Atsushi Kawarago, Toru Kaneko, and Kenjiro T Miura. Shape reconstruction and image restoration for non-flat surfaces of documents with a stereo vision system. In *Proc. ICPR*. IEEE, 2004. 2

[45] Shaodi You, Yasuyuki Matsushita, Sudipta Sinha, Yusuke Bou, and Katsushi Ikeuchi. Multiview Rectification of Folded Documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 1, 2, 3, 6, 7, 8

[46] Li Zhang, A. M. Yip, M. S. Brown, and Chew Lim Tan. A Unified Framework for Document Restoration Using Inpainting and Shape-from-shading. *Pattern Recognition*, 42(11):2961–2978, 2009. 1, 2

[47] Li Zhang, Yu Zhang, and Chew Tan. An improved physically-based method for geometric restoration of distorted document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):728–734, 2008. 1, 3