Learning an Isometric Surface Parameterization for Texture Unwrapping

Sagnik Das[1], Ke Ma[1,2], Zhixin Shu[3], and Dimitris Samaras[1]

[1] Stony Brook University, Stony Brook NY 11790, USA
{sadas}@cs.stonybrook.edu
[2] Snap Inc.
[3] Adobe Research

Abstract. In this paper, we present a novel approach to learn texture mapping for an isometrically deformed 3D surface and apply it for texture unwrapping of documents or other objects. Recent work on differentiable rendering techniques for implicit surfaces has shown high-quality 3D scene reconstruction and view synthesis results. However, these methods typically learn the appearance color as a function of the surface points and lack explicit surface parameterization. Thus they do not allow texture map extraction or texture editing. We propose an efficient method to learn surface parameterization by learning a continuous bijective mapping between 3D surface positions and 2D texture-space coordinates. Our surface parameterization network can be conveniently plugged into a differentiable rendering pipeline and trained using multi-view images and rendering loss. Using the learned parameterized implicit 3D surface we demonstrate state-of-the-art document-unwarping via texture extraction in both synthetic and real scenarios. We also show that our approach can reconstruct high-frequency textures for arbitrary objects. We further demonstrate the usefulness of our system by applying it to document and object texture editing. Code and related assets are available at: https://github.com/cvlab-stonybrook/Iso-UVField

Keywords: document unwarping, texture unwrapping, neural rendering

1 Introduction

Reconstructing 3D shapes from images is a core problem in computer vision and graphics research. With the progress in differentiable rendering [54,24,44,27,33], recent learning-based 3D reconstruction approaches have achieved impressive results using 2D supervision from a single image [11,20,12,39,61] or multi-view images [55,66]. These methods achieve high quality 3D reconstruction using differentiable rendering with various 3D representations such as 3D meshes [61], volumetric representations [40], or implicit functions [39]. In recent neural rendering methods such as NeRF [40] and IDR [66], continuous representations such as volume or implicit functions achieve significantly better reconstruction results than meshes or voxels because they do not discretize the 3D surface a

Fig. 1. The proposed forward-backward network can be utilized in unwrapping and editing a surface texture: the flattened texture can be edited and warped back to produce a texture edited image. In the top row we edit the unwrapped texture by overlaying a color grid. In the bottom row we edit the unwrapped texture by swapping the 'English' and 'Japanese' text. In the bottom row the desired texture mask is highlighted by a yellow dashed polygon. The warped texture is pasted at the masked region in different views.

priori. However, these continuous representations usually do not encode explicit surface parameterization, which would allow 3D shape re-texturing, editing the existing texture in the 2D texture space, or recovering 2D texture from 3D surfaces. One of the most direct applications of 2D texture unwrapping in a geometrically constrained manner, is document unwarping, i.e., the inference of a document's flatbed-scanned version from a casual photo of a potentially creased document. Moreover, 2D texture unwrapping could be equally valuable for other domains such as garments, common objects or faces. In this paper, we use the terms texture unwrapping and unwarping interchangeably.

Our novel texture mapping approach learns surface parameterization for isometrically deformed surfaces by learning continuous bijective functions between 3D surface positions and 2D texture-space coordinates. We use a signed distance function (SDF) [8] to represent the geometry and model the appearance as a function of the 2D texture coordinates. By utilizing implicit differentiable rendering (IDR), [66] we can reconstruct the 3D shape and learn the corresponding UV parameterization of the surface simultaneously. This is possible only with a per-pixel rendering loss and the appropriate geometric regularization.

We utilize two fully connected multi-layer perceptrons (MLPs) to learn a bijective mapping between 3D shapes and 2D texture space. More specifically, the *forward* MLP maps the 3D surface coordinates to 2D texture coordinates and the *backward* MLP maps the 2D texture coordinates to corresponding 3D surface coordinates. Following IDR [66], we obtain the 3D surface coordinates by sphere-tracing along the ray cast through each pixel. Our appearance rendering

is formulated as a function of the 3D and the texture coordinates. Therefore, the forward and backward MLPs can be trained with a 2D pixel-wise loss between the rendered image and the given ground truth image. To the best of our knowledge, this is the first neural rendering method that can learn a geometrically constrained UV parameterization for implicit surfaces.

Thus, our method is also the first method which utilizes implicit surface (signed distance function) based neural rendering for document unwarping. It is a challenging task due to the presence of geometric and photometric distortions in a document. For this particular problem we introduce a shape-specific texture mapping prior to initialize the forward MLP (3D to 2D mapping). This prior is learned from a large dataset of UV mapped document meshes, assuming that the document texture space maps to a 2D rectangle. This assumption regularizes the forward MLP to output a high-quality texture space that avoids degenerate solutions (see Fig. 3). Moreover, we introduce a conformality constraint in the backward MLP, which is consistent with how a paper folds in the physical world, i.e., without any stretch or tear. We can directly extend our method to work on rigidly deforming objects other than paper which follow similar physical properties such as fabric, soda cans etc. We also show that our method is robust to small deviations from the assumed conformality constraint, e.g. in the case of face texture unwrapping.

The main contributions of our paper are the following: 1) We propose an efficient way to learn a texture parameterization for implicit neural representations using a differentiable rendering framework. Without 3D supervision, it only requires multi-view images as ground truth and a texture mapping prior. 2) We show that our method can be effectively used for document unwarping tasks by learning a prior for explicit texture mapping on the document shape. We show that this prior can be learned from a dataset of texture-mapped meshes. Furthermore, this prior is also suitable for other objects sharing similar geometric property as papers. 3) We show that our method is effective for document image unwarping and texture editing (see Fig. 1). We achieve a 25% relative improvement over a publicly available state-of-the-art [13] in terms of mean local distortion across 750 views from fifteen synthetic scenes. Additionally, we achieve a $\sim$25% improvement in optical character recognition (OCR) in terms of character and word error rate. For the texture editing task, we show significant qualitative improvement over NeuTex [64].

2 Previous Work

Neural Rendering. Neural rendering generates images and videos by integrating conventional computer graphics rendering pipelines into deep neural networks [56]. It enables explicit or implicit control of scene properties, including illumination, geometry, texture, etc. Neural rendering can synthesize semantic photos [46,3], novel views [23,53], relighting [65,36], facial/body reenactment [7,63], estimate scene properties etc. Kato [24] proposed a differentiable neural renderer using an approximate gradient for rasterization. Liu [32]

proposed SoftRas, which extended differentiable rasterization. Li [27] further demonstrated the feasibility of integrating ray-tracing in deep neural networks. More recently, implicit surface or volume rendering has become mainstream in neural rendering approaches such as IDR [66] and NeRF [40]. These approaches are based on multi-view surface reconstruction to associate the scene geometry to the appearance in different views. NeRF is extended to lot of variants including PixelNeRF [68], MVSNeRF [10], dynamic NeRF [29,48], GRAF [51], etc.

Texture Mapping. Texture mapping is an essential step in the computer graphics rendering pipeline. It defines a correspondence between a vertex on the 3D mesh and a pixel in the 2D texture image. To find such a mapping, FlexiStickers [58] required users to specify a sparse set of correspondences. Bi [6] proposed a patch-based texture mapping method using the 3D shape and images from multiple views. Morreale [43] used networks to represent 3D surfaces/shapes. Apart from the above general texture mapping methods, some approaches focus on a specific object categories such as faces [16,9] and human bodies [42,69]. Recently, AtlasNet [20] represented a 3D mesh as a collection of parametric surfaces showing texture mapping is trivial to obtain from a 2D parametric surface. A similar idea was adopted by Bednarik [4] where they introduced geometric constraints when learning the decomposition. More recently, NeuTex [64] aims to recover the texture of a subject using NeRF [40]. However, NeuTex uses a spherical UV domain without any geometric constraints. Therefore, the recovered texture is not smooth which is not suitable for document unwarping. Moreover, since NeRF [40] doesn't learn an explicit geometry, NeuTex requires a coarse point-cloud to initialize the *backward* MLP. With an SDF based [66] rendering scheme, our approach does not require such an initialization routine. We can jointly learn the texture mapping and the geometry from scratch.

Document Unwarping. Document unwarping is a special application of texture mapping: the 3D object is usually a rectangular piecewise-developable surface and the texture is well structured, containing straight text lines, (usually) rectangular text blocks and figures etc. Previous work usually adopted a two-step methodology: 1) 3D surface estimation and 2) deformed surface flattening. The 3D surface of a deformed document can be estimated from shading [60], multi-view images [59], text lines [57], local character orientations [38], document boundaries [26], and learning-based strategies [47]. Flattening the obtained 3D surface always involves an expensive optimization process under certain geometry constraints such as conformality [67] or isometries [2]. Flattening could be easier if the obtained 3D shape had a low dimensional parameterization like Generalized Cylindrical Surface (GCS) [25]. Some studies [14,30,37] proposed to unwarp each patch on the surface individually and then stitch the unwarped patches together. In recent years, data-driven methods [34,13,28,35,15,17] have addressed document unwarping by leveraging large-scale synthetic datasets. These datasets contain deformed document images and their corresponding ground truth UV coordinates. Methods trained on synthetic images often suffer from generalization performance due to the domain gap between synthetic and real data. In this paper, we utilize neural rendering techniques to learn a surface parameter-

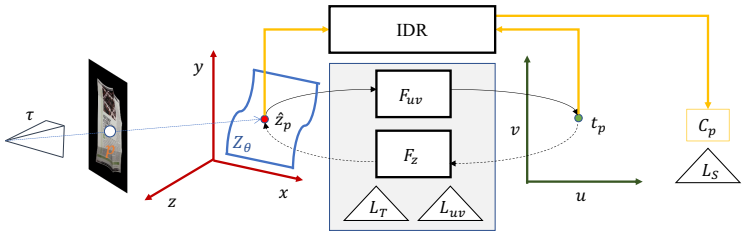

Fig. 2. Proposed surface parameterization learning using the forward (F_{uv}) and backward MLP (F_z): given camera pose τ, and a pixel p, we jointly learn the geometry represented by a SDF Z_θ, the F_{uv}, and the F_z. $\hat{z}_p$ is the ray-surface intersection point in 3D domain and t_p is the corresponding texture coordinate in UV domain. The yellow arrows denote the input and output of the IDR [66], and C_p is the predicted RGB color. Triangles denote the losses defined in Eq. 10.

ization of a deformed document. We simultaneously estimate both 3D shapes and UV coordinates with a cycle consistency loss and geometric constraints. By leveraging the information from multi-view images, the proposed method demonstrates better document unwarping performance compared to a previous state-of-the-art [13]. Furthermore, our method only needs multi-view images and corresponding foreground masks for training, eliminating the need of large-scale document image datasets with paired warping field ground truth.

3 Method

In Sec. 3.1 we first describe some preliminaries about surface parameterization and IDR. Then we describe the proposed approach that utilize a recent differentiable rendering method, IDR [66] for surface reconstruction and jointly learn the texture mapping of the learned implicit surface using two MLPs.

3.1 Preliminaries

Surface Parameterization. The problem of surface parameterization focuses on finding a bijective mapping F between a surface $Z \in \mathbb{R}^3$ and a polygonal domain $\Omega \in \mathbb{R}^n$. For a parametric or discrete surface representation, we can explicitly compute this mapping [58] using constrained optimization. In contrast, implicit surfaces are represented as continuous functions and cannot be readily parameterized. In this paper, we propose to learn such bijective mapping between a learned implicit surface and a 2D planar domain $\Omega \in \mathbb{R}^2$ using our proposed forward and backward MLPs. Ω is the texture space or UV space, parameterized using 2D UV coordinates $\mathbf{t} = (u, v)$. We can use any continuous parameterization function as the UV space. Since this work particularly focuses on document unwarping, we choose the UV space to be a regular 2D grid.

Implicit Differentiable Rendering. Implicit Differentiable Rendering [66] reconstructs the geometry of an object from multi-view images as the zero level

set, Z_θ of an MLP S,

$$Z_\theta = \{\mathbf{z} \in \mathbb{R}^3 \mid S(\mathbf{z}; \theta) = 0\} \tag{1}$$

where θ are the learnable parameters. To render the surface Z_θ, IDR uses another MLP to model the radiance (RGB color) as a function of the surface point ($\mathbf{z}_p$), corresponding surface normal ($\mathbf{n}_p$), view direction ($\mathbf{v}_p$) and a global geometry feature vector ($\mathbf{g}_p$):

$$C_p = A(\mathbf{z}_p, \mathbf{n}_p, \mathbf{v}_p, \mathbf{g}_p) \tag{2}$$

Here, C_p denotes the predicted color at pixel p and A denotes the appearance MLP. The surface point is obtained by a sphere-tracing method [22] along the ray $r_p(\tau)$ through pixel p. $\tau \in \mathbb{R}^k$ denotes camera parameters of the scene. Additionally, IDR also presents a differentiable way to obtain a ray and geometry intersection point ($\hat{\mathbf{z}}_p$) as a function of the camera ray. Although,the IDR can disentangle geometry and appearance, it only allows to re-render a new geometry with a learned appearance MLP, A. Editing a texture or extracting a surface texture map is not possible in a vanilla IDR framework.

3.2 Learning Surface Parameterization

To learn a meaningful parameterization of the implicit surface Z_θ, we represent the radiance at pixel p as a function of the UV space. To this end, we modify the IDR model (Eq. 2):

$$C_p = A_{uv}(\mathbf{t}_p, \mathbf{z}_p, \mathbf{n}_p, \mathbf{v}_p, \mathbf{g}_p) \tag{3}$$

The texture parameterized appearance MLP is modeled as a function of the texture coordinate $\mathbf{t}_p$ at surface point $\mathbf{z}_p$, corresponding to a pixel p. We can jointly train the surface MLP (S) and texture parameterized appearance MLP (A_{uv}) using a pixel-wise rendering loss between the predicted radiance (C_p) and ground truth radiance (C_p^{gt}) at pixel p. A schematic diagram of the proposed approach is shown in Fig. 2.

Forward and backward texture parameterization. We represent the mapping between the 3D surface and 2D texture space using the *forward* function F_{uv}: $\mathbf{z} \rightarrow \mathbf{t}$. The F_{uv} is modeled as an MLP. It is trained by mapping a ray-surface intersection point $\hat{\mathbf{z}}_p$ to its corresponding texture coordinate $\mathbf{t}_p$. p denotes the pixel location. Now to establish the bijective mapping (discussed in Sec. 3.1) between the surface and texture space we utilize a *backward* function F_z: $\mathbf{t} \rightarrow \mathbf{z}$. F_z is an MLP that learns an inverse mapping between the texture and the 3D space. It is trained by mapping a texture coordinate $\mathbf{t}_p$ to its corresponding ray-surface intersection point $\hat{\mathbf{z}}_p$.

Shape specific prior for F_{uv}. Jointly training the forward, backward and rendering network leads to the wrong UV mapping with local minima (see Fig. 3) where multiple $\hat{\mathbf{z}}_p$ map to a single texture coordinate. To avoid such

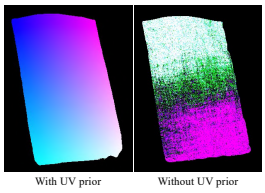

Fig. 3. Without a prior the forward network, F_{uv} leads to degenerate cases: multiple 3D points $\hat{z}_p$ are mapped to the same texture coordinate t_p.

degenerate cases, we initialize F_{uv} with a texture mapping prior, learned from a large dataset of UV mapped meshes. We assume the input shape to be a isometrically deformed quadrilateral and the corresponding UV space to be a regular grid ($\in [0.0, 1.0]$). The top leftmost and the bottom rightmost 3D coordinate of the shape maps to $(u, v) = (0, 0)$ and $(u, v) = (1, 1)$ respectively. To learn $\hat{F}_{uv}$ we utilize a collection of UV mapped meshes from the Doc3D [13] dataset and train an MLP with the same parameters as F_{uv}. For each scene, we use $\hat{F}_{uv}$ to initialize the weights of F_{uv} and train jointly with S and A_{uv}. Although this learned prior ($\hat{F}_{uv}$) is designed to learn a suitable texture mapping for document unwarping, we experimentally show this prior can be readily used for other domains as well.

Deformation constraints for $F_{\mathbf{z}}$. Conformal map [21] allows a 3D domain to be mapped to a texture domain with low distortion satisfying the bijective property between domains. We use a conformality constraint for F_z to ensure the deformation properties mentioned above. We define the conformality constraint in terms of the metric tensor, $\mathbf{J}^\top\mathbf{J}$ of the $F_{\mathbf{z}}$, where $\mathbf{J}$ is the Jacobian of $F_{\mathbf{z}}$ (Eq. 4):

$$\mathbf{J} = \begin{bmatrix} \frac{\delta F_{\mathbf{z}}}{\delta u} & \frac{\delta F_{\mathbf{z}}}{\delta v} \end{bmatrix} = [D_u \quad D_v] \qquad \mathbf{J}^\top\mathbf{J} = \begin{bmatrix} D_u^\top D_u & D_u^\top D_v \\ D_u^\top D_v & D_v^\top D_v \end{bmatrix} = \begin{bmatrix} E & F \\ F & G \end{bmatrix} \quad (4)$$

The conformality constraint is defined as $\mathbf{J}^\top\mathbf{J} = \beta\mathbf{I}$. Here β is a unknown local scaling function and $\mathbf{I}$ is the identity matrix. For developable surfaces which can be physically flattened without any stretch e.g. papers, β doesn't vary across the parameterization space. Therefore, we consider a fixed global scale ($[\beta_u, \beta_v]$) for the conformality constraint.

Unwarping by sampling $F_{\mathbf{z}}$. To unwarp the texture, we determine a foreground pixel at $p = (x, y)$ in the input image that should be projected to (u, v) in the unwarped image. Here the unwarped image refers to the texture space. Foreground pixel refers to a pixel within the pre-defined object mask. The coordinates (u, v) and p are associated by $F_{\mathbf{z}}$ and τ: for a (u, v) coordinate, its corresponding point in 3D is obtained by $\hat{z}'_p = F_{\mathbf{z}}(u, v)$. Given the camera parameter τ, $\hat{z}'_p$ is projected to p in the input image. Thus for each pixel in the unwarped texture, we can find its corresponding pixel in the input image which is all we need for unwarping (More details in supplemantary).

3.3 Loss Functions

We use the rendering losses on the predicted color, C_p, and predicted document mask M_p at pixel p to train the geometry S. Here $M_p \in \{0, 1\}$ refers to whether the pixel p is occupied ($M_p = 1$) by the shape or not ($M_p = 0$). We assume masks

are provided as input. Additionally, we employ appropriate regularization losses to jointly train S, A_{uv}, F_{uv} and F_z.

Loss for S. Following IDR [66], for each p we apply a sphere-tracing [22] algorithm followed by implicit differentiation to find the intersection point of the ray $r_p(\tau)$ and the surface Z_θ. Given the ground truth RGB color C_p^{gt} and the predicted RGB color C_p, the RGB loss is defined as:

$$L_{rgb} = \frac{1}{|P|} \sum_{p \in P_{in}} \left\| C_p^{gt} - C_p \right\|_1 \tag{5}$$

Where P is the set of pixels in the minibatch. The pixels $P_{in} \subset P$ for which ray-surface intersection has been found and $M_p = 1$. The mask loss is defined as:

$$L_{mask} = \frac{1}{\alpha |P|} \sum_{p \in P_{out}} CE(M_p^{gt}, M_p) \tag{6}$$

Here $P_{out} = P \setminus P_{in}$, α is a tunable parameter and $CE(.)$ is the cross-entropy loss. The value of $M_p = \mathcal{M}_{p,\alpha}(\theta, \tau)$ is a differentiable function of the learned Z_θ [66]. Additionally, to force Z_θ to be a approximate signed distance function we use Eikonal Regularization [19]:

$$L_{ek} = \mathbb{E}_z (\|\nabla_z S(\mathbf{z}; \theta)\| - 1)^2 \tag{7}$$

where z denotes uniformly sampled points within a bounding box of the 3D domain.

Loss for $F_{\mathbf{uv}}$. Although we initialize $F_{\mathbf{uv}}$ with learned prior parameters, we constrain the predicted 2D texture coordinates during training in order to avoid non-uniform mapping of the 3D and the UV domain which can squeeze or stretch the warped texture (example in supplementary). We employ a Chamfer distance between the $\mathbf{t}_p$ and uniformly sampled 2D points $\mathcal{T} \in [0, 1]$ to ensure $F_{\mathbf{uv}}$ approximately outputs $\mathcal{U} \sim [0, 1]$. This regularization term is defined as:

$$L_{uv} = CD_{p \in P_{in}}(\mathcal{T}, \mathbf{t}_p) \tag{8}$$

here $CD(.)$ denotes the Chamfer distance and t_p the predicted texture coordinates corresponding to ray-surface intersection points $\hat{\mathbf{z}}_p$.

Loss for $F_{\mathbf{z}}$. $\hat{z}'_p$ is the output of $F_{\mathbf{z}}$. $F_{\mathbf{z}}$ is trained with weighted regression loss between $\hat{z}_p$ and $\hat{z}'_p$:

$$L_z = \frac{1}{|P_{in}|} \sum_{p \in P_{in}} w_p (\hat{z}_p - \hat{z}'_p)^2 \tag{9}$$

w_p is a pre-calculated per-pixel weight based on the document mask (M) which assigns higher value to the pixels at the boundary of the document. (More weight calculation details in supplementary).

Additionally, to constrain $F_{\mathbf{z}}$ to be a fixed scale conformal mapping [4]. On the elements of the metric tensor, E, F and G defined in Eq. 4, we employ the following constraints:

$$L_E = \frac{1}{|P_{in}|} \sum_{p \in P_{in}} (E_p - \tilde{E})^2 \;\; L_G = \frac{1}{|P_{in}|} \sum_{p \in P_{in}} (G_p - \tilde{G})^2 \;\; L_F = \frac{1}{|P_{in}|} \sum_{p \in P_{in}} (F_p)^2$$

Here $\tilde{E}$ and $\tilde{G}$ is the mean of E and G. Our combined loss function is defined as:

$$L = \underbrace{(L_{rgb} + \gamma_1 L_{mask} + \gamma_2 L_{ek})}_{L_S} + \rho L_{uv} + \underbrace{(\delta_1 L_z + \delta_2 L_E + \delta_3 L_G + \delta_4 L_F)}_{L_T} \quad (10)$$

Here γ, ρ and δ denote the hyperparameters associated with the losses.

3.4 Training Details

The surface MLP $S(\mathbf{z}, \theta)$ consists of 8 layers with a hidden layer dimension of 128, with a skip connection to the middle layer [45]. The rendering network A_{uv} has 4 layers with hidden layer dimension of 512 and uses a sine activation function [52] at each layer. F_{uv} and F_z share identical architecture with 8 layers with 512 dimensional hidden units and sine activation [52]. Following NeRF [40], we use a k dimensional Fourier mapping ($\chi_k : \mathbb{R} \rightarrow \mathbb{R}^{2k}$) to learn high frequency details in the shape, RGB and the UV space. For S, A_{uv} we follow the setting of [66], and set $k = 6$ and $k = 4$ respectively. For F_{uv} and F_z we empirically set number of Fourier bands $k = 10$. We start with an initial learning rate of 10^{-5} and train for 80K iterations by halving the learning rate twice at 16K and 24K iterations. Initially, α is set to 50 and doubled during the training at 4K, 6K and 8K iterations. We set $\gamma_1 = 100.0$, $\gamma_2 = 0.1$ and $\rho = 0.001$. δ_1 is set to 0.001 for the initial 25K iterations. Afterward, δ_1 is multiplied by a factor 3 at every 10K iterations for a maximum of 7 times. δ_2, δ_3 and δ_4, are set to zero for the initial 50K iterations. Only L_z is sufficient to achieve a good texture to 3D mapping during the shape optimization phase. Afterwards we set $\delta_2 = \delta_3 = 0.001$ and $\delta_4 = 0.01$. The metric tensor calculation is implemented using auto-differentiation.

Initializing S and F_z. We can start optimizing S from the standard IDR initialization (SDF of a sphere). However, we notice that a better initialization can significantly improve the training time as well as the quality of the shape reconstruction. For object specific application like document unwarping we found that initializing S with a similar object can significantly reduce the training time and converges in a half number of iterations. Furthermore, we also found that initializing F_z to produce a planar point-cloud can further reduce our training convergence time. To this end, we pre-train the F_z to produce a plane. More details are discussed in supplementary.

4 Experimental Results

First, we quantitatively compare the proposed method with a state-of-the-art document unwarping method DewarpNet [13]. Our quantitative and qualitative experiments are performed on 15 synthetic and 10 real documents. Second, we apply our method to texture editing for documents and other objects such as soda can, t-shirt, and human face. Last, we conduct ablation studies to demonstrate the effectiveness of our proposed loss functions (reported in supplementary due to space constraints).

4.1 Evaluation Dataset and Metrics

For document unwarping, the synthetic evaluation data consists of 15 scenes rendered using Blender [1] following a rendering pipeline similar to Doc3D. Each scene consists of 50 random views sampled from a 45^o solid angle in the upper hemisphere. The real-world evaluation data consists of five scenes from the dataset of [67] and nine scenes captured by us. All the synthetic data and some of the real data include the document scan as the unwarping ground truth which are used for quantitative evaluation. Apart from documents we use 4 real objects for qualitative comparison. Each scene consists of 5-20 images per scene. We manually annotate the masks for each scene. To obtain camera poses for the real-world data, we use COLMAP [50]. We should note that for objects such as soda can, t-shirts, and faces, we assume a consistent foreground mask is available for all the views, designating the part of the texture to be unwrapped. For these objects, we also use the same F_{uv} learned for the document unwarping task. The learned prior from Doc3D dataset is usable as long as the surface somewhat follows the rectangular shape assumption.

We use image-based evaluation metrics for quantitative evaluation, including Local Distortion (LD) and Multi-Scale Structural Similarity (MS-SSIM). These are standard metrics used for document unwarping evaluation [13,34]. LD is based on dense SIFT flow [31] between the unwarped and scanned images. Image similarity metric MS-SSIM [62] is based on local image statistics (mean and variance) of the unwarped and scanned (ground truth) images calculated over multiple Gaussian pyramid scales. We use the same settings as [13,34] for fair comparison.

4.2 Document Unwarping

The quantitative comparison with the state-of-the-art model [13] is shown in Table 1 for the synthetic and real scenes. In terms of average performance of all the views (*all views* col. in Table 1) we improve the LD by $\sim 45\%$ compared to [13]. Since we use multi-view images for training, our results are more consistent across all the views compared to DewarpNet, which is also a key reason for the significant improvement.

We also report in a more practical evaluation scenario (*frontal view* column of Table 1) where we compare our results with DewarpNet for a frontal view

Methods	Synthetic				Real			
	All views		*Frontal view*		*All views*		*Frontal view*	
	MSSIM↑	**LD↓**	**MSSIM ↑**	**LD ↓**	**MSSIM ↑**	**LD ↓**	**MSSIM ↑**	**LD ↓**
DewarpNet	0.5382	7.81	0.5965	5.37	0.4601	10.25	0.4724	7.85
Proposed	**0.6302**	**4.31**	**0.6405**	**4.02**	**0.4951**	**7.16**	**0.494**	**6.28**

Table 1. Quantitative comparison of DewarpNet [13] and proposed method on synthetic and real images. *All views* report the mean result of all the views across all scenes and *Frontal view* denotes the mean result of one frontal view from each scene. Frontal view can be considered as the easiest or most probable view among all the views.

unwarping of the document. This setting also shows 25% relative improvement of LD compared to DewarpNet due to the stability of the method across different views. Since DewarpNet is trained on a synthetic dataset, it's generalizability limitation is reflected through this experiment.The choice of the best unwarped result is often subjective. We conjecture that since [13] is a single image unwarping method, it should perform well on simpler deformations and frontal view images. However, it is not always the case. Qualitative comparisons on real images in Fig. 4 show DewarpNet often generates artifacts even for reasonably frontal views and simple deformations. Comparatively, our results are qualitatively superior. Similar trend is observed in synthetic scenes. We qualitatively compare the frontal view results of DewarpNet with our results across 6 scenes in Fig. 5. In Fig. 5 our results are clearly better than the DewarpNet in all cases, with straighter lines and better rectified structure. More qualitative comparisons are available in supplementary material.

The quantitative comparison for real scenes are reported in Table 1 (right). We achieve significantly better LD than DewarpNet on both the frontal view evaluation and when averaged across all views. However, we notice that the improvement in terms of MS-SSIM is not that prominent due to its sensitivity to subtle perceptually unimportant global transformations such as translation by few pixels. We also note that quantitative scores are comparatively worse for the real scenes than synthetic scenes due to the fewer available views (10-15 compared to 50). Moreover, in absence of sufficient texture and views our method may result in unsatisfactory unwarping results. Such data are a failure case of IDR since there is insufficient information to reconstruct the 3D shape. As a result of the poor 3D shape, our texture parameterization network produces an inferior unwarping result (More details are available in supplementary). We also report qualitative comparisons with [67] and [13] on additional real documents in supplementary.

OCR Evaluation. We evaluated the OCR performance on 5 real scenes across 77 images in Table 2. We use Edit Distance (ED) [41], Character Error Rate (CER) and Word Error Rate (WER) as our evaluation metrics. ED is defined as the total number of substitutions (s), insertions (i) and deletions (d)

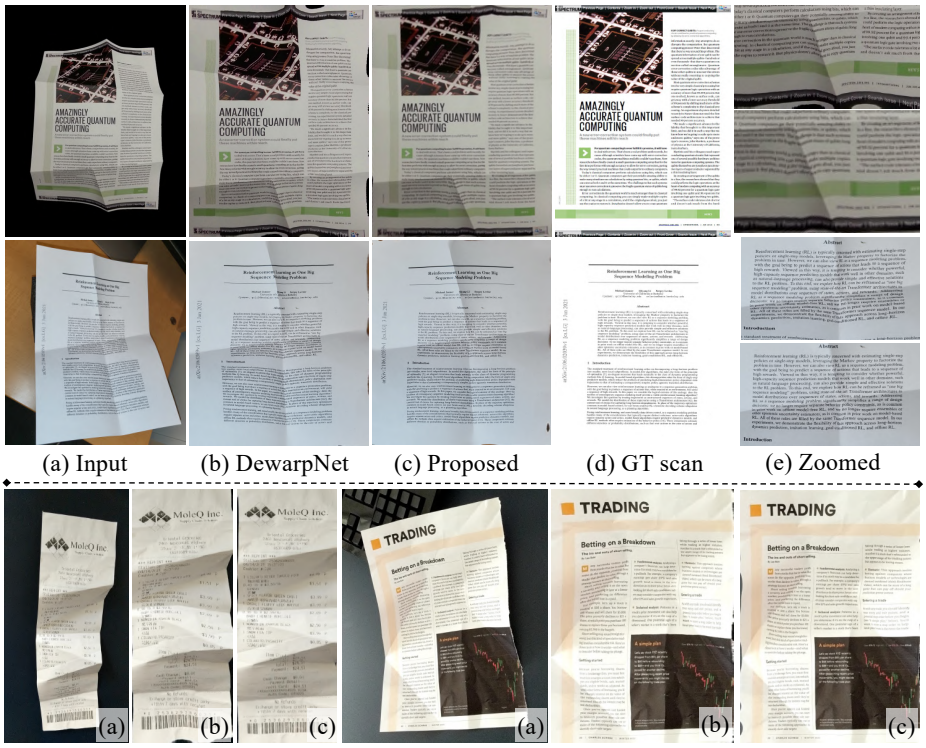

Fig. 4. Qualitative comparison with DewarpNet [13] on real images: (a) Input image, (b) DewarpNet unwarping, (c) Proposed unwarping, (d) GT scanned image, (e) enlarged regions: DewarpNet (*top*), and proposed (*bottom*). We use reasonable frontal view of the document for a fair comparison.

required to obtain the reference text, given the recognized text. The reference text is obtained by running the OCR algorithm on the scanned ground truth image of each document. CER is defined as: $(s + i + d)/N$ where N is the number of characters in the reference text. We use Tesseract 4.1.1 based LSTM OCR engine for this experiment. Our unwarped results reduce the ED, CER and WER by ∼25%. This improvement proves our unwarped results are more suitable for downstream applications like OCR.

Methods	ED↓	CER (std)↓	WER (std)↓
DewarpNet	798.30	0.2827 (0.12)	0.4646 (0.17)
Proposed	**600.78**	**0.2122** (**0.10**)	**0.3568** (**0.11**)

Table 2. Comparison of OCR metrics: We improve the OCR performance of [13] by ∼25% in terms of Edit Distance (ED), Character (CER), and Word Error Rate (WER).

4.3 Texture Editing

In addition to document unwarping, our proposed forward and backward MLP

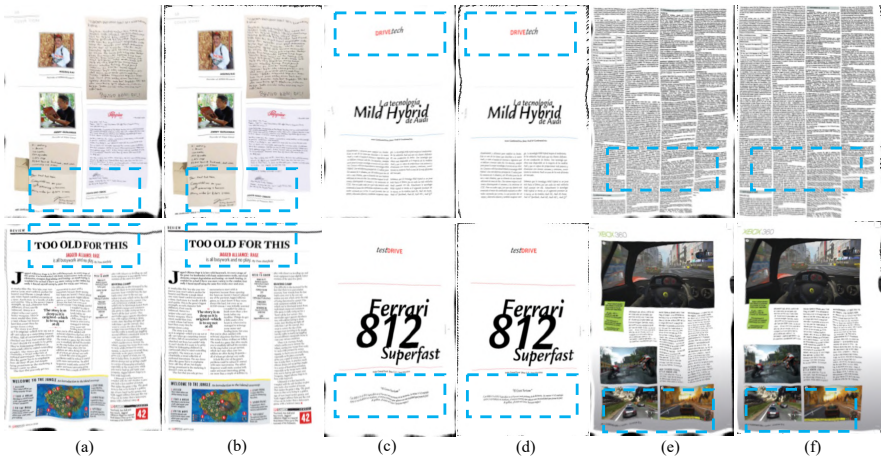

Fig. 5. Frontal view unwarping comparison of DewarpNet (a,c,e) and the proposed method (b,d,f) on synthetic images. Proposed results are clearly better with straighter lines. Follow the blue dashed boxes for discrimitative regions.

can also be used for high quality texture editing. We show texture editing examples in Fig. 1 and Fig. 7. We use the backward MLP to unwarp the texture from the input image, then edit the texture and warp it back to image space using the learned forward MLP (More details in supplementary). The proposed method can unwarp any isometrically deformed surface such as fabric or metal. It also works quite well when deformation is not exactly isometric, e.g., human faces [49]. In Fig. 6, we compare our texture editing results with NeuTex [64], a NeRF based texture unwrapping method. Compared to NeuTex [64], our results contain better details due to the forward prior and the geometric constraints.

Fig. 6. Comparison with NeuTex [64]: a prior work that aims to recover texture in a NeRF based setting, fails to recover high frequency details of the texture. Comparatively, our method clearly does a better job since we directly sample the texture from the input image.

5 Limitations and Future Work

Our proposed method for a scene can be trained in approximately 6 hours for 448×448 resolution images using a single Titan Xp GPU. The current training time per scene is high compared to DewarpNet's inference time which makes it unsuitable for real-time applications. However, we would like to note that in the current implementation sphere-tracing takes almost 50-60% of the running time. With a faster version of the sphere-tracing we can readily achieve a faster

(a) Input and mask (b) Edited texture (c) Different views of the edited texture

Fig. 7. Examples of texture editing non-document surfaces. This demonstrates our method is flexible beyond document unwarping and can be seamlessly used for other domains [49] as long as the isometric assumption is not strongly violated. The foreground mask is shown using a yellow dashed polygon.

framework. Moreover, neural rendering is an active research field and there are multiple other works that are focusing on improving the speed and generalization abilities [18,5]. Therefore, a faster training can be achieved following any newer or faster alternatives of IDR.

In this paper, we successfully applied our method on some toy objects other than documents. However, application of our method is limited by the isometric deformation assumption. For more complex UV spaces (e.g., texture atlas), learning the prior may require decomposing the shape to multiple simple UV maps where each UV map follow the isometric assumption. The proper way to do this is beyond the scope of this paper, however we believe it's an exciting future work. Another strong assumption of our method is the learned $\hat{F}_{uv}$ prior assumes the texture to be a continuous mapping bounded in a quadrilateral. This constraint suit the rectangular paper shape and improve empirical results in a specific task. More general objects will require different constraints e.g., spherical UV domain, local scaling of the conformal map etc.

6 Conclusions

We have introduced a neural rendering based architecture that can learn texture parameterized 3D shapes from multi-view images. This is the first work to learn surface parameterization of an implicit neural representation, to the best of our knowledge. We have demonstrated the applicability of our approach on multiple synthetic and real scenes for the task of document unwarping and object texture editing. We achieve state-of-the-art texture unwrapping and editing results.

Acknowledgements. This work was done when Ke Ma was at Stony Brook University. This work was partially supported by the Partner University Fund, the SUNY2020 ITSC, the FRA project "Deep Learning for Large-Scale Rail Defect Inspection" and gifts from Adobe and Amazon.

References

1. Blender - a 3D modelling and rendering package
2. Bartoli, A., Gerard, Y., Chadebecq, F., Collins, T., Pizarro, D.: Shape-from-template. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(10), 2099–2118 (2015)
3. Bau, D., Strobelt, H., Peebles, W., Wulff, J., Zhou, B., Zhu, J.Y., Torralba, A.: Semantic photo manipulation with a generative image prior. ACM Transactions on Graphics (TOG) **38**(4) (2019)
4. Bednarik, J., Parashar, S., Gundogdu, E., Salzmann, M., Fua, P.: Shape reconstruction by learning differentiable surface representations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
5. Bergman, A.W., Kellnhofer, P., Wetzstein, G.: Fast training of neural lumigraph representations using meta learning (2021)
6. Bi, S., Kalantari, N.K., Ramamoorthi, R.: Patch-based optimization for image-based texture mapping. ACM Transactions on Graphics (TOG) **36**(4), 106–1 (2017)
7. Chan, C., Ginosar, S., Zhou, T., Efros, A.A.: Everybody dance now. In: Proceedings of the International Conference on Computer Vision (2019)
8. Chan, T., Zhu, W.: Level set based shape prior segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE (2005)
9. Chen, A., Chen, Z., Zhang, G., Mitchell, K., Yu, J.: Photo-realistic facial details synthesis from single image. In: Proceedings of the International Conference on Computer Vision (2019)
10. Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H.: MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. arXiv preprint arXiv:2103.15595 (2021)
11. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5939–5948 (2019)
12. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: European conference on computer vision. pp. 628–644. Springer (2016)
13. Das, S., Ma, K., Shu, Z., Samaras, D., Shilkrot, R.: DewarpNet: Single-image document unwarping with stacked 3D and 2D regression networks. In: Proceedings of the International Conference on Computer Vision (2019)
14. Das, S., Mishra, G., Sudharshana, A., Shilkrot, R.: 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. pp. 125–128. DocEng '17, Association for Computing Machinery (2017). https://doi.org/10.1145/3103010.3121030
15. Das, S., Singh, K.Y., Wu, J., Bas, E., Mahadevan, V., Bhotika, R., Samaras, D.: End-to-end piece-wise unwarping of document images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4268–4277 (2021)
16. Deng, J., Cheng, S., Xue, N., Zhou, Y., Zafeiriou, S.: Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
17. Feng, H., Wang, Y., Zhou, W., Deng, J., Li, H.: Doctr: Document image transformer for geometric unwarping and illumination correction. arXiv preprint arXiv:2110.12942 (2021)

18. Garbin, S.J., Kowalski, M., Johnson, M., Shotton, J., Valentin, J.: Fastnerf: High-fidelity neural rendering at 200fps (2021)
19. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit geometric regularization for learning shapes. arXiv preprint arXiv:2002.10099 (2020)
20. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3d surface generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
21. Haker, S., Angenent, S., Tannenbaum, A., Kikinis, R., Sapiro, G., Halle, M.: Conformal surface parameterization for texture mapping. IEEE Transactions on Visualization and Computer Graphics **6**(2), 181–189 (2000)
22. Hart, J.C.: Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. The Visual Computer **12**(10), 527–545 (1996)
23. Hedman, P., Philip, J., Price, T., Frahm, J.M., Drettakis, G., Brostow, G.: Deep blending for free-viewpoint image-based rendering. ACM Transactions on Graphics (TOG) **37**(6), 1–15 (2018)
24. Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
25. Kil, T., Seo, W., Koo, H.I., Cho, N.I.: Robust Document Image Dewarping Method Using Text-Lines and Line Segments. In: Proceedings of the International Conference on Document Analysis and Recognition. pp. 865–870. IEEE, Institute of Electrical and Electronics Engineers (2017)
26. Koo, H.I., Kim, J., Cho, N.I.: Composition of a dewarped and enhanced document image from two view images. IEEE Transactions on Image Processing **18**(7), 1551–1562 (2009)
27. Li, T.M., Aittala, M., Durand, F., Lehtinen, J.: Differentiable monte carlo ray tracing through edge sampling. ACM Transactions on Graphics (TOG) **37**(6), 1–11 (2018)
28. Li, X., Zhang, B., Liao, J., Sander, P.V.: Document Rectification and Illumination Correction using a Patch-based CNN. ACM Transactions on Graphics (TOG) (2019)
29. Li, Z., Niklaus, S., Snavely, N., Wang, O.: Neural scene flow fields for space-time view synthesis of dynamic scenes. arXiv preprint arXiv:2011.13084 (2020)
30. Liang, J., DeMenthon, D., Doermann, D.: Geometric rectification of camera-captured document images. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(4), 591–605 (2008)
31. Liu, C., Yuen, J., Torralba, A.: Sift flow: Dense correspondence across scenes and its applications. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(5), 978–994 (2011)
32. Liu, S., Li, T., Chen, W., Li, H.: Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In: Proceedings of the International Conference on Computer Vision (2019)
33. Liu, S., Saito, S., Chen, W., Li, H.: Learning to infer implicit surfaces without 3d supervision. arXiv preprint arXiv:1911.00767 (2019)
34. Ma, K., Shu, Z., Bai, X., Wang, J., Samaras, D.: DocUNet: Document Image Unwarping via A Stacked U-Net. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Institute of Electrical and Electronics Engineers (2018)
35. Markovitz, A., Lavi, I., Perel, O., Mazor, S., Litman, R.: Can you read me now? Content aware rectification using angle supervision. In: Proceedings of the European Conference on Computer Vision. Springer (2020)

36. Meka, A., Haene, C., Pandey, R., Zollhöfer, M., Fanello, S., Fyffe, G., Kowdle, A., Yu, X., Busch, J., Dourgarian, J., et al.: Deep reflectance fields: High-quality facial reflectance field inference from color gradient illumination. ACM Transactions on Graphics (TOG) **38**(4), 1–12 (2019)
37. Meng, G., Huang, Z., Song, Y., Xiang, S., Pan, C.: Extraction of virtual baselines from distorted document images using curvilinear projection. In: Proceedings of the International Conference on Computer Vision (2015)
38. Meng, G., Su, Y., Wu, Y., Xiang, S., Pan, C.: Exploiting Vector Fields for Geometric Rectification of Distorted Document Images. In: Proceedings of the European Conference on Computer Vision (2018)
39. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4460–4470 (2019)
40. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing scenes as neural radiance fields for view synthesis. In: Proceedings of the European Conference on Computer Vision (2020)
41. Miller, F.P., Vandome, A.F., McBrewster, J.: Levenshtein Distance: Information Theory, Computer Science, String (Computer Science), String Metric, Damerau?Levenshtein Distance, Spell Checker, Hamming Distance. Alpha Press (2009)
42. Mir, A., Alldieck, T., Pons-Moll, G.: Learning to transfer texture from clothing images to 3d humans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
43. Morreale, L., Aigerman, N., Kim, V., Mitra, N.J.: Neural surface maps (2021)
44. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3504–3515 (2020)
45. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 165–174 (2019)
46. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
47. Pumarola, A., Agudo, A., Porzi, L., Sanfeliu, A., Lepetit, V., Moreno-Noguer, F.: Geometry-Aware Network for Non-Rigid Shape Prediction from a Single View. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Institute of Electrical and Electronics Engineers (2018)
48. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-nerf: Neural radiance fields for dynamic scenes. arXiv preprint arXiv:2011.13961 (2020)
49. Ramon, E., Triginer, G., Escur, J., Pumarola, A., Garcia, J., Giro-i Nieto, X., Moreno-Noguer, F.: H3d-net: Few-shot high-fidelity 3d head reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5620–5629 (2021)
50. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
51. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: Graf: Generative radiance fields for 3d-aware image synthesis. In: Advances in Neural Information Processing Systems (2020)

52. Sitzmann, V., Martel, J.N., Bergman, A.W., Lindell, D.B., Wetzstein, G.: Implicit neural representations with periodic activation functions. In: arXiv (2020)
53. Sitzmann, V., Thies, J., Heide, F., Nießner, M., Wetzstein, G., Zollhofer, M.: Deepvoxels: Learning persistent 3d feature embeddings. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
54. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. arXiv preprint arXiv:1906.01618 (2019)
55. Tang, C., Tan, P.: Ba-net: Dense bundle adjustment network. arXiv preprint arXiv:1806.04807 (2018)
56. Tewari, A., Fried, O., Thies, J., Sitzmann, V., Lombardi, S., Sunkavalli, K., Martin-Brualla, R., Simon, T., Saragih, J., Nießner, M., et al.: State of the art on neural rendering. In: Computer Graphics Forum. vol. 39, pp. 701–727. Wiley Online Library (2020)
57. Tian, Y., Narasimhan, S.G.: Rectification and 3D reconstruction of curved document images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Institute of Electrical and Electronics Engineers (2011)
58. Tzur, Y., Tal, A.: FlexiStickers: Photogrammetric texture mapping using casual images. In: Proceedings of the ACM SIGGRAPH Conference on Computer Graphics. Association for Computing Machinery (2009)
59. Ulges, A., Lampert, C.H., Breuel, T.: Document Capture Using Stereo Vision. In: Proceedings of the 2004 ACM Symposium on Document Engineering. pp. 198–200. DocEng '04, Association for Computing Machinery (2004). https://doi.org/10.1145/1030397.1030434
60. Wada, T., Ukida, H., Matsuyama, T.: 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)
61. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 52–67 (2018)
62. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thirty-Seventh Asilomar Conference on Signals, Systems and Computers. Institute of Electrical and Electronics Engineers (2003)
63. Wei, S.E., Saragih, J., Simon, T., Harley, A.W., Lombardi, S., Perdoch, M., Hypes, A., Wang, D., Badino, H., Sheikh, Y.: Vr facial animation via multiview image translation. ACM Transactions on Graphics (TOG) **38**(4), 1–16 (2019)
64. Xiang, F., Xu, Z., Hašan, M., Hold-Geoffroy, Y., Sunkavalli, K., Su, H.: NeuTex: Neural texture mapping for volumetric neural rendering. arXiv preprint arXiv:2103.00762 (2021)
65. Xu, Z., Sunkavalli, K., Hadap, S., Ramamoorthi, R.: Deep image-based relighting from optimal sparse samples. ACM Transactions on Graphics (TOG) **37**(4), 1–13 (2018)
66. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., Lipman, Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. Advances in Neural Information Processing Systems **33** (2020)
67. You, S., Matsushita, Y., Sinha, S., Bou, Y., Ikeuchi, K.: Multiview Rectification of Folded Documents. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
68. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelNeRF: Neural radiance fields from one or few images. arXiv preprint arXiv:2012.02190 (2020)

69. Zhao, F., Liao, S., Zhang, K., Shao, L.: Human parsing based texture transfer from single image to 3D human via cross-view consistency. In: Advances in Neural Information Processing Systems (2020)