Learning Shape-from-Shading for Deformable Surfaces

Jan Bednařík Pascal Fua Mathieu Salzmann

EPFL, CVLab
Lausanne, Switzerland

Abstract. Recent years have seen the development of mature solutions for reconstructing deformable surfaces from a single image, provided that they are relatively well-textured. By contrast, recovering the 3D shape of texture-less surfaces remains an open problem, and essentially relates to Shape-from-Shading. In this paper, we introduce a data-driven approach to this problem. We introduce a general framework that can predict diverse 3D representations, such as meshes, normals, and depth maps. Our experiments show that, in contrast with the well-textured scenario, meshes a ill-suited to handle texture-less 3D reconstruction. Furthermore, we demonstrate that our approach generalizes well to unseen objects, and that it yields higher-quality reconstructions than a state-of-the-art SfS technique, particularly in terms of normal estimates. Our reconstructions accurately model the fine details of the surfaces, such as the creases of a T-Shirt worn by a person.

Keywords: Deformable surfaces, 3D reconstruction, Shape-from-Shading, Deep Learning

1 Introduction

Existing approaches to monocular reconstruction of 3D deformable surfaces are designed either for well-textured surfaces [1,2], or, less frequently, partially-textured ones [3,4]. Furthermore, most of them have only been demonstrated on relatively smooth surfaces. In this paper, we tackle the problem of recovering the shape of complex texture-less surfaces from a single image, which is close in spirit to Shape-from-shading (SfS) with the added difficulty that we must handle complex phenomena such as sharp-creases and self-shadowing. The T-shirt of Fig. 1 being worn by someone who moves illustrates that.

SfS is one of the oldest Computer Vision problems [5,6,7]. Yet to this day, it remains largely unsolved because it is such an ill-posed inverse problem, except in tightly controlled lighting environments [8]. The early methods were variatonal ones that required very strong assumptions about the world, such as the presence of a single light source together with simple reflectance properties of the surfaces to be reconstructed, which are rarely satisfied. Recent ones [9,10,11] have focused on replacing some of these assumptions by measurements of the surface and

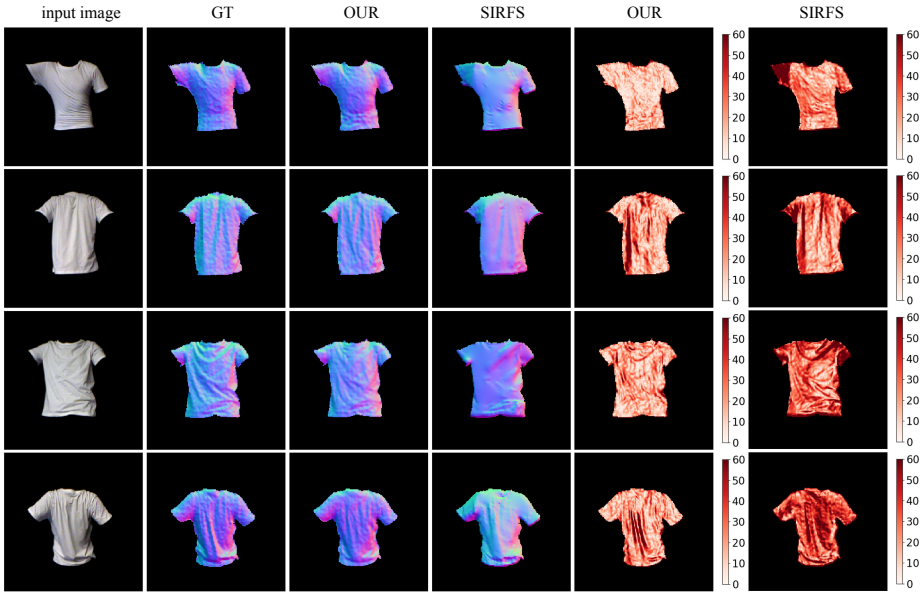

Fig. 1. Depth recovery of T-Shirt Comparison or our method to that of [9] on a deforming T-Shirt. The second to fourth columns depict the ground truth and recovered normals in terms of colors. The last two columns show differences in degrees between the recovered normals and the ground truth ones. Note that the discrepancies are much smaller in our case and that the sharp creases are better recovered.

lighting properties, yet still rely on relatively simple geometric and photometric models to remain computationally tractable.

In this paper, we show that a data driven approach enables us to operate under much weaker assumptions that *are* sufficiently well satisfied in everyday life to make the method truly practical in an environment where the lighting can be complex and inter-reflections, shadows, and sharp creases are prevalent. Fig. 1 depicts such a situation in which we outperform one the best currently available algorithms [9]. It would seem natural to follow the most popular trend in modeling deformable surfaces and to train a Deep Net to regress from the image to the shape parameters of a surface mesh, as was done for well-textured surfaces in [12]. We will demonstrate, however, that this is not the best. It is more effective to train a network that predicts a dense map or depths, normals, or both, as was done by the SfS pioneers [5].

Because we do not constrain the surface to be smooth and allow the network to learn about complex effects such as self-shadowing and occlusions, we can recover very severe deformations such as the sharp folds that can be seen in Fig. 1. Furthermore, as evidenced by our experiments, training on a single surface allows us to generalize to other ones of different shape and material properties without any re-training or fine-tuning. Our contribution is therefore an approach to shape-from-shading that can recover much more complex deformations than

earlier ones under realistic lighting conditions. Furthermore, we captured a large dataset to perform our experiments, which we will make publicly available.

2 Related Work

Traditionally, the SfS problem has been posed as a variational problem involving the optimization of physically-inspired objective functions to impose brightness, smoothness, and integrability constraints. In its original form [5,6,7], the problem is underconstrained and its solution plagued by ambiguities [13,14], which can be formally resolved only in very specific cases, such as when the camera and light source are co-located [8] or when additional stereo information is available [15].

Known lighting, and absence of interreflections and cast shadows are often assumed, as in the approach of [16] to jointly recover albedo and shape so as to explain the image as well as possible. In [17,18], while known, the lighting is assumed to be natural, which makes it possible to treat the thee color channels of the image in a manner similar to that of photometric stereo. Assumptions are also often made about the shape. For example, in [10], quadratic functions are fitted to local image patches of different sizes, which allows the prediction of normals if the surface is sufficiently smooth, while in [19,20], exemplars are used to provide shape priors.

Recently, the most popular strategy has been to jointly infer multiple modalities that contribute to the image formation process. In particular, several techniques produce both normal maps and surface reflectance [11,17,21,22], and the method [9] predicts a depth map in addition. This has been one of our motivations for developing a multi-stream CNN model that outputs multiple shape representations, as will be discussed in Section 3. A second motivation has been the observation that disentangling the representation of intrinsic surface properties, such as normals, albedo, and lighting can be beneficial in a Deep Learning context [23]. However, in our approach, we do not attempt to recover the lighting or reflectance explicitly, since such measurements are difficult to obtain ground-truth annotations for. Instead, we let the network learn how to handle these quantities from data. As evidenced by our experiments, even without explicitly modeling or predicting lighting and material BRDF, our network can successfully reconstruct fine surface details of complex shapes acquired under realistic conditions.

So far, there have been relatively few learning-based approaches to SfS. We only know of the works reported in [11,24]. In [11], not only normals but also reflectance properties are predicted, which makes the ground truth required to train the system hard to obtain. While [24] only requires ground-truth normals, it is dedicated to processing infrared images. By contrast, our approach can be trained from surface normals only and takes a standard RGB image as input.

In the context of monocular reconstruction of deformable surfaces, the most recent methods rely on CNNs to regress from the image to mesh vertices [12,25]. However, while [12] can recover complex deformations, it focuses on well-textured surfaces. By contrast, [25] handles poorly-textured surfaces, but visual inspection

of their results clearly shows that the method oversmoothes the surface shape. Our approach focuses on texture-less objects, and, as depicted by our results in Fig. 1, is able to reconstruct fine-grained deformations, such as the creases of a T-Shirt. Furthermore, we show that mesh representations are outperformed by normal- and depth-based ones for this task.

3 Our Approach

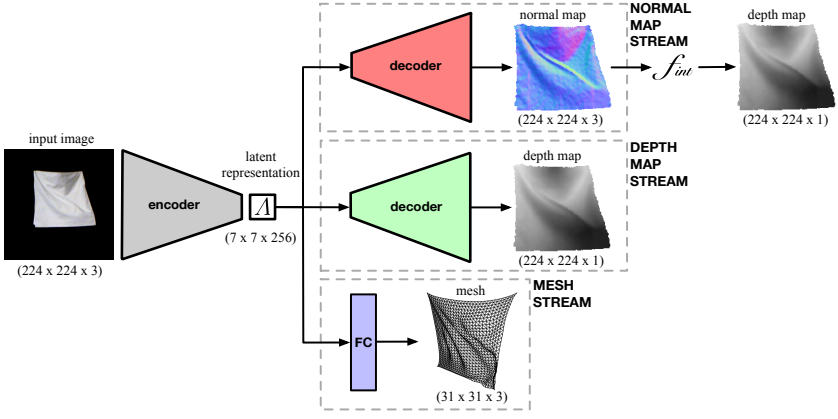

Fig. 2. Surface reconstruction architecture. Our model is based on the SegNet [26] one, but here we consider multiple output branches. The encoder, in gray, outputs the same latent representation Λ for the normal, depth, and vertex branch. A different decoder, shown in red, green, and blue, is then used for each. In other words, this creates three potential steams, which can be either trained individually or jointly. When recovering normals only, an additional integration step is required to compute depths.

3.1 Problem Formulation

Let $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ be an RGB image of size $W \times H$ and $\mathbf{B} \in \mathbb{N}_0^{H \times W \times 3}$ a binary mask that denotes the foreground region to be recovered. Our goal is to learn a mapping $f_{SR} : \mathbf{I} \odot \mathbf{B} \rightarrow \mathbf{S}$, where $\mathbf{S}$ represents the corresponding 3D surface. For deformable surfaces, a natural 3D representation would be a vector $\mathbf{S}_M$ containing the 3D vertex coordinates of a triangulated mesh, as in [12]. However, other representations such as a depth map $\mathbf{S}_D$ or a normal map $\mathbf{S}_N$, which are more prevalent in SfS papers, can be also be used. In fact, these representations are not mutually exclusive, and we can train a network to return one or more of them, as shown in Fig. 2. In this section, we discuss this general scenario. However, in practice, we have found that the more natural, mesh-based representation was

not as effective as the other two, and would even harm performance when used in conjunction with them.

Given a calibrated camera, a triangulated mesh with V vertices can be expressed as a vector $\mathbf{S}_M \in \mathbb{R}^{3V}$ of 3D points in the camera coordinate frame. By contrast, a depth map and a normal map can be encoded as images instead of vectors. Specifically, the depth map $\mathbf{S}_D$ is a $W \times H$ floating point image, and the normal map $\mathbf{S}_N$ is a three-channel floating point image of size $W \times H \times 3$ representing the x, y and z coordinates of the normal vector expressed in the camera coordinate frame. Training depth maps can be easily acquired using existing depth sensors, such as the Microsoft Kinect camera. This data can be converted into ground-truth normal maps by differentiating the depth maps. By contrast, obtaining training data for 3D meshes for real images is harder and requires much more processing, since depth sensors do not provide correspondences between points on the surface. However, unlike the other two representations, 3D meshes can represent self-occluded parts of the surface, albeit at the cost of constraining the topology much more. We will discuss these tradeoffs in more detail in the results section.

3.2 Shape Recovery Networks

In this work, we rely on the SegNet deep autoencoding architecture [26] depicted by Fig. 2 to regress from the image to each of our three representations.

Let $\mathbf{I}_m = \mathbf{I} \odot \mathbf{B}$ be the foreground image. The encoder performs feature extraction and outputs a latent representation tensor $\Lambda(\mathbf{I}_m) \in \mathbb{R}^{H_L \times W_L \times C_L}$, whose spatial size is $(H_L \times W_L)$ and third dimension C_L is the number of filters in the last convolutional layer of the encoder. Assuming that the features that must be learned are the same independently of the final shape representation, we use the same encoder in all three cases. Note, however, that the weights of this encoder will differ if we train it, for instance, for depth map prediction only or for mesh prediction only. Keeping the encoder design and shape of latent representation $\Lambda(\mathbf{I}_m)$ the same for all three scenarios allows us to not only train each model separately but also jointly, which helps the model learning more robust feature extractors, as shown in Section 5.

Let Ψ_M, Ψ_D, Ψ_N be the decoders for mesh vertices, depth, and normals, respectively. Since the depth and normal maps both have an image topology, as does the output of the original SegNet, we use the same SegNet decoder architecture for these two modalities, except for the number of convolutional filters in the last convolutional layer, that is, 1 for depth and 3 for normals, and for the fact that these outputs do not need to be passed through a softmax. By contrast, when using the mesh representation, the output size is significantly smaller and shaped as a vector. We therefore take Ψ_M to be a single convolutional layer followed by average pooling and a fully connected layer to regress to the vertices' coordinates.

3.3 Loss Functions

Let us assume to be given N training samples. We represent each one as a tuple $(\mathbf{I}^n, \mathbf{B}^n, \mathbf{v}^n, \mathbf{D}^n, \mathbf{N}^n)$, that is, an input image $\mathbf{I}^n$ with corresponding foreground mask $\mathbf{B}^n$, ground-truth mesh vertices $\mathbf{v}^n$, depth map $\mathbf{D}^n$ and normal map $\mathbf{N}^n$. We define 3 loss functions for the three potential outputs of our network.

To train Ψ_M, we define the loss as the Mean Square Error between the vertex locations and the ground truth. That is,

$$\mathcal{L}_M = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{V}\sum_{i=1}^{V}\|\mathbf{v}_i^n - \Psi_M(\Lambda(\mathbf{I}_m^n))_i\|^2 \ , \tag{1}$$

where a subscript i denotes the vertex number.

Since we use training data whose average distance to the camera is roughly constant and focus on recovering local high-frequency deformations, to train Ψ_D, we minimize the loss

$$\mathcal{L}_D = \frac{1}{N}\sum_{n=1}^{N}\frac{\sum_i |\mathbf{D}_i^n - \Psi_D(\Lambda(\mathbf{I}_m^n))_i|\mathbf{B}_i^n}{\sum_i \mathbf{B}_i^n} \ , \tag{2}$$

where i denotes the image location. Note that we only take into account pixels within the binary mask $\mathbf{B}$. In other words, we handle the depth ambiguity by recovering depth variations around the mean depth of our training data, instead of using a scale invariant measure as in [27]. As will be discussed in Section 5, to evaluate accuracy at test time, we rescale the prediction so as to align it with the ground truth.

Similarly, to train Ψ_N, we define a loss $\mathcal{L}_N$ that relies on a linearized version of the cosine similarity [28] and add to it a term that favors unit length vectors. We take it to be

$$\mathcal{L}_N = \frac{1}{N}\sum_{n=1}^{N}\frac{\left[\sum_i \mathbf{B}_i^n\left(\alpha \arccos\left(\frac{\mathbf{N}_i^n\hat{\mathbf{N}}_i^n}{\|\mathbf{N}_i^n\|\|\hat{\mathbf{N}}_i^n\|+\epsilon}\right)\frac{1}{\pi} + (||\hat{\mathbf{N}}_i^n|| - 1)^2\right)\right]}{\sum_i \mathbf{B}_i^n}, \tag{3}$$

where $\hat{\mathbf{N}}^n = \Psi_N(\Lambda(\mathbf{I}_m^n))$ denotes the predicted normal map, ϵ is a small positive constant that prevents divisions by zero and increases numerical stability, and α sets the relative influence of the two terms in the loss function. In our experiments, we chose $\alpha = 10$.

4 Real-World SfS Dataset

Successful training of most of the Deep Net models depends on access to large training databases. By contrast, the datasets used in the SfS literature remain relatively small. For example, the algorithm of [10] relies on 7 rigid objects under 20 different directional lighting. In [29], an augmented version of the MIT intrinsic image dataset [30] containing 20 rigid objects under various illumination

is used while the method of [31] works with a database of 6 human faces. Some authors use only synthetic data [11] while others augment the limited amount of real-data with synthetic data [10,9].

By contrast and to fully exploit the capacity of our model while avoiding overfitting, we captured a new large dataset of real deforming surfaces. We acquired sequences of RGB images and corresponding depth maps of a rectangular piece of cloth (`cloth`) and of a T-shirt (`tshirt`) undergoing complex deformations and seen under varying lighting conditions. We chose the cloth for two reasons. First, being a generic piece of cotton fabric makes it universal enough to capture a wide distribution of local deformations and appearance, even if it differs globally from other objects such as garments. Second, its flat rest state makes it easy to represent by a triangular mesh, as is often done in deformable surface reconstruction [32,2,1,12], and to test our mesh-based model. By contrast, the T-Shirt was chosen as a more complex real-world object to demonstrate the generality of our approach and the fact that training on `cloth` produces good results on `tshirt`.

The resulting dataset comprises a total of 22500 image-depth pairs, which is much larger than existing real-world SfS datasets. We will make our data publicly available upon acceptance of the paper. We now describe the acquisition process.

4.1 Acquisition Setup

Fig. 3 illustrates our setup. We placed the deformable surface, that is, the cloth or the T-shirt, in front of a dark background and captured synchronized RGB images and depth maps using a Microsoft Kinect camera, positioned so that its optical axis is roughly perpendicular to the background plane.

We used three fixed incandescent lamps, L_{right}, L_{left} and L_{center}, and a fourth one, L_{dyn}, that can move. All four were slightly slanted upwards and pointed towards the back of the room, that is, in the direction opposite to the camera's optical axis. As a result, the deforming surface was mostly illuminated by light reflecting off the walls and coming from the *radiance regions* shown in yellow in Fig. 3. This setup simulates directional but diffuse and soft lighting. The range of motion of L_{dyn} was chosen so that its radiance regions were slightly larger than those of the other three put together, which should result in a richer light distribution, as shown in Fig. 4. In the remainder of this section, we will consider 4 separate scenarios in which we use each lamp individually and will refer to them as L_{right}, L_{left}, L_{center}, and L_{dyn}.

The Microsoft Kinect camera was calibrated, thus yielding known camera intrinsics and depth maps in direct correspondence with the RGB images. We recorded sequences at 5 FPS to avoid capturing too many duplicates of nearly identical shapes. Each recorded frame consists of a 640 x 480 RGB image and a 640 x 480 depth map. Because of the deformation undergone by the surfaces, some of the frames feature substantial motion blur, which further increases both realism and the challenge.

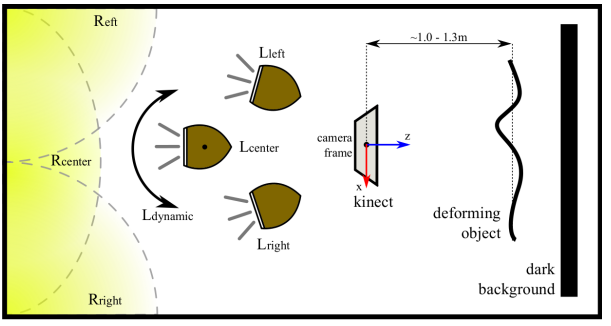

Fig. 3. Data acquisition setup. The deformable surface is placed between a Microsoft Kinect camera and a dark background. We use three static light sources, L_{right}, L_{left} and L_{center}, pointing towards the back wall and a fourth one, L_{dyn}, which can move. The deformable surface is therefore lit by complex indirect lighting.

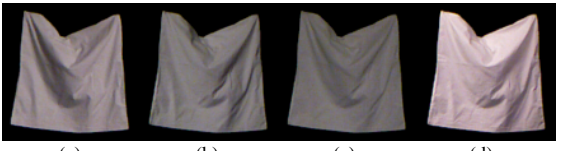

Fig. 4. Shading effects on a rectangular piece of cloth. We show the effects of (a) L_{right}, (b) L_{left}, (c) L_{center} and (d) a randomly chosen frame for L_{dyn}.

4.2 Deforming Surfaces

To acquire the images, we pinned the rectangular cloth to a fixed bar along a given edge or corners and manually deformed the rest. By contrast, the T-shirt was worn by a person making random body motions, and was captured separately from the front and from the back.

Altogether, this resulted in 19 sequences of 15809 samples for `cloth` and 12 sequences of 6739 samples for `tshirt`. A representative set of samples from our final dataset is depicted in Fig. 5

4.3 Data Preprocessing

As explained in Section 3, we assume the foreground binary mask **B** to be known for each image. To create it, we segmented the imaged objects in the RGB images by simple thresholding followed by hole filling of the biggest connected component. The masks were then used to segment the corresponding depth maps. Furthermore, the RGB images were white balanced using a standard color checker to account for the often unsatisfactory automatic white balance of the Kinect camera. Both the RGB images and depth maps were cropped to the spatial size of 224 x 224, which is the expected input/output size of SegNet. Since the raw depth maps contain noise and holes, we performed distance based clustering and hole filling by interpolation. The normal maps were computed by differentiating the depth maps in a finite difference sense.

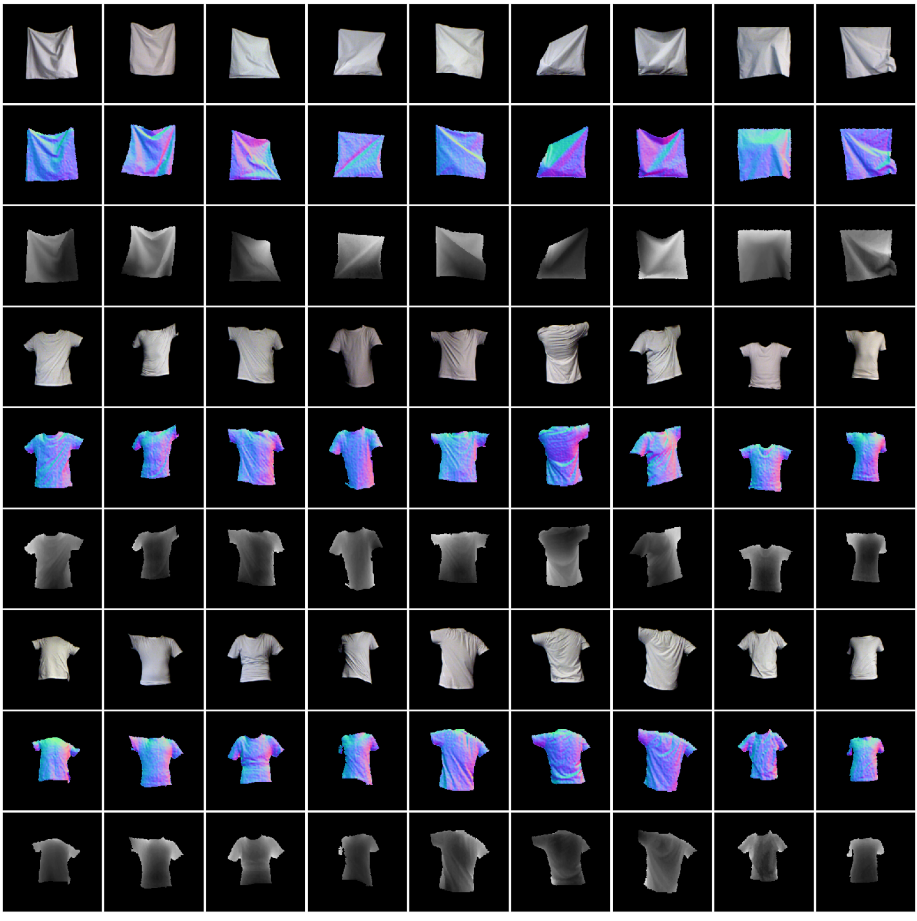

Fig. 5. Randomly chosen samples from out `cloth` and `tshirt` datasets together with corresponding GT normals and depth maps.

5 Experiments and Results

Recall from Section 3.2 that we can use the architecture depicted by Fig. 2 to recover normals, depths, or vertex coordinates either independently or jointly. In this section, we will focus on independent recovery of these three different representations, joint recovery of depth and normals and joint recovery of depth, normals and vertices. We will refer to the first three as **Normals**, **Depth**, and **Coords**, respectively, and to the remaining two as **N+D** and **N+D+C**. Note that in the last two cases, one can then evaluate the error of either one of the outputs. We indicate this as, e.g., **N+D/N** if we evaluate the normals.

We train and test our networks using the `cloth` and `tshirt` datasets under the four lighting scenarios introduced in Section 4. Each dataset includes separate

Experiment	Train obj.	Train light.	Test obj.	Test light.
`cloth-cloth`	cloth	L_r, L_l, L_d	cloth	L_c
`tshirt-tshirt`	tshirt	L_r, L_c, L_d	tshirt	L_l
`cloth-tshirt`	cloth	L_r, L_l, L_c, L_d	tshirt	L_r, L_l, L_c

Table 1. List of the experiments we conducted.

training and testing sequences, and we can either train and test on the same object or train on one and test on the other. We can also train with one particular set of lights and test with a different one. In both cases, this allows us to gauge the generalization abilities of our approach.

Table 1 summarizes the experiments we have conducted and whose results we report below. For each one, we randomly select 100 samples from the test sequences and report results on.

5.1 Implementation Details

For all the experiments described in this section, we used the Adam [35] optimizer to train the network. We started by training a network that only computes normals with a fixed learning rate of 0.001 and parameter α of the loss function of Eq. 3 set to 10. For **Normals**, we simply let the optimization proceed. For **N+D**, we applied an early stopping that stops the training once the validation loss stops decreasing for 30 consecutive epochs. We then started estimating both normals and depths by minimizing the loss function $\mathcal{L}_{\mathcal{ND}} = \beta\mathcal{L}_{\mathcal{N}} + \gamma\mathcal{L}_{\mathcal{D}}$, where we fixed the mixing coefficients $\beta = 1, \gamma = 3$ to promote the training of the yet untrained depth map stream. We halved the learning rate each time the validation loss stopped decreasing for 30 epochs in a row. For **N+D+C**, we similarly added the vertex-wise loss of Eq. 1 and continued training. For **Depth** and **Coords**, we proceeded exactly as for **Normals**. Our implementation relies on Keras with a Tensorflow backend.

5.2 Metrics

To evaluate the accuracy of the predicted mesh coordinates, we use the mean vertex-wise Euclidean distance, similar to the MSE of Eq. 1 we used to formulate the training loss, but without squaring the distances. We will refer to this metric as `mL2`.

Since the depth maps we produce are subject to an inherent global scale ambiguity [27], we first align the corresponding point cloud to the ground-truth using Procrustes transformation [33]. More precisely, let $\Theta(\mathbf{A}, \mathbf{K})$ be the 3D point cloud associated to depth map $\mathbf{A}$ with corresponding intrinsic matrix $\mathbf{K}$, and let $\Omega(\mathbf{P}, \mathbf{D}_b)$ denote the Procrustes transformation of cloud $\mathbf{P}$ with respect to depth map $\mathbf{D}_b$. Given a set of N ground-truth depth maps $\mathbf{D}^n$ and predicted

Experiment	Coords	N+D+C/C	N+D+C/D	N+D/D
cloth-cloth	49	41	15	**14.18**

Table 2. Evaluation of the effectiveness of meshes. We observe that, when using meshes, we obtain at best an `mL2` error of 41mm, with an **N+D+C** model evaluated on the mesh output. This is significantly larger than the `mL1` error of 15mm obtained by the same model evaluated on its predicted depth maps (**N+D+C/D**) and suggests that meshes are not appropriate in our context. This is further evidenced by the smaller error of **N+D/D**, which does not rely on meshes at all.

ones Δ^n, we compute accuracy in terms of the metric

$$mL1 = \frac{1}{N}\sum_{n=1}^{N}\frac{\sum_i |\mathbf{D}_i^n - \Omega(\Theta(\Delta^n, \mathbf{K}))_i^{(z)}|\mathbf{B}_i^n}{\sum_i \mathbf{B}_i^n}\ , \tag{4}$$

where the superscript (z) indicates that we only take the z value after Procrustes alignment.

Finally, for normal maps, we integrate the normals to recover the corresponding depth and again use the `mL1` metric. We also report the mean and median angular errors, which we denote as `mAE` and `dAE`, respectively, as well as the fractions of normals exhibiting smaller angular error than 10°, 20° and 30°.

5.3 Effectiveness of Meshes or the Lack Thereof

As mentioned in Section 1, our initial intuition was to follow the trend in deformable surface reconstruction and represent the surface as a triangular mesh. Here, we evaluate the results of such a representation, compared to depth and normal maps, and show that it is not as effective as them in our context. This is evidenced by the results in Table 2, where we compare **Coords**, **N+D+C/C**, **N+D+C/D** and **N+D/D**. When using meshes only (**Coords**), we obtain an `mL2` error of 49mm, which can be improved to 41mm by jointly using the normal and depth maps (**N+D+C/C**). However, evaluating the same 3-stream network using the predicted depth map (**N+D+C/D**) yields an `mL1` error of 15mm only. While the two metrics are not identical, such a large gap still evidences the better accuracy of the depth maps. To further evidence this, we can see that removing the meshes from the network and evaluating depth again (**N+D/D**) yields an even lower error of 14.18mm, thus showing that meshes in fact harm the performance. In the following experiments, we therefore discard the meshes from our approach.

5.4 Separate vs Joint Learning

In Table 3, we report the accuracy of **Normals**, **Depth**, and **N+D**. In the latter case, we can evaluate either the predicted normals or the predicted depth maps, which we denote as **N+D**/Nb and **N+D**/Db, respectively. We evaluate

Experiment	N+D/N Normals/N		N+D/D Depth/D	
cloth-cloth	**13.42**	13.67	**14.18**	16.4
tshirt-tshirt	11.53	**10.94**	**12.63**	13.73
cloth-tshirt	**21.33**	21.52	**24.86**	27.03

Table 3. Comparison of Depth, Normals and N+D in different scenarios. In general, the normal predictions yield lower `mL1` error than the predicted depth maps, and joint training outperforms the single-stream models.

different scenarios consisting of either training and testing on the same object, or training on `cloth` and testing on `tshirt`.

As can be seen in the table, using the normal predictions, followed by integration, tends to yield lower errors than the predicted depth maps. More importantly, training jointly on normals and depth performs best overall, which is in keeping with the idea that forcing the network to learn features that disentangle the different contributions helps [23,34]. Interestingly, training on `cloth` and testing on `tshirt` degrades the accuracy but still yields a competitive result as we will see in the following section. This is further evidenced by Fig. 6, which shows that our approach can recover substantial amounts of details from the T-shirt, whether trained on `tshirt` or on `cloth`.

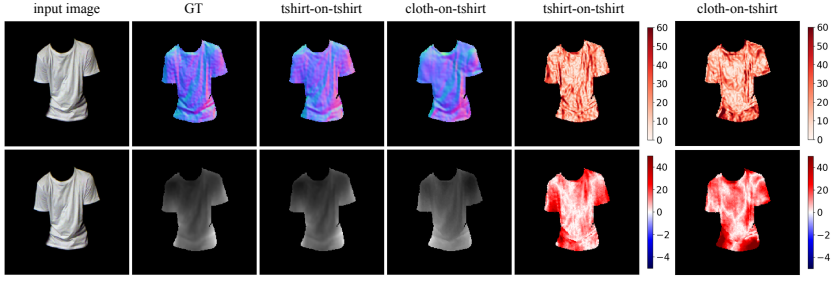

Fig. 6. Qualitative results of N+D. Top row: Analysis of the predicted normals. Bottom row: Analysis of the predicted depth. From left to right, we show: the input image, the ground-truth, our predictions in the two different scenarios and error maps in the two different scenarios. The error maps represent the angular error for the normals and the `mL1` for the depth. Note that we can recover many fine creases from the T-shirt.

5.5 Comparing against a State-of-the-Art SfS Approach

Here, we compare our results to those of the SIRFS method [9], which we briefly described inSection 2. This choice was motivated by the fact that SIRFS constitutes a state-of-the-art SfS method whose code is publicly available. Given the input image and segmentation mask, SIRFS performs intrinsic image decomposition into a normal map, depth map, lighting and reflectance. To compare with

Experiment	Method	**mAE** [°]	**dAE** [°]	$< \mathbf{10}^\circ$	$< \mathbf{20}^\circ$	$< \mathbf{30}^\circ$	**mL1** [mm]
cloth-cloth	SIRFS	37.98	33.52	7.25	24.93	43.96	22.28 ± 0.34
	OURS	**20.93**	**15.1**	**29.1**	**65.47**	**83.01**	$\mathbf{13.42 \pm 0.11}$
tshirt-tshirt	SIRFS	30.17	25.53	11.78	36.63	59.62	22.57 ± 0.45
	OURS	**20.82**	**15.68**	**27.26**	**63.78**	**82.54**	$\mathbf{11.53 \pm 0.11}$
cloth-tshirt	SIRFS	30.08	25.93	10.49	35.03	59.15	22.09 ± 0.38
	OURS	**28.1**	**23.51**	**12.96**	**40.43**	**65.21**	$\mathbf{21.33 \pm 0.35}$

Table 4. Comparison with the method of [9]. Our approach outperforms SIRFS in all metrics, even in the challenging `cloth-tshirt` scenario, with a particularly large gap for the first 5 metrics that evaluate the quality of the predicted normals. We report mean values averaged over different random sampling of 100 test examples with standard deviations for the `mL1` metric.

our method we take SIFRS's normal map and depth map predictions, integrate the normals and align to the ground-truth depth map, as explained in Section 5.2 and we do the same for our own results.

The results of this comparison are summarized in Table 4, where we evaluate several normal-based error metrics and one depth-based metric, the `mL1`. For the latter, we report the best results of SIRFS obtained either directly from the depth estimates, or by integrating the normal estimates. For our approach, we report the results based on the predicted normals, since we have found that they were slightly better than those obtained from our depth predictions. Note that we outperform SIRFS in all metrics. In the most challenging scenario (`cloth-tshirt`), our method still achieves lower errors, particularly in metrics evaluating normal quality. Figs. 1 and 7 show qualitative results for normal prediction. Our method clearly outperforms SIRFS when it comes to reconstructing the finer details of local creases.

In Table 5, we compare the run-times of our approach with those of SIRFS. Note that our method performs *orders of magnitude faster*. This is due to the fact that SIRFS relies on a costly optimization, whereas, in our case, all the heavy-lifting was done at training time and inference only requires a feed-forward pass through the network.

Model	SIRFS	OUR_N	OUR_D	OUR_N+D
t [s]	113.653	**0.01**	**0.01**	0.016

Table 5. Comparison of the run-times of our approach with those of SIRFS. We report the average time needed to process one input image of size 224×224 px.

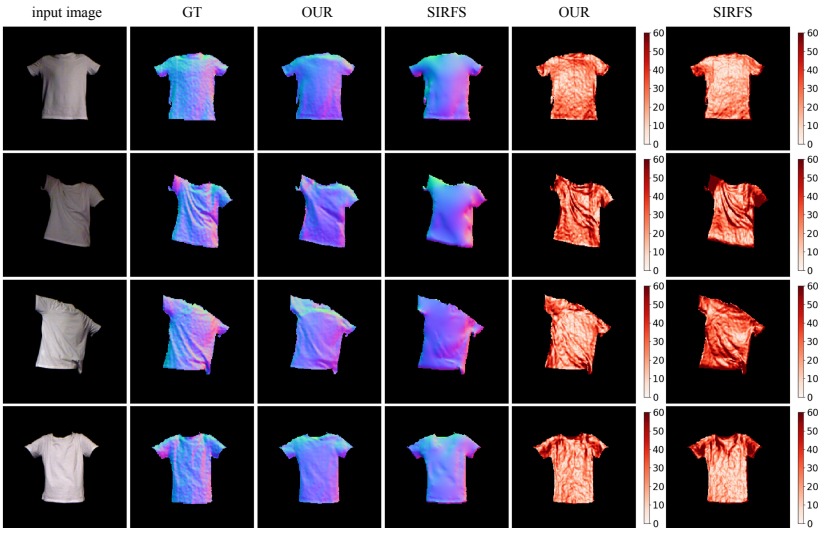

Fig. 7. Qualitative comparison to SIRFS for the `cloth-tshirt` scenario. Even in this challenging scenario, our method is able to recover finer details than SIRFS.

6 Conclusion

We have introduced a framework to reconstructing the 3D shape of a texture-less, deformable surface from a single image. To this end, we have followed a data-driven approach, thus essentially learning to perform Shape-from-Shading. Our experiments have demonstrated that, while meshes have proven effective to deal with well-textured deformable surfaces, they are much less well-suited than depth- and normal-based representations for texture-less ones. Furthermore, our comparison with a state-of-the-art SfS method has shown that our reconstructions were more accurate, particularly in terms of normal quality. This is the case even when training our model on one object and testing it on a different one. We expect that such a generalizability would further increase were we to use larger amounts of training data. Therefore, in the future, we will dedicate time to creating a larger-scale dataset of texture-less, deformable objects.

References

1. Ngo, D., Ostlund, J., Fua, P.: Template-Based Monocular 3D Shape Recovery Using Laplacian Meshes. IEEE Transactions on Pattern Analysis and Machine Intelligence **38**(1) (2016) 172–187
2. Bartoli, A., Gérard, Y., Chadebecq, F., Collins, T.: On Template-Based Reconstruction from a Single View: Analytical Solutions and Proofs of Well-Posedness for Developable, Isometric and Conformal Surfaces. In: Conference on Computer Vision and Pattern Recognition. (2012)
3. Varol, A., Shaji, A., Salzmann, M., Fua, P.: Monocular 3d reconstruction of locally textured surfaces. PAMI **34**(6) (June 2012) 1118–1130
4. Wang, X., Salzmann, M., Wang, F., Zhao, J.: Template-free 3d reconstruction of poorly-textured nonrigid surfaces. ECCV (2016)
5. Horn, B., Brooks, M.: Shape from Shading. MIT Press (1989)
6. Zhang, R., Tsai, P.S., Cryer, J.E., Shah, M.: Shape from Shading: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **21**(8) (1999) 690–706
7. Durou, J.D., Falcone, M., Sagona, M.: Numerical Methods for Shape from Shading : A New Survey with Benchmarks. Computer Vision and Image Understanding **109**(1) (2008) 22–43
8. Prados, E., Faugeras, O.: Shape from Shading: A Well-Posed Problem? In: Conference on Computer Vision and Pattern Recognition. (June 2005)
9. Barron, J.T., Malik, J.: Shape, Illumination, and Reflectance from Shading. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(8) (2015) 1670–1687
10. Xiong, Y., Chakrabarti, A., Basri, R., Gortler, S., Jacobs, D., , Zickler, T.: From Shading to Local Shape. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(1) (October 2015) 67–79
11. Richter, S.R., Roth, S.: Discriminative shape from shading in uncalibrated illumination. In: Conference on Computer Vision and Pattern Recognition. (2015) 1128–1136
12. Pumarola, A.: Geometry-Aware Network for Non-Rigid Shape Prediction from a Single View. Conference on Computer Vision and Pattern Recognition (2018)
13. Belhumeur, P., Kriegman, D., Yuille, A.: The Bas-Relief Ambiguity. International Journal of Computer Vision **35**(1) (1999) 33–44
14. Ecker, A., Jepson, A.D.: Polynomial shape from shading. In: Conference on Computer Vision and Pattern Recognition. (June 2010) 145–152
15. Samaras, D., Metaxas, D., Fua, P., Leclerc, Y.: Variable Albedo Surface Reconstruction from Stereo and Shape from Shading. In: Conference on Computer Vision and Pattern Recognition. (June 2000)
16. Barron, J., Malik, J.: High-frequency shape and albedo from shading using natural image statistics. Conference on Computer Vision and Pattern Recognition (2011) 2521–2528
17. Oxholm, G., Nishino, K.: Shape and reflectance from natural illumination. In: European Conference on Computer Vision. (2012) 528–541
18. Johnson, M.K., Adelson, E.H.: Shape estimation in natural illumination. In: Conference on Computer Vision and Pattern Recognition. (2011) 2553–2560
19. Hassner, T., Basri, R.: Example based 3d reconstruction from single 2d images. In: Conference on Computer Vision and Pattern Recognition Workshops. (June 2006) 15–15
20. Huang, X., Gao, J., Wang, L., Yang, R.: Examplar-based shape from shading. In: 3DIM. (2007) 349–356

21. Zoran, D., Krishnan, D., Bento, J., Freeman, B.: Shape and illumination from shading using the generic viewpoint assumption. In: Advances in Neural Information Processing Systems. (2014) 226–234
22. Choe, G., Narasimhan, S.G., So Kweon, I.: Simultaneous estimation of near ir brdf and fine-scale surface geometry. In: Conference on Computer Vision and Pattern Recognition. (June 2016)
23. Shu, Z., Yumer, E., Hadap, S., Sunkavalli, K., Shechtman, E., Samaras, D.: Neural Face Editing with Intrinsic Image Disentangling. In: Conference on Computer Vision and Pattern Recognition. (2017)
24. Yoon, Y., Choe, G., Kim, N., Lee, J.Y., Kweon, I.S.: Fine-scale surface normal estimation using a single nir image. European Conference on Computer Vision (2016)
25. Danecek, R., Dibra, E., Öztireli, A.C., Ziegler, R., Gross, M.: Deepgarment : 3d garment shape estimation from a single image. Eurographics (2017)
26. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
27. Eigen, D., Puhrsch, C., Fergus, R.: Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network. In: Advances in Neural Information Processing Systems. (2014) 2366–2374
28. Trigeorgis, G., Snape, P., Kokkinos, I., Zafeiriou, S.: Face Normals "In-the-Wild" Using Fully Convolutional Networks. In: Conference on Computer Vision and Pattern Recognition. (2017) 340–349
29. Barron, J., Malik, J.: Shape, Albedo, and Illumination from a Single Image of an Unknown Object. In: Conference on Computer Vision and Pattern Recognition. (June 2012)
30. Grosse, R., Johnson, M.K., Adelson, E.H., Freeman, W.T.: Ground truth dataset and baseline evaluations for intrinsic image algorithms. In: Conference on Computer Vision and Pattern Recognition. (Sept 2009) 2335–2342
31. Khan, S., Shah, M.: Tracking Multiple Occluding People by Localizing on Multiple Scene Planes. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(3) (March 2009) 505–519
32. Salzmann, M., Fua, P.: Linear Local Models for Monocular Reconstruction of Deformable Surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(5) (2011) 931–944
33. Stegmann, M.B., Gomez, D.D.: A brief introduction to statistical shape analysis (2002)
34. Kulkarni, T.D., Whitney, W., Kohli, P., Tenenbaum, J.B.: Deep Convolutional Inverse Graphics Network. In: arXiv. (2015)
35. Kingma, D., Ba, J.: Adam: A Method for Stochastic Optimisation. In: International Conference for Learning Representations. (2015)