Geometry-Aware Network for Non-Rigid Shape Prediction from a Single View

A. Pumarola[1] A. Agudo[1] L. Porzi[2] A. Sanfeliu[1] V. Lepetit[3] F. Moreno-Noguer[1]
[1]Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain
[2]Mapillary Research, Graz, Austria
[3]Laboratoire Bordelais de Recherche en Informatique, Université de Bordeaux, France

Abstract

We propose a method for predicting the 3D shape of a deformable surface from a single view. By contrast with previous approaches, we do not need a pre-registered template of the surface, and our method is robust to the lack of texture and partial occlusions. At the core of our approach is a geometry-aware deep architecture that tackles the problem as usually done in analytic solutions: first perform 2D detection of the mesh and then estimate a 3D shape that is geometrically consistent with the image. We train this architecture in an end-to-end manner using a large dataset of synthetic renderings of shapes under different levels of deformation, material properties, textures and lighting conditions. We evaluate our approach on a test split of this dataset and available real benchmarks, consistently improving state-of-the-art solutions with a significantly lower computational time.

1. Introduction

Motivated by the current success of deep learning methods for estimating a depth map from a single image of a scene [18, 19, 20], in this paper we tackle the related problem of estimating the underlying parametric model defining the shape of a non-rigid surface from a single image. This problem has been traditionally addressed in the context of the Shape-from-Template (SfT) paradigm [9], requiring a reference template image of the surface for which the 3D geometry is known, and a set of 3D-to-2D point correspondences or a mapping between this template and the input image. This approach, however, may be difficult to hold in practice, specially when considering low-textured surfaces.

In this work we relax previous assumptions and present a learning-based approach that allows for globally non-rigid surface reconstruction from a single image without relying on point correspondences, and which in particular, shows robustness to situations rarely addressed previously: lack of surface texture and large occlusions. Our model is based on a fully differentiable Deep Neural Network that estimates a 3D shape from a single image in an end-to-end manner, and builds upon three branches that enforce geometry consistency of the solution.

More exactly, as illustrated in Fig. 1, a first branch of the proposed architecture (the '2D Detection Branch') is responsible for localizing the mesh onto the image, and for fitting a 2D grid to it. The 2D vertices of this grid are then lifted to 3D by the 'Depth Branch', a regressor that combines the 2D detector confidence maps and the input image features. Finally, a 'Shape Branch' is responsible for recovering the full shape while ensuring that the estimated 3D coordinates correctly re-project onto the image. During training, this branch also incorporates a novel fully-differentiable layer that performs a Procrustes transformation and aligns the estimated 3D mesh with the ground truth one. This branch is important as it was proven important to perform Procrustes alignment in previous approaches for adapting to datasets with different reference frames and metrics. It also favors convergence of the learning process.

Since there is no dataset large enough to train data-hungry deep learning algorithms such as ours, we have created our own using a rendering tool. We have synthesized 128,000 photo-realistic pairs input 2D-image/3D-shape accounting for different levels of deformations, amount and type of texture, material properties, viewpoints, lighting conditions and occlusion. Figure 3-Top shows some examples. Evaluation on a test split of this dataset demonstrates remarkable improvement of our network compared to state-of-the-art SfT techniques, which typically rely on known 3D-to-2D correspondences, especially under strong occlusions and poorly-textured surfaces. Furthermore, our model learned with synthetic data can be easily fine-tuned to real sequences, using just a few additional real training samples. Results on the CVLab sequences [48] with a bending paper and a deforming t-shirt again clearly show that our method outperforms existing approaches.

In summary, our main contributions are: 1) the first—to the best of our knowledge—fully-differentiable model for non-rigid surface reconstruction from a single image that does not require initialization, accurate knowledge of

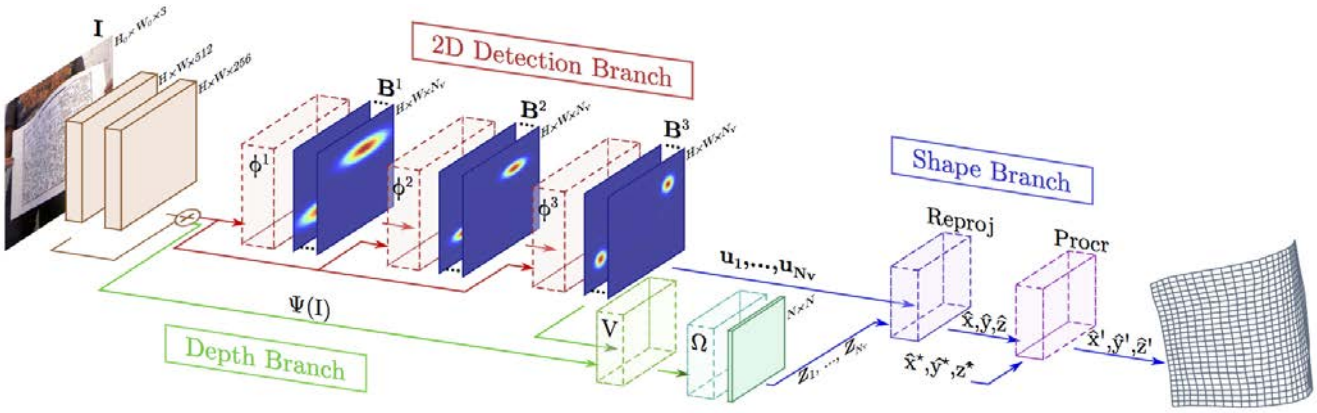


Figure 1. **Overview of our approach.** The proposed architecture consists of three main branches. The '2D Detection Branch' is responsible for the 2D location of the mesh and the associated belief maps. The 'Depth Branch' lifts the 2D detected mesh by leveraging on image cues and the detection uncertainties. Finally, the 'Shape Branch' fuses the 2D detections and their estimated depths to obtain 3D shape in such a way that perspective projection is enforced. An additional 'Procrustes Layer' is used during training to align the estimated mesh with the ground truth one.

the template, 3D-to-2D correspondences, nor hand-crafted constraints; 2) a geometry-aware architecture that embeds a pinhole camera model and encodes rigid alignment during training; and 3) a large photo-realistic dataset of images of non-rigid surfaces annotated with the corresponding 3D shapes, which will be made publicly available, and we hope it will inspire future research in the field.

2. Related Work

Reconstructing non-rigid surfaces from monocular images is known to be a severely ill-posed problem which requires introducing different sources of prior knowledge in order to be solved. In this section, we will split related work into methods that define these priors based on pre-defined models (either physically-based or handcrafted) and techniques that learn them from training data.

Early approaches described non-rigid surfaces using models inspired by physics, such as superquadrics [33], thin-plates [31], elastic models [24] and finite-elements [32]. These representations, however, could not accurately approximate the non-linear behavior of large deformations.

More complex deformations can be captured by SfT approaches [9, 12, 35, 37, 38, 40, 42, 43, 49], which aim at recovering the surface geometry given a reference configuration in which the template shape is known, and a set of 3D-to-2D correspondences between this shape and the input image. On top of this, additional constraints enforcing isometry [43], conformal warps [9] and photometric consistency [35, 37] are considered. While effective, SfT methods are very sensitive to the initial set of matches, which may be difficult to establish in practice, especially under occlusions, low textured surfaces and varying illumination.

Temporal information is another typically exploited prior. Non-rigid-shape-from-motion techniques generally extend Tomasi and Kanade's rigid factorization algorithm [46] to recover deformable shape and camera motion from a sequence of 2D tracks, exploiting physical [3] and low-rank constraints on the shape [1, 4, 25, 47], trajectory [5] or the forces inducing the deformation [2]. Again, these methods rely on the fact that 2D point tracks can be readily computed, limiting thus their general applicability to relatively well-textured surfaces.

The need of point correspondences is circumvented by template-free approaches that perform a per-point 3D reconstruction by minimizing an objective function on geometric and photometric cues [6, 16, 51, 53]. The shading models considered by these approaches, however, use to be oversimplifications of the reality, either considering brightness constancy [51] or Lambertian surfaces lit by point light sources [53].

More realistic deformation and appearance models can be learned from training data. The first attempt along this line corresponds to the active appearance models [13], which learned low-dimensional 2D models for face tracking. This was later extended to 3D by the active shape and morphable models [10, 30], and by methods integrating these models into the SfT formulation [36]. Yet, all these approaches still rely on feature points detected over the whole surface or at its boundary [44], which are difficult to obtain in practice.

Following the success of recent deep convolutional networks in related topics such as 3D human pose recovery [29, 34, 39], depth [17, 18, 19, 27, 41, 54] and surface normal reconstruction on rigid objects [7, 8, 17, 50], we introduce a unified formulation for the problem of estimating non-rigid shape from single images, that simul-

taneously performs 2D detection and 3D lifting while enforcing geometry consistency. The framework we propose allows tackling a series of situations which, to the best of our knowledge, are not jointly addressed by existing approaches for reconstructing deformable surfaces: it does not require pre-computing point correspondences, it is effective on poorly textured surfaces, it is robust to partial occlusions and corrupted object boundaries, and works well under varying lighting conditions. Moreover, 3D shape inference is fast as often with deep networks.

Probably the most closely related work to ours is that of Tewari *et al*. [45], which trains a deep auto-encoder model for monocular face reconstruction. However, this work relies on a low-rank shape model that limits their feasible solutions to shapes with relatively small deformations. Furthermore, the range of textures for face reconstruction is limited while we consider general textures.

3. Our Approach

Our framework for estimating a non-rigid shape from a single image is shown in Fig. 1. We have devised an architecture with three branches, each responsible of reasoning about a different geometric aspect of the problem. The first two branches are arranged in parallel and perform probabilistic 2D detection of the mesh in the image plane and depth estimation (red and green regions in the figure, respectively). These two branches are then merged (blue region in the figure) in order to lift the 2D detections to 3D space, such that the estimated surface correctly re-projects onto the input image and it is properly aligned with the ground truth shape. In the results section we will show that reasoning in such a structured way provides much better results than trying to directly regress the shape from the input image, despite using considerably deeper networks.

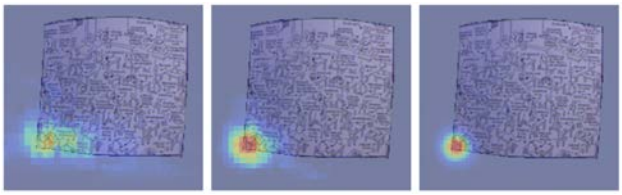
4. Geometry-Aware Network

In this section we formulate the problem and describe the network architecture we propose, which is made of three main branches named 2D Detection Branch, the Depth Branch, and the Shape Branch. We also define the loss layer for learning the whole model.

4.1. Problem Formulation

We aim at designing a deep learning framework that directly estimates a non-rigid 3D shape from an input RGB image $\mathbf{I} \in \mathbb{R}^{H_o \times W_o \times 3}$. The shape is represented as a triangulated 3D mesh with N_v vertices $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_{N_v})$, where $\mathbf{x}_i = (x_i, y_i, z_i)$ are the coordinates of the i-th vertex, expressed in the camera coordinate system. In the following, we assume the structure of the mesh to be known, being a $N \times N$ rectangular grid, i.e., $N_v = N^2$.

We also assume the calibration parameters of the camera to be known, namely the focal lengths, f_u and f_v, and the principal point (u_c, v_c).



t=1 t=2 t=3

Figure 2. **Refinement of the 2D vertices position.** Output (for one specific vertex) of the regressor Φ^t for three consecutive time steps. Note how the uncertainly in the vertex location is progressively reduced.

4.2. 2D Detection Branch

Given an input image $\mathbf{I}$, the first step consists in extracting image features from a pre-trained network, in our case we concatenate two Resnet V2 blocks [22]. For each block, the stride of the last unit is set to one, in order to keep the same spatial resolution for the two units. Let us denote these features as $\Psi(\mathbf{I}) \in \mathbb{R}^{H \times W \times C}$.

The image features are then fed into the 2D detection network, which is responsible for estimating the 2D locations of the mesh vertices $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_{N_v}) \in \mathcal{U}$, where $\mathbf{u}_i = (u_i, v_i)$ and $\mathcal{U}$ is the set of all (u, v) pixel locations in the input image $\mathbf{I}$. Drawing inspiration on the convolutional pose machines [52] for human pose estimation, the 2D location of each vertex $\mathbf{u}_i$ is represented as a probability density map $\mathbf{B}_i \in \mathbb{R}^{H \times W}$ computed over the entire image domain as:

$$\mathbf{B}_i[u, v] = P(\mathbf{u}_i = (u, v)) \ \ , \forall \ (u, v) \in \mathcal{U}. \tag{1}$$

As in [52] these belief maps are estimated in an iterative manner. In particular, let $\mathbf{B}^t = (\mathbf{B}^t_1, \dots, \mathbf{B}^t_{N_v}) \in \mathbb{R}^{H \times W \times N_v}$ be the concatenation of all belief maps at iteration t. This tensor is estimated by a regressor function Φ^t, which takes as input the image features and the concatenated belief maps at the previous stage $t-1$:

$$\Phi^t(\Psi(\mathbf{I}), \mathbf{B}^{t-1}) \rightarrow \mathbf{B}^t \ . \tag{2}$$

In the first step, the regressor is only fed with the image features, that is $\Phi^1 \equiv \Phi^1(\Psi(\mathbf{I}))$. We denote by T_{max} the maximum number of iterations. As it is shown in Fig. 2, after each iteration, the location of the vertices is progressively refined.

In order to implement the regressor $\Phi^t(\cdot)$ we use again ResNet V2 blocks followed by two convolutional layers. The output of each Φ^t is normalized with respect to H and W to guarantee that $\sum_{u=1}^{H} \sum_{v=1}^{W} \mathbf{B}^t_i[u, v] = 1$, $\forall i \in \{1, \dots, N_v\}$, and $\forall t \in \{1, \dots, T_{\text{max}}\}$.

Finally, it is worth noting that the 2D Detection Branch we have just described is fully differentiable. The output $\mathbf{u}_i = (u_i, v_i)$ for the $i-$th vertex can be estimated as the following weighted sum over the last belief map $\mathbf{B}^{T_{\max}}$:

$$u_i = \frac{\sum_{(u,v)\in\mathcal{U}} u \cdot \mathbf{B}_i^{T_{\max}}[u,v]}{\sum \mathbf{B}_i^{T_{\max}}}, \qquad v_i = \frac{\sum_{(u,v)\in\mathcal{U}} v \cdot \mathbf{B}_i^{T_{\max}}[u,v]}{\sum \mathbf{B}_i^{T_{\max}}}$$

where $\sum \mathbf{B}_i^{T_{\max}}$ sums over all elements of $\mathbf{B}_i^{T_{\max}}$. These 2D estimates will be forwarded to the 'Shape Branch' described in Section 4.4, while the belief maps in $\mathbf{B}^{T_{\max}}$ will be used to infer the depth value for each of the vertices in the 'Depth Branch' described in Section 4.3.

4.3. Depth Branch

The belief maps $\mathbf{B}_i^{T_{\max}}$ of the 2D vertex locations in the above section are forwarded to the 'Depth Branch', to estimate the depth coordinate z_i for every vertex. Note that previous works in related problems like 3D human pose estimation [29, 34] have not taken advantage of the uncertainty typically associated to the feature detectors.

To do so, the proposed layer produces new feature maps $\mathbf{V}(\mathbf{B}^{T_{\max}}, \Psi(\mathbf{I})) \in \mathbb{R}^{N\times N\times C}$, that condition the input feature maps $\Psi(\mathbf{I}) \in \mathbb{R}^{H\times W\times C}$ with the probability maps $\mathbf{B}^{T_{\max}} \in \mathbb{R}^{H\times W\times N_v}$, that is:

$$\mathbf{V}[j(i), k(i), c] = \sum_{(u,v)\in\mathcal{U}} \mathbf{B}_i^{T_{\max}}[u,v] \cdot \Psi(\mathbf{I})[u,v,c] \quad (3)$$

$\forall i \in \{1,\dots,N_v\}, c \in \{1,\dots,C\}$, where $(j(i), k(i))$ converts the i-th input of an N_v-dimensional vector into a two dimensional input of an $N \times N$ matrix (recall that $N_v = N^2$).

These image features conditioned on the vertices 2D locations are then used as input of a regressor $\Omega(\cdot)$ to estimate the vertices' depth:

$$\Omega(\mathbf{V}(\mathbf{B}^{T_{\max}}, \Psi(\mathbf{I}))) \rightarrow (z_1, \dots, z_{N_v}). \quad (4)$$

Again, the regressor $\Omega(\cdot)$ consists in two ResNet V2 blocks followed by two convolutional layers and the full branch (conditioned features + regressor) is fully differentiable.

4.4. Shape Branch

The 2D locations and depth estimates are merged in order to estimate the shape while enforcing the projection constraints and rigid alignment consistency.

Given the estimates (u_i, v_i, z_i) in Eqs. (3) and (4) of the two first branches, the 3D position $\mathbf{x}_i = (x_i, y_i, z_i)$ of each vertex is recovered with a differentiable layer that models the pinhole reprojection model:

$$x_i = z_i \cdot \frac{u_i - u_c}{f_u}, \qquad y_i = z_i \cdot \frac{v_i - v_c}{f_v}, \qquad z_i = z_i. \quad (5)$$

This gives us an estimate of the deformable shape $\mathbf{X}$, and we could train the network by considering the L2 loss $||\mathbf{X} - \mathbf{X}^*||_2^2$ where $\mathbf{X}^*$ is the ground truth 3D shape. However, we propose introducing an additional layer, which computes the Procrustes alignment error between $\mathbf{X}$ and $\mathbf{X}^*$ in a fully differentiable manner, and build our loss function based on this error. Although this layer is removed at test time, we observed that it favors the convergence during training, helps adapting to different datasets, and most importantly, it improves the capacity of the rest of the network to capture the non-rigid component of the shape.

The Procrustes layer ('Procr' box in Fig. 1) is implemented by first normalizing $\mathbf{X}$ and $\mathbf{X}^*$ with respect to translation and scale. Let us denote by $\hat{\mathbf{X}} = (\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_{N_v})$ and $\hat{\mathbf{X}}^* = (\hat{\mathbf{x}}_1^*, \dots, \hat{\mathbf{x}}_{N_v}^*)$ these normalized versions.

Following [14], we can then compute the alignment error between $\hat{\mathbf{X}}$ and $\hat{\mathbf{X}}^*$, without having to explicitly estimate their relative rotation and translation as follows:

$$\text{Err\_Align}(\hat{\mathbf{X}}, \hat{\mathbf{X}}^*) = \sqrt{\frac{\sum_{i=1}^{N_v} |\hat{\mathbf{x}}_i|^2 + |\hat{\mathbf{x}}_i^*|^2 - 2\lambda_{\max}}{N_v}} \quad (6)$$

where $\lambda_{\max}$ is the maximum eigenvalue of a 4×4 matrix built in terms of the elements of $\hat{\mathbf{X}}$ and $\hat{\mathbf{X}}^*$. Since there exist differentiable approximations of the eigendecomposition (for example, the function `tf.self_adjoint_eigvals` in Tensorflow), the full 'Shape branch' is again differentiable.

4.5. Learning the Model

The cost function that we aim to minimize is a combination of the 3D alignment error in Eq. (6) and the 2D detection error produced at the output of each regressor Φ^t, for $t = \{1, \dots, T_{\max}\}$:

$$\mathcal{L} = \text{Err\_Align}(\hat{\mathbf{X}}, \hat{\mathbf{X}}^*) + \gamma \sum_{t=1}^{T_{\max}} \|\mathbf{B}^t - \mathbf{B}^*\|_2^2, \quad (7)$$

where $\mathbf{B}^*$ is a heat-map generated by placing Gaussian peaks at the ground truth 2D locations (u_i^*, v_i^*) of the mesh vertices. γ denotes a weight used to give similar orders of magnitude to each of the terms of the loss function.

Training Details. The model is trained with the synthetically generated dataset described in the next section, made of $H_o \times W_o = 224 \times 224$ images. The image features $\Psi(\mathbf{I})$ are obtained from a Resnet V2 network pre-trained on ImageNet, resulting in feature maps of size $H \times W \times C = 56 \times 56 \times 768$. In all our experiments we consider meshes of spatial resolution $N \times N = 9 \times 9$, thus, $N_v = 81$. The resulting belief maps B^t will be therefore of size $56 \times 56 \times 81$. In the '2D Detection Branch', we fixed the maximum number of iterations to $T_{\max} = 3$, as further stages did barely change the resulting belief maps distributions.

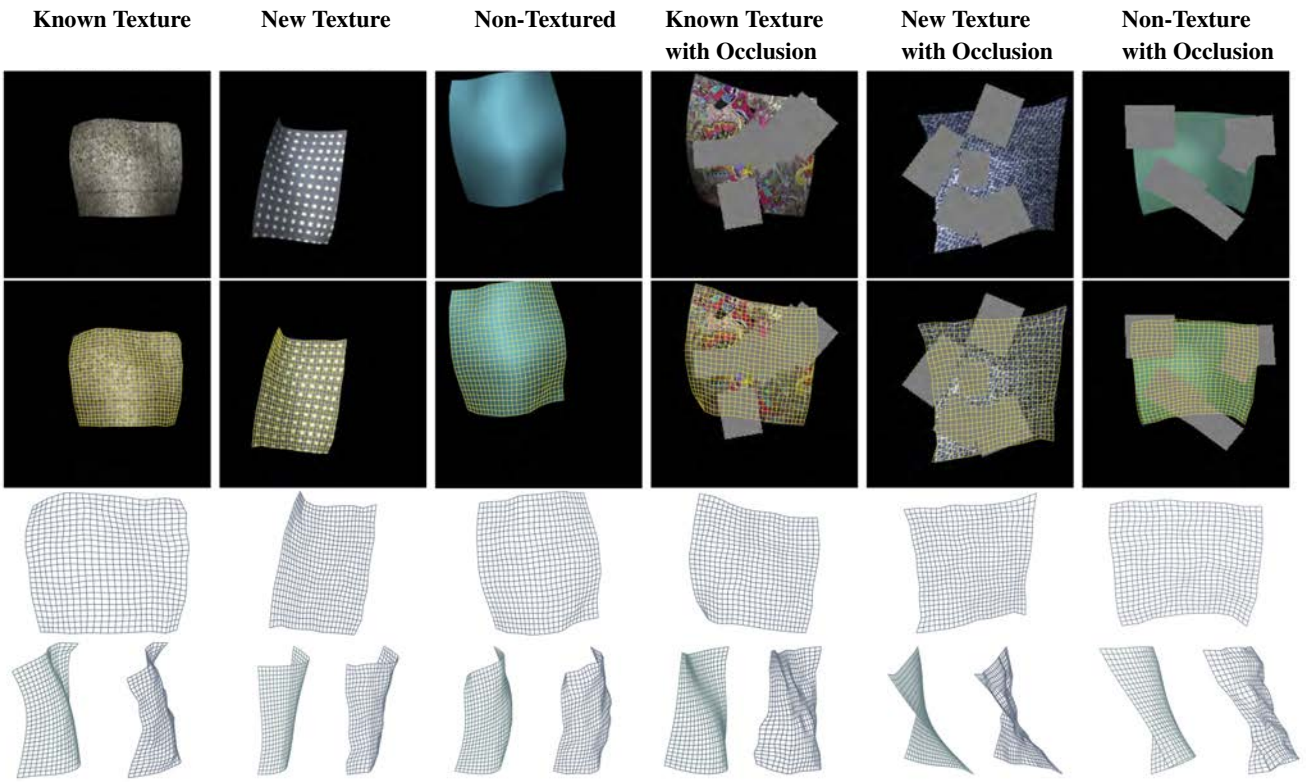


Figure 3. **Results on synthetic data.** Reconstructions samples in each of the six cases we consider (surfaces with known, new or no-texture, and with and without occlusions). **First Row:** Input image. **Second Row:** 3D estimated mesh projected onto the input image. **Third Row:** 3D estimated mesh seen from the camera view. **Last Row:** Side view of the ground truth mesh and our estimation (green and blue meshes, respectively). The reconstruction error is indicated at the bottom, to give significance to the errors in Table 1.

The training procedures is split in two stages: initially, only the regressors Φ^t are trained. Then, regressors Φ^t and Ω are jointly trained. In both cases, the parameters of the feature extractor $\Psi(\mathbf{I})$ are kept fixed. In Eq. (7) we set $\gamma = 5 \cdot 10^{-3}$. We use Adam solver [23] with a batch size of 3 images and weight decay of $4 \cdot 10^{-5}$. Every 2 epochs we exponentially decay the learning rate, which is initially set to $2 \cdot 10^{-4}$.

5. Dataset

It is well known that deep networks require large amounts of training data. However, the only existing dataset we are aware of that contains non-rigid surfaces annotated with ground-truth 3D shape is [48], which includes 505 images of a bending paper and a deforming t-shirt. This is far below what is needed, specially if we expect our network to generalize to non-observed textures. For this purpose, we have created a large synthetic dataset with 128,000 samples rendered with Autodesk™- Maya. Each sample consists of a 224×224 image and a 9×9 deformed shape. A few examples of the dataset are shown in Fig. 3-Top.

We generated our dataset by varying textures, deformations and lighting conditions. Concretely, we have chosen 200 different textures from [15] which is formed by repetitive patterns, rich, poor and plain textures. The deformations were generated for 40 different meshes (same topology but varying aspect ratios and sizes). The mesh dynamics were rendered by simulating a hanging piece of material held with up to 4 pins and moving with the wind. Four different materials, defined with four different stiffness matrices, were considered. The scene was lit by one point light source of high intensity with a random position, plus a component of ambient illumination. In all cases, we assumed a Lambertian reflectance.

The rendered dataset was augmented with all three possible flips of each image. Additionally, for each image, three new ones were generated by applying a random rigid transformation on the corresponding deformable surface. At training time, the dataset was further augmented with random color changes at pixel level (hue, saturation, contrast and brightness). The dataset will be made publicly available.

6. Experimental Validation

We now present results on synthetic and real data. We compare our approach, which we dub *DeformNet*,

Method	Known Text	New Text	No-Text	Time (ms)
Ba15Iso	8.54 / -	8.72 / -	- / -	495
Ba15Iso-It	5.65 / -	6.78 / -	- / -	15,507
Ba15Conf	30.50 / -	31.91 / -	- / -	11,232
Ch14IsoLsq	6.74 / -	6.95 / -	- / -	2618
Ch14IsoLsq-It	4.85 / -	5.3 / -	- / -	14,813
Resnet-50 V2	**0.92 / 3.83**	11.23 / 18.50	8.39 / 9.43	**152**
DeformNet	2.64 / 4.57	**3.28 / 4.09**	**2.86 / 4.62**	219

Table 1. **Evaluation on synthetic data.** Euclidean average distance between 3D ground-truth and estimated 3D reconstruction. Each pair 'err1 / err2' indicates the error without and with occlusions, respectively. Execution time in the last column is computed as the average time (in ms) to reconstruct a sample. Symbol '-' indicates that the method was not evaluated on this scenario, as they correspond to situations (no texture or large occlusions) that can not be addressed by template-based analytical solutions.

with the following state-of-the-art template-based solutions: Ba15Iso, the isometry-based solution proposed in [9]; Ba15Conf, a conformal-based approach, also from [9]; Ch14IsoLsq, the least-squares isometric reconstruction of [12]. We denote by Ba15so-It and Ch14IsoLsq-It the same previous methods after executing 25 iterations of the non-linear refinement proposed in [11]. This refinement step could not be applied to Ba15Conf due to computational time constraints. [12] showed that Ch14IsoLsq-It systematically outperformed the same baselines we consider here and also the methods introduced in [11, 40, 42]. We therefore consider Ch14IsoLsq-It to be the best current analytic approach to assess the potential of our solution. Additionally, we also compare against a deep network baseline, consisting of a ResNet-50 V2 architecture [22] directly inferring 3D mesh coordinates.

In the following, we will report the reconstruction error, computed as the L2 distance between the estimated and the ground truth shapes (dimensionless for the synthetic results and in mm for the real ones). As common practice, the estimated meshes are aligned to the ground truth before evaluation using a Procrustes transformation. Additionally, in order to make a fair comparison, all methods requiring the pixels coordinates of the mesh, are fed with the estimates $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_{N_v})$ obtained with our network, augmented to a few hundreds of template-to-image correspondences by interpolation. We would like to point that our network produces an error of approximately 2 pixels in these 2D detections, and computing them using feature descriptors such as SIFT [28], generally led to worse results as these type of descriptors are prone to fail for non-textured surfaces with repetitive pattens and self-occlusions.

6.1. Evaluation on Synthetic Data

We evaluated all methods on a test set of our dataset consisting of 1208 independent samples generated with random values of shape and camera pose. These test samples are split into three subsets: 553 unknown shapes with a texture seen at training time ('Known Texture'), 553 unknown shapes with a texture not seen at training time ('New Texture'), and 102 unknown shapes without texture or very poorly textured ('Non-Textured'). Additionally we have simulated occlusions by covering the input images with a number of gray rectangular patches randomly distributed. Examples of the type of input images for each test case are shown in Fig. 3-Top.

Template-based analytical methods (Ba15Iso, Ba15Conf, Ch14IsoLsq and their iterative versions) were only evaluated on the textured and non-occluded cases, as they are methods that by construction can not realistically address the lack of texture or strong occlusions. Alternatively, to make the learning approaches (Resnet-50 V2 and DeformNet) robust to occlusions, the two networks were retrained with the 'gray-patched' images. No retraining was done to handle the lack of texture.

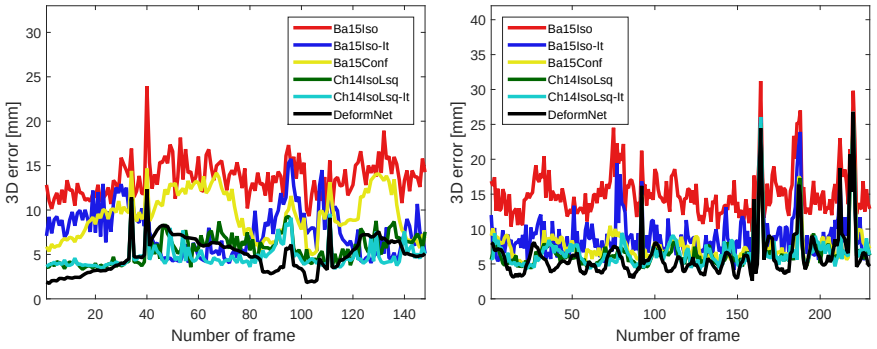
Table 1 summarizes the results of the synthetic evaluation. When dealing with textured and non-occluded images, Ch14IsoLsq-It is, as expected, the most accurate solution among the analytical methods. Regarding the learning approaches, Resnet-50 V2 turns to work very well under known textures. However, its performance suffers a big drop when dealing with textures not seen during training and with poorly textured surfaces. DeformNet performs consistently well in all situations, outperforming in all cases the analytical solutions. commentParticularly interesting is the case when dealing with new textures that are occluded, in which we obtain an accuracy very similar to the best analytical methods (we obtain 3.62mm versus 3.57mm for competing methods) when dense non-occluded correspondences are provided.

Figure 3 shows examples of the reconstructed meshes obtained by our approach. Note that when there are no occlusions, the recovered shape highly resembles the ground truth, even for non-textured surfaces and not previously seen textures. When the input image is corrupted by occlusions, our solutions turn to be noisier, but even in this case, they are very close to the ground truth.

Computation Times. Another advantage of learning based approaches is that once they are learned, they are much faster than the analytical solutions. The last column of Table 1 shows that computing the shape can be done in a fraction of a second for either Resnet-50 V2 and our approach, between one and two orders of magnitude faster than analytical methods.

6.2. Evaluation on Real Data

We also evaluate all methods on two real datasets provided by CVLab [48], which consist in video sequences of a bending paper and a deforming t-shirt, with 193 and 312 frames, respectively. As common practice, the back-



Method	Paper	T-Shirt
Ba15Iso	13.45	15.12
Ba15Iso-It	7.76	9.01
Ba15Conf	9.26	6.67
Ch14IsoLsq	5.66	6.66
Ch14IsoLsq-It	4.81	6.71
Resnet-50 V2	16.38	39.04
DeformNet	**4.79**	**6.11**

Figure 4. **Evaluation on the CVLab sequences [48].** The two graphs plot the 3D reconstruction error per frame (in mm) for all methods in the two real sequences (Left: Paper bending sequence, Right: T-shirt sequence). The results of Resnet-50 V2 are not plotted as it was not able to generalize to these sequences. **Right.** Mean reconstruction errors of all methods.

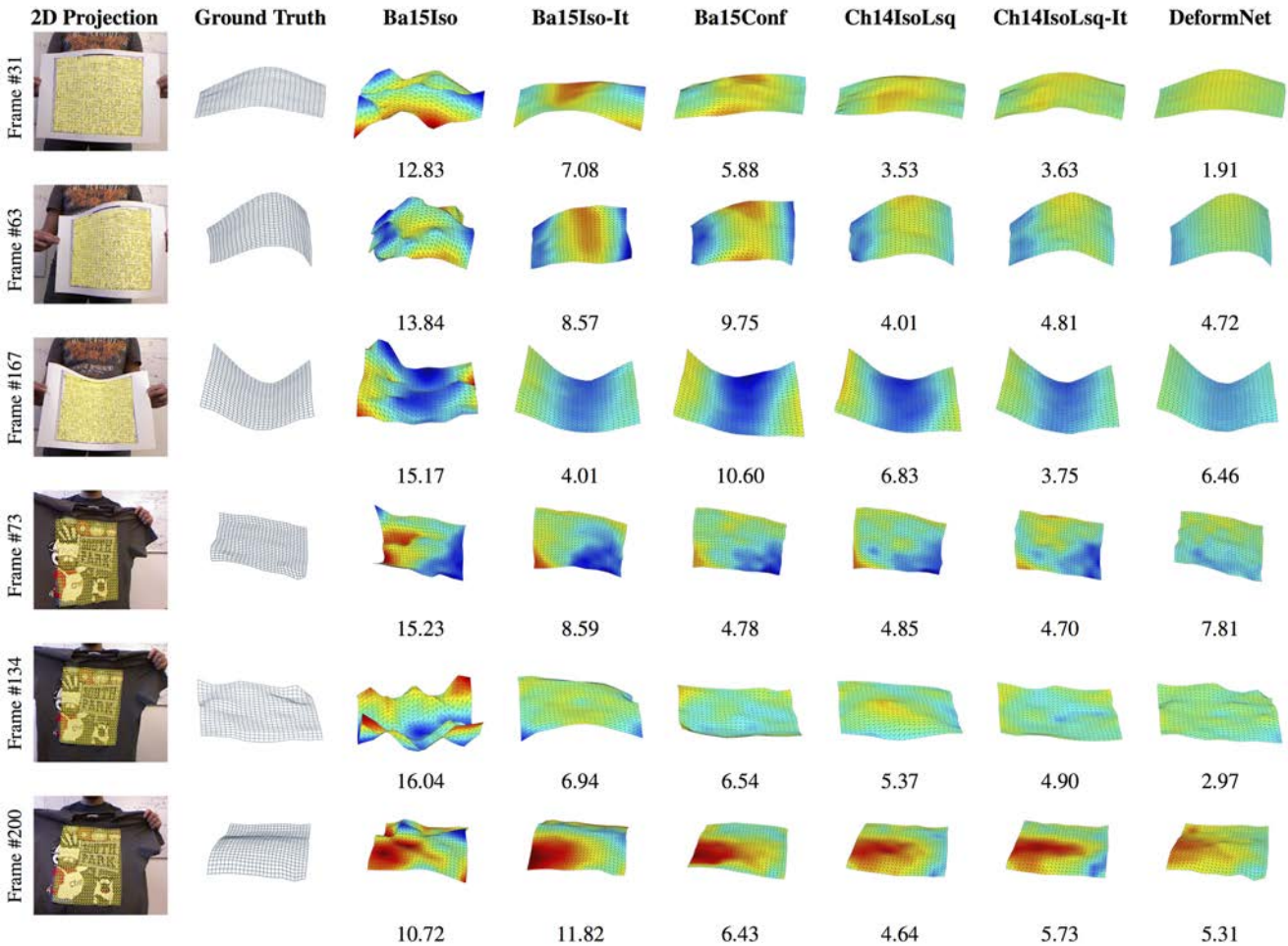


Figure 5. **Reconstructed meshes on the 'paper bending' and 't-shirt' CVLab sequences.** Results on Resnet-50 are not included as it did not generalize to real sequences. Each shape is color coded according to its reconstruction error. Larger errors appear in red, and small errors in dark blue. Below each reconstructed shape we indicate the mean reconstruction error (in mm).

ground of the sequences was subtracted. Additionally, both for Resnet-50 V2 and DeformNet, we performed a finetuning of the networks with a very small portion of the dataset (15% first frames). This finetuning was necessary to capture the bounds of the real deformations and adapt to the true illumination conditions that were not rendered by the synthetic dataset. In all methods we evaluated with the rest of the 85% of the frames. Again, for the fairness of com-

parison, the analytical solutions were fed by the 2D inputs of the mesh obtained by DeformNet, augmented to 500 correspondences using interpolation. The mean 2D location error (in pixels) obtained using DeformNet was 1.24 (paper bending sequence) and 2.28 (t-shirt sequence).

In Fig. 4 we plot the 3D reconstruction error per frame for all methods. The table on the right of the figure summarizes the results. Again, our DeformNet is the most accurate approach. In the bending paper sequence the analytic solution of Ch14IsoLsq-It is very close to ours, although DeformNet improves this method by a larger margin in the t-shirt sequence. In any event, recall that DeformNet performs inference per image in a fraction of a second while Ch14IsoLsq-It requires about 15 seconds. For these sequences, Resnet-50 V2, the other deep learning baseline we considered, performs very poorly demonstrating that the specific architecture we use in DeformNet allows for a much better generalization.

Finally, Fig. 5 shows a few reconstructed shapes obtained for each of the methods. Below each sample, we indicate the reconstruction errors. Note that samples with errors of about 4mm (in the paper bending sequence) or 6mm (in the t-shirt sequence) are already very good solutions. This is the magnitude of the error obtained by DeformNet.

6.3. Discussion

One of the most significant aspects of our network is its ability to generalize to unknown textures (see results in Table 1). We conjecture that this is the result of two factors: 1) training with a large variety of textures, and 2) separating the network into two input branches, one for performing 2D detection and the other to modulate input image features using the belief maps of the 2D detections. That is, our two branches allow us to correctly combine appearance and geometry. Note that the Resnet-50 V2 baseline we evaluated was also trained with a variety of textures, but it was not capable to generalize to new textures.

It is well known that on developable surfaces one may reconstruct shape from only the image boundaries [21]. One might therefore think that the robustness of DeformNet to new textures might be because our architecture learns to infer shape from the boundaries. In order to evaluate this, we performed the following experiment.

Blurred contours. In order to lower the dependency of DeformNet on the contours, we retrained it on a training set in which the surface boundaries of the input images were artificially corrupted by both adding random noise to the 2D coordinates of the boundary vertices and then blurring the contours. This strategy was also used in [26] to evaluate planar homographies. We then tested our architecture on the full dataset and obtained an error of 3.77mm, which is just slightly above the results reported in Table 1. Therefore, we can conclude that our network does not highly depend on the boundaries and exploits the whole image.

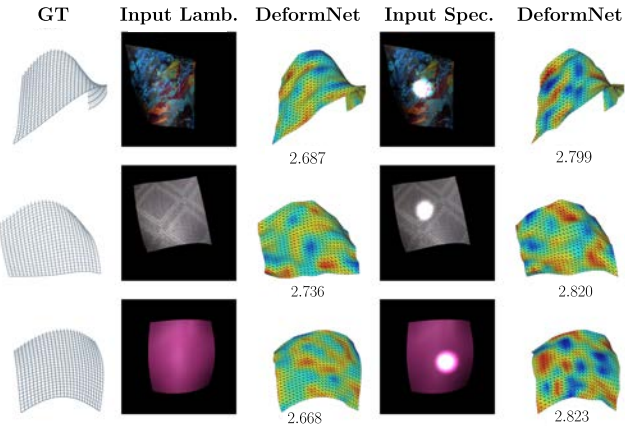


Figure 6. **Reconstruction under artificial specularities.** As in Fig. 5, each shape is color coded according to its reconstruction error.

Relaxing Lambertian reflectance assumptions. To further test our model limits Fig. 6 presents an evaluation of the model under synthetic specularities. The network also shows robustness to this scenario, and the overall reconstruction error (2.82) remains very similar to the case with Lambertian assumptions.

7. Conclusion

We have proposed the first deep network that estimates the 3D shape of a non-rigid surface from a single image. For this purpose we have designed an architecture that can be trained in an end-to-end manner, but that internally splits the problem in three stages: 2D detection, depth estimation and shape inference. The three stages are intimately connected and are executed by ensuring the satisfaction of geometric constraints such as correct 3D-to-2D reprojection and 3D-to-3D alignment between the estimated and the ground truth shapes. In order to train this network, we have rendered a large synthetic dataset of shapes under different levels of deformation, varying textures, material properties and illumination conditions. We have shown this network to outperform existing analytical solutions while being much more efficient, being able to tackle situations with large amounts of occlusion and very poorly textured surfaces. As part of future work, we aim at extending this solution to more complex deformations and further exploring the connections of our solution with analytic photometric methods.

Acknowledgments: This work is supported in part by a Google Faculty Research Award, by the Spanish Ministry of Science and Innovation under projects HuMoUR TIN2017-90086-R, ColRobTransp DPI2016-78957 and María de Maeztu Seal of Excellence MDM-2016-0656; and by the EU project AEROARMS ICT-2014-1-644271. We also thank Nvidia for hardware donation.

References

[1] A. Agudo, J. M. M. Montiel, L. Agapito, and B. Calvo. Modal space: A physics-based model for sequential estimation of time-varying shape from monocular video. *JMIV*, 57(1):75–98, 2017.

[2] A. Agudo and F. Moreno-Noguer. Learning shape, motion and elastic models in force space. In *ICCV*, 2015.

[3] A. Agudo and F. Moreno-Noguer. Simultaneous pose and non-rigid shape with particle dynamics. In *CVPR*, 2015.

[4] A. Agudo and F. Moreno-Noguer. DUST: Dual union of spatio-temporal subspaces for monocular multiple object 3D reconstruction. In *CVPR*, 2017.

[5] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *TPAMI*, 33(7):1442–1456, 2011.

[6] A. O. Balan, M. J. Black, H. Haussecker, and L. Sigal. Shining a light on human pose: On shadows, shading and the estimation of pose and shape. In *ICCV*, 2007.

[7] A. Bansal, X. Chen, B. Russell, A. G. Ramanan, et al. Pixelnet: Representation of the pixels, by the pixels, and for the pixels. *arXiv preprint arXiv:1702.06506*, 2017.

[8] A. Bansal, B. Russell, and A. Gupta. Marr revisited: 2D-3D alignment via surface normal prediction. In *CVPR*, 2016.

[9] A. Bartoli, Y. Gérard, F. Chadebecq, T. Collins, and D. Pizarro. Shape-from-template. *TPAMI*, 37(10):2099–2118, 2015.

[10] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, 1999.

[11] F. Brunet, R. Hartley, A. Bartoli, N. Navab, and R. Malgouyres. Monocular template-based reconstruction of smooth and inextensible surfaces. In *ACCV*, 2010.

[12] A. Chhatkuli, D. Pizzaro, and A. Bartoli. Stable template-based isometric 3d reconstruction in all imaging conditions by linear least-squares. In *CVPR*, 2014.

[13] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *TPAMI*, 23(6):681–685, 2001.

[14] E. A. Coutsias, C. Seok, and K. A. Dill. Using quaternions to calculate rmsd. *JCC*, 25(15):1849–1857, 2004.

[15] D. Dai, H. Riemenschneider, and L. Van Gool. The synthesizability of texture examples. In *CVPR*, 2014.

[16] M. de La Gorce, N. Paragios, and D. J. Fleet. Model-based hand tracking with texture, shading and self-occlusions. In *CVPR*, 2008.

[17] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[18] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014.

[19] R. Garg, G. Carneiro, and I. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016.

[20] C. Godard, O. M. Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[21] N. Gumerov, A. Zandifar, R. Duraiswami, and L. S. Davis. Structure of applicable surfaces from single views. In *ECCV*, 2004.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.

[23] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[24] Y. Kita. Elastic-model driven analysis of several views of a deformable cylindrical object. *TPAMI*, 18(12):1150–1162, 1996.

[25] M. Lee, C. H. Choi, and S. Oh. A procrustean markov process for non-rigid structure recovery. In *CVPR*, 2014.

[26] S. Lieberknecht, S. Benhimane, P. Meier, and N. Navab. A dataset and evaluation methodology for template-based tracking algorithms. In *ISMAR*, 2009.

[27] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *TPAMI*, 38(10):2024–2039, 2016.

[28] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[29] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3D human pose estimation. *ICCV*, 2017.

[30] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004.

[31] T. McInerney and D. Terzopoulos. A finite element model for 3D shape reconstruction and nonrigid motion tracking. In *ICCV*, 1993.

[32] T. McInerney and D. Terzopoulos. A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis. *CMIG*, 19(1):69–83, 1995.

[33] D. Metaxas and D. Terzopoulos. Constrained deformable superquadrics and nonrigid motion tracking. In *CVPR*, 1991.

[34] F. Moreno-Noguer. 3D human pose estimation from a single image via distance matrix regression. In *CVPR*, 2017.

[35] F. Moreno-Noguer and P. Fua. Stochastic exploration of ambiguities for nonrigid shape recovery. *TPAMI*, 35(2):463–475, 2013.

[36] F. Moreno-Noguer and J. M. Porta. Probabilistic simultaneous pose and non-rigid shape recovery. In *CVPR*, 2011.

[37] F. Moreno-Noguer, M. Salzmann, V. Lepetit, and P. Fua. Capturing 3D stretchable surfaces from single images in closed form. In *CVPR*, 2009.

[38] J. Östlund, A. Varol, D. T. Ngo, and P. Fua. Laplacian meshes for monocular 3D shape recovery. In *ECCV*, 2012.

[39] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3D human pose. *arXiv preprint arXiv:1611.07828*, 2016.

[40] M. Perriollat, R. Hartley, and A. Bartoli. Monocular template-based reconstruction of inextensible surfaces. *IJCV*, 95(2):124–137, 2011.

[41] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *CVPR*, 2016.

[42] M. Salzmann and P. Fua. Reconstructing sharply folding surfaces: A convex formulation. In *CVPR*, 2009.

[43] M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua. Closed-form solution to non-rigid 3D surface registration. *ECCV*, 2008.

[44] M. Salzmann, R. Urtasun, and P. Fua. Local deformation models for monocular 3D shape recovery. In *CVPR*, 2008.

[45] A. Tewari, M. Zollhöfer, H. Kim, P. Garrido, F. Bernard, P. Pérez, and C. Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. *arXiv preprint arXiv:1703.10580*, 2017.

[46] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 9(2):137–154, 1992.

[47] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: estimating shape and motion with hierarchical priors. *TPAMI*, 30(5):878–892, 2008.

[48] A. Varol, M. Salzmann, P. Fua, and R. Urtasun. A constrained latent variable model. In *CVPR*, 2012.

[49] S. Vicente and L. Agapito. Soft inextensibility constraints for template-free non-rigid reconstruction. *ECCV*, 2012.

[50] X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *CVPR*, 2015.

[51] X. Wang, M. Salzmann, F. Wang, and J. Zhao. Template-free 3D reconstruction of poorly-textured nonrigid surfaces. In *ECCV*, 2016.

[52] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[53] R. White and D. A. Forsyth. Combining cues: Shape from shading and texture. In *CVPR*, 2006.

[54] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. *arXiv preprint arXiv:1704.02157*, 2017.