This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Image-Guided Shape-from-Template Using Mesh Inextensibility Constraints

Thuy Tran Ruochen Chen Shaifali Parashar
CNRS, École Centrale de Lyon, INSA Lyon, Université Claude Bernard Lyon 1, LIRIS, UMR5205
{dinh-vinh-thuy.tran, ruochen.chen, shaifali.parashar}@liris.cnrs.fr

Abstract

Shape-from-Template (SfT) refers to the class of methods that reconstruct the 3D shape of a deforming object from images/videos using a 3D template. Traditional SfT methods require point correspondences between images and the texture of the 3D template in order to reconstruct 3D shapes from images/videos in real time. Their performance severely degrades when encountered with severe occlusions in the images because of the unavailability of correspondences. In contrast, modern SfT methods use a correspondence-free approach by incorporating deep neural networks to reconstruct 3D objects, thus requiring huge amounts of data for supervision. Recent advances use a fully unsupervised or self-supervised approach by combining differentiable physics and graphics to deform 3D template to match input images. In this paper, we propose an unsupervised SfT which uses only image observations: color features, gradients and silhouettes along with a mesh inextensibility constraint to reconstruct at a $400\times$ *faster pace than (best-performing) unsupervised SfT. Moreover, when it comes to generating finer details and severe occlusions, our method outperforms the existing methodologies by a large margin. Code is available at https://github.com/dvttran/nsft.*

1. Introduction

Recovering 3D shapes of objects from monocular RGB images, obtained through a video sequence or wide-baseline viewpoints, is a key goal in 3D computer vision. Given a 3D template of an object, comprising a known geometric configuration of the object along with known texture details, [3, 44] established that a unique 3D shape can be recovered from a single image of an object in a configuration different from the 3D template. It showed that combining projective geometry of the calibrated images with isometric (or geodesic-preserving) constraints on the 3D template deformation, the underlying 3D structure of any point in the image can be uniquely identified merely from the point-wise registration between the input image and the texture map associated with the 3D template. Template-based 3D reconstruction from images, a.k.a SfT, is therefore proven to be a well-posed problem with reliable solutions [8] that can perform real-time 3D reconstructions [9, 32, 47, 50]. [31] extends the traditional SfT to adapt to handle occlusions by interpolating the non-corresponding points on images from the reconstructed 3D points of the relevant neighbors. However, all these methods fail when the deforming object causes large (self-)occlusions, sharp motion, and high perspectivities in the images, such as in the case of the deforming cloth shown in Figure 1. This is because a unique solution of 3D shape is possible only if the point correspondences between the template texture map and the input images are known. Such occlusion artifacts appear quite often while capturing deforming objects; thus the inefficacy of traditional SfT to process them poses a severe limitation to their applicability. Sharp motions and strong perspectivities also significantly degrade the reliability of point correspondences which causes these methods to fail. To mitigate this issue, several modern SfT methods [10–12, 25, 37, 48] use deep neural networks (DNN) that can jointly perform template-image registration and 3D reconstruction. These methods may require a large amount of

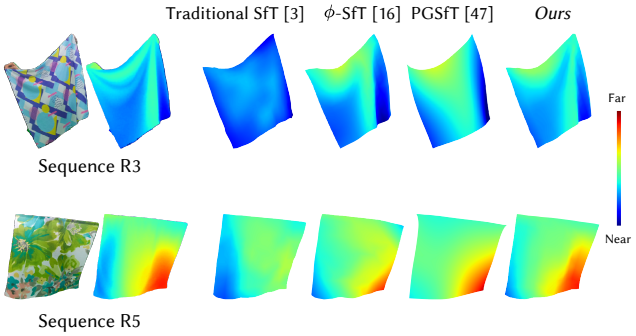

Figure 1. Image-guided SfT vs SoTA. Our method recovers finer details, resolves self-occlusions and handles strong perspectivities better than existing methods. We evaluated traditional SfT [3] with correspondences obtained from Cotracker v3 [18].

data for supervision and/or several restrictions on the scene and camera. However, they fail to recover finer details and severe occlusions.

Alternatively, ϕ-SfT [17] proposed to take advantage of the physics-based simulation of thin-shell objects [22, 30] to deform a 3D template and render using differentiable graphics to match input images in a complete unsupervised fashion. Although successful in handling fine details and severe occlusions to some extent, it incurs a huge computational cost, which makes it practically limited. Recently, PGSfT [49] used self-supervised learning of physics-based thin-shell simulations [6, 45] to learn a neural cloth model which allowed a $400\times$ speedup over [17], although with a degraded performance in handling severe occlusions and generating fine details.

In this paper, like [17], we present a correspondence-free, unsupervised SfT technique to reconstruct 3D shapes. However, instead of using physics-based simulation of thin shell objects, we rely on simple image-related measures only to guide the 3D template mesh deformations. Consequently, we achieve a $400\times$ computational speed up with a significantly improved performance than [17], especially while recovering finer details and challenging motion. Moreover, we initialize the simulation of a given video frame with its previous one, allowing a simpler and faster prediction of the current shape. Furthermore, we enforce an inextensibility measure of the template mesh rather than a strict isometric deformation [3] to allow the template to change its geometry according to the deformations observed in the images. Such a non-strict measure allows us to deal with isometric objects such as a paper and elastic objects such as clothes equally well. Our experiments show that the proposed method outperforms traditional and modern methodologies by large margins, especially while reconstructing finer details and severe occlusions.

2. Related Works

Traditional SfT methods. Given the registration between images and the texture map of the 3D template, these methods reconstruct surfaces as seen in input images assuming certain constraints on the deformations. [4, 36, 43, 44] assume that the template is represented with a mesh and enforce inextensibility constraints on its deformations to reconstruct the surfaces. [3, 8] assume a continuous representation of the template and enforce strict isometric constraints on its deformations. The reconstructions are obtained as a solution to a Partial Differential Equation. These methodologies have been extended to other deformation models such as conformality [3], equiareality [5, 34], ARAP [33], smoothness [55] and elasticity [1, 14, 26–28] in order to incorporate more realistic deformation scenarios. These methods are mostly computationally efficient and perform well in generic scenarios, but they are prone to high errors when encountered with strong deformations, severe occlusions, and lighting changes in the images. Under these conditions, the point correspondences obtained from the registration are highly inaccurate which contributes strongly to the degradation in their performance. [23, 32] attempted to jointly estimate the shape and registration in order to overcome these issues; however, they could not achieve any significant performance improvement.

DNN-based SfT methods. [12, 37] consider the template to be a flat, regular mesh and estimate the displacement of 3D vertices to comply with observed images using an encoder-decoder network. These methods learn object-specific models that rely on both registration and reconstructions as ground truth for the supervision. Such data is difficult to obtain, and the learned models are limited to the images seen in training scenarios for which registration could be computed. Therefore, these methods do not generalize well and do not handle complex deformation scenarios. [11] proposed an object-specific model that can be learned without using registration data as labels and is not limited to flat surfaces. It trains an encoder-decoder network from a large amount of deformed shapes (typically 100K samples) of the given object. It can reconstruct volumetric objects as well using an additional RGBD sensor as an input to monitor the non-visible surface. It can generalize better to unseen object configurations than [12, 37], but cannot handle severe occlusions and lighting changes. [25] simplifies [11] by restricting the training to learn registration. Instead of relying on ground truth optical flow for training as in [12, 37], it uses an off-the-shelf optical flow method to calculate training registration and imposes external losses to control the quality of tracks. [10, 48] extend [12] and [11] respectively by training networks with fixed object geometry but variable textures; thus improving the models from strictly object-specific to only geometry-specific. However, this does not significantly improves performance. All these methods cannot handle strong deformations, sharp motions, severe occlusions and fail to generate finer details.

Inspired by physics-based simulation [21, 22, 30, 38] and recent advances in physics recovery from videos [13, 16, 20, 41, 52], [17] proposed to solve SfT using differentiable physics and graphics to recover template deformation/motion according to input images in an unsupervised fashion. This approach does not require/compute registration, and recovers a decent 3D shape even in the cases of strong deformations and severe occlusions. A huge downside is that it is prohibitively slow: it can take upto 30 hours to process 50-60 images. Measures have been proposed to speed-up the computation; however they are based upon reducing the quality of 3D template mesh which significantly affects the quality of reconstruction. Using the strategy proposed in [2] to transform physics-based simulation into an optimisation problem, [6, 45] perform a self-

supervised learning of clothed garments under various motions. [49] adapted this methodology to solve SfT using self-supervised learning of template deformations. It learns a mesh-specific and a motion-specific model with a $400\times$ speedup over [17]; however at a degraded performance.

In this paper, we simplify the unsupervised physics-based simulation in [17] to achieve a $400\times$ speedup while outperforming it by large margins, even in complex deformations scenario. Instead of estimating the physical parameters related to stress, strain and bending energies, we pose the template deformation as an offset prediction of template mesh's vertices. This is a widely used approach in computer graphics [6, 45]. Instead of directly manipulating the physical forces to achieve equilibria in the above-mentioned energies, we rely solely on image observations to indirectly find the equilibrium matching the input images. In addition, we force mesh inextensibility on the template as a regularizer. It allows us to deal with both isometric and elastic surfaces, allowing mesh extensions pertaining to input images.

Summary of contributions.

1) We propose a simple, image-guided framework to simulate deformations using only vision cues (color, edges, gradients) to recover the motion as opposed to explicit estimation of physical properties.

2) We model deformations with mesh offset prediction using a deformation network. It is a compact, low-dimensional module that captures the essential deformations while naturally leaving out redundant learning variables associated with explicit physics-based learning.

3) We use an adaptive data loss structure to compute vision losses which allows us to effectively handle shading variations arising due to deformations. It is crucial for reconstructing finer details.

4) We propose a simplified optimization strategy that reduces computational complexity from $O(T^2N)$ in [17, 49] to $O(TN)$, where T is the number of frames in the length of the video sequence and N is average number of iterations needed to reconstruct a single frame. Moreover, our frame-wise optimization implicitly enforces temporal coherence rather than optimizing temporal losses additionally.

5) Our method outperforms SoTA by large margins in terms of speed and accuracy. It handles complex deformation scenarios extremely well. We show the strengths of our method, optimization strategy and data loss structure with extensive evaluation.

3. Image-guided SfT

Figure 2 shows our pipeline. We consider template as a textured triangle mesh $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{T})$ with topological connectivity given by the edges $\mathrm{e} \in \mathcal{E}$, the triangular faces $\mathrm{f} \in \mathcal{F}$ and texture map $\mathcal{T}$. The geometry of the mesh is the set of vertex coordinates $\mathbf{x} = \{x_{\mathrm{v}} \in \mathbb{R}^3 \mid \mathrm{v} \in \mathcal{V}\} \in \mathbb{R}^{|\mathcal{V}|\times 3}$. We consider that the 3D shape at the initial frame is a rigid transformation of the template. We denote $\mathbf{x}_0$ for the template and $\mathbf{x}_t$, for $t = 1, \ldots, T$ the shapes to be reconstructed. In addition, we assume that the video input is accompanied by the camera calibration, the intrinsic matrix $\mathbf{K} \in \mathbb{R}^{3\times 3}$. Since we are reconstructing as per camera frame, we do not require camera extrinsics to be known.

3.1. Deformation Model

Surface moves according to a deformation field $\varphi_t : \mathbb{R}^3 \to \mathbb{R}^3$. To reconstruct a surface shape, we need to learn such a vertex-wise mapping and denote $\mathbf{x}_t := \varphi(\mathbf{x}_0; \mathbf{z}_t)$ for some latent variable $\mathbf{z}_t$. Unlike physics-based methods that explicitly optimize the physics parameters, e.g. scene or material conditions, etc., we parametrize the surface deformation via a *deformation network* $f_\theta(\mathbf{x}_0, t)$ such that

$$\mathbf{x}_t = \mathbf{x}_0 + f_\theta(\mathbf{x}_0, t). \tag{1}$$

Such a neural network inherently provides a continuous mapping from the domain of vertex coordinates (and time) to their corresponding displacements. Moreover, it allows us to transfer the learning in $\mathbf{z}_t$ to learning the network parameters, which enables an adjustable balance between expressivity and computational efficiency by tuning the network architecture. We refer to Appendix 6 for discussion related to the advantages of our choice of deformation modeling with respect to the well established ones, such as vertex offset prediction, physics-based modeling, etc.

3.2. Differentiable Renderer

Once the camera calibration is known, we can utilize off-the-shelf differentiable renderers [19, 39] to optimize the shapes by aligning the vision cues within video sequence. We can see such renderers as differentiable functions that map the shape $\mathbf{x}_t$ to an RGB image $\hat{I}_t$ and a silhouette $\hat{S}_t$. We note that the renderers require the mesh to be textured, i.e. a texture image $\mathcal{T}$ and its corresponding UV map in the normalized coordinate space $[0, 1]^2$ are available.

3.3. Data Loss Structure

Before diving into the explicit loss functions, we explain the data loss structure common to all vision losses. We express the *adaptive* data loss between a pair of prediction and ground-truth data as

$$\ell(\hat{y}, y) = w(|\hat{y} - y|) \odot \ell_p(\hat{y}, y), \tag{2}$$

where $\odot$ denotes the element-wise product, ℓ_p is the ℓ-p norm and the weighting factor

$$w(d) = \alpha \exp\left(\frac{d}{\sigma}\right), \tag{3}$$

controlled by the hyper-parameters α and σ. This scheme amplifies element-wise errors exponentially, guiding the optimizer to focus on significant discrepancies.

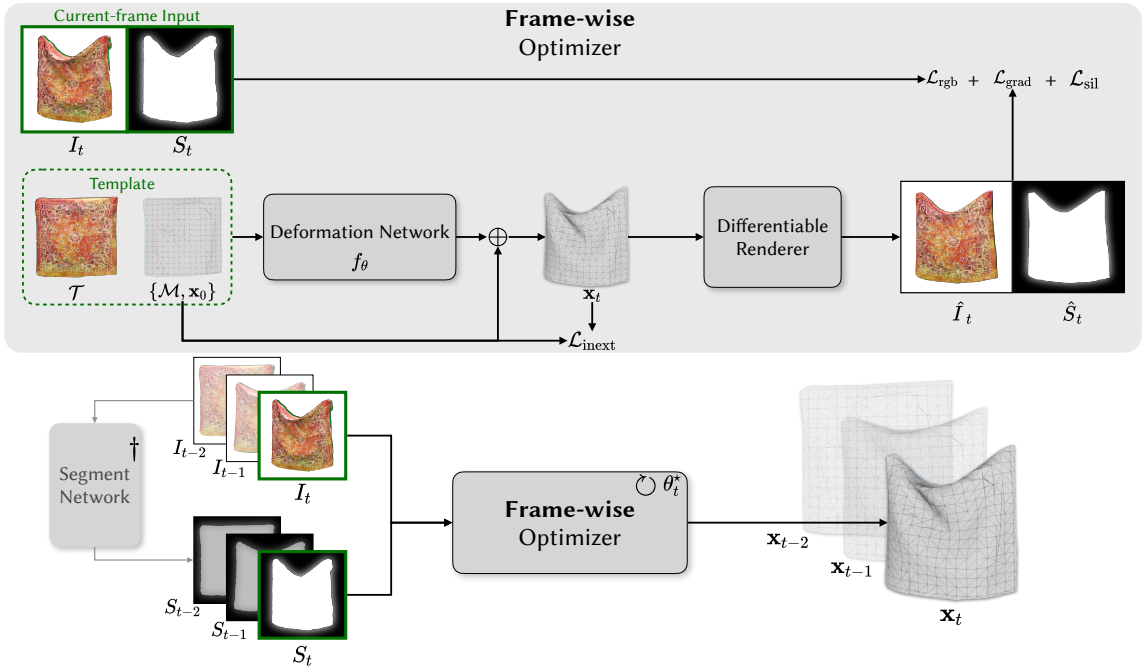

Figure 2. **Overview of our frame-wise image-guided SfT pipeline.** The template is provided with a texture map $\mathcal{T}$, the mesh connectivity $\mathcal{M}$ and the initial shape $\mathbf{x}_0$. We learn the reconstruction on a video sequence frame-by-frame. At current frame t, the deformation network f_θ predicts a displacement from $\mathbf{x}_0$ to produce the deformed shape $\mathbf{x}_t$. A differentiable renderer then projects this mesh to produce a rendered RGB image $\hat{I}_t$ and silhouette $\hat{S}_t$. We compute pixel-wise vision losses and a mesh inextensibility regularization, and optimize for the best parameters $\theta_t^\star$ via backpropagation. These parameters are passed forward to initialize the next frame's optimization (the symbol $\circlearrowright$). †If ground-truth is not available, an off-the-shelf segmentation network can be employed to generate the reference silhouette masks.

Such an adaptive loss is particularly useful for our vision losses. In practice, many renderers either neglect or approximate lighting effects, while real objects exhibit dynamic variations in brightness across video frames. Thus, the rendered surface can visually differ from the ground truth. However, the color mismatch typically remains more significant than any lighting discrepancies, allowing our adaptive loss to effectively handle these shading variations without sacrificing performance on regions with large errors.

3.4. Loss Functions

Like [17], we optimize the shapes using two main losses on RGB images and silhouettes (i.e., masks of the object in the frames). For in-the-wild videos, where the silhouettes might not be available, we propose to use Segment Anything Model 2 (SAM 2) [40] that can generate the object masks in images and videos from user prompts as point click or bounding box, etc.

RGB Loss. It measures the pixel-wise difference between the rendered frame $\hat{I}_t$ and the video frame $\tilde{I}_t$ that is masked to isolate the object of interest, has the form

$$\mathcal{L}_{\text{rgb}}(\mathbf{x}_t) = \ell(\hat{I}_t, I_t). \tag{4}$$

Silhouette Loss. The object position within the frame is further constrained by silhoutte loss. Using data loss ℓ as above, we define the silhouette loss as

$$\mathcal{L}_{\text{sil}}(\mathbf{x}_t) = \ell(\hat{S}_t, S_t), \tag{5}$$

where $\hat{S}_t$ is the rendered silhouette and S_t denotes the ground-truth mask. A Gaussian blur filter is also applied for smooth gradients on the boundaries, as in [17].

Image Gradient Loss. To exploit more information from the video, we propose to use the image gradient loss

$$\mathcal{L}_{\text{grad}}(\mathbf{x}_t) = \ell(k(\hat{I}_t), k(I_t)), \tag{6}$$

where $k(\cdot)$ denotes an image gradient operator (e.g., Sobel). By extracting first-order information from each image, this operator highlights edges and localized intensity variations, which can be crucial for guiding surface reconstructions on well-textured objects.

3.5. Mesh Inextensibility Regularization

To keep the optimization more stable, we use a mesh inextensibility introduced by [6], of the form

$$\mathcal{L}_{\text{inext}}(\mathbf{x}_t) = w_{\text{inext}} \sum_{\text{v} \in \mathcal{V}} |\det(\hat{C}_{t,\text{v}} - \Lambda_{0,\text{v}})|, \tag{7}$$

where $\hat{C}_{t,\mathrm{v}}$ represents the predicted covariance matrix at vertex $x_{t,\mathrm{v}}$ about its neighborhood $\mathcal{N}_{t,\mathrm{v}}$ with mean $\bar{x}_{t,\mathrm{v}}$ given by

$$\hat{C}_{t,\mathrm{v}} = \frac{1}{|\mathcal{N}_{t,\mathrm{v}}|} \sum_{\mathrm{n} \in \mathcal{N}_{t,\mathrm{v}}} (x_{t,\mathrm{n}} - \bar{x}_{t,\mathrm{v}})(x_{t,\mathrm{n}} - \bar{x}_{t,\mathrm{v}})^\top, \quad (8)$$

and the matrix $\Lambda_{0,\mathrm{v}}$ is derived from the eigenvalues of the template covariance matrix $C_{0,\mathrm{v}}$ as

$$\Lambda_{0,\mathrm{v}} = \begin{bmatrix} \lambda^1_{0,\mathrm{v}} & 0 & 0 \\ 0 & \lambda^2_{0,\mathrm{v}} & 0 \\ 0 & 0 & \lambda^3_{0,\mathrm{v}} \end{bmatrix}. \quad (9)$$

The weighting factor w_{inext} is computed adaptively with respect to the mesh scale.

3.6. Frame-wise Optimization

As discussed in Section 3.1, our goal is to learn the optimal parameters of the deformation network to reconstruct the shape $\mathbf{x}_t$, using the loss functions and regularization above. Our main idea is to reconstruct the object shapes for each frame independently. In particular, for each frame t, we seek the parameters $\theta_t^\star$ such that the reconstructed surface $\mathbf{x}_t = \mathbf{x}_0 + f_{\theta_t^\star}(\mathbf{x}_0, t)$ minimizes the losses aforementioned. Once attained, the parameters $\theta_t^\star$ is passed as the initialization for the next frame, $\theta_{t+1}^{(0)} = \theta_t^\star$. This choice benefits from temporal continuity in video data, where consecutive frames tend to have similar displacements, thus requiring relatively small adjustments to the network. This frame-wise scheme is the key to our significant speedup, especially for long video sequences. More details on complexity are described in Appendix 8.1.

4. Experiments

We implemented our framework in *PyTorch* [35], leveraging automatic differentiation to use gradient-based optimization to learn the deformation network, and the *libigl* [15] library for geometry processing on mesh. We used the *nvdiffrast* [19] to render as it is faster than *PyTorch3D* [39].

We specify here the default setting for most of our experiments. Our base network is a simple multi-layer perceptron (MLP) with ReLU activations and 8 hidden layers of width 256 for all video sequences. The frame-wise optimization strategy is employed with AdamW optimizer [24], learning rate of 10^{-4} and weight decay of 10^{-2}. The learning process warms up with 500 iterations and then runs 200 iterations for each frame. The data loss in Equations (2) and (3) is used for $\mathcal{L}_{\text{rgb}}$, $\mathcal{L}_{\text{sil}}$ and $\mathcal{L}_{\text{grad}}$ with $\alpha = 10$ and $\sigma = 1$. The Sobel operator is used on both first- and second-order to compute $\mathcal{L}_{\text{grad}}$ using the *Kornia* library [42]. The experiments are evaluated on a single NVIDIA V100 GPU.

We evaluated our method on Kinect-paper dataset [51] with 193 images and depth maps recorded using Kinect v1. We used SAM2 to generate image masks. We also experimented with ϕ-SfT dataset [17]. It has 4 synthetic sequences of cloth deformations simulated using ArcSim [22] and 9 real sequences cloth deformation with various shapes and textures recorded from Azure Kinect v2 camera. The texture and template are provided as 3D surface corresponding to first frame in the sequence. Then, we provide an ablation study on various choices of attributes and architectures used in our method.

Table 1. Depth map RMSE comparison on Kinect Paper dataset.

	DeepSfT [11]	TD-SfT [10]	SfT [3]	*Ours*
RMSE (mm)	6.97	3.37	6.17	4.01

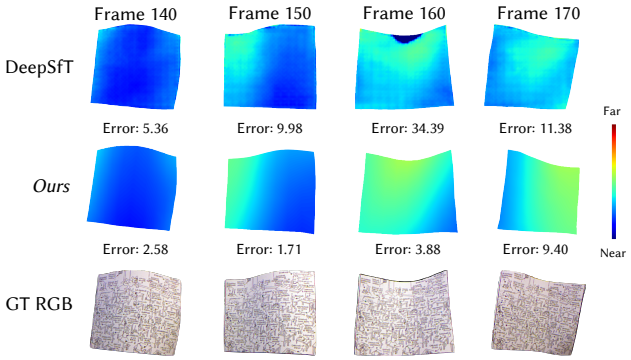

Figure 3. Visual comparison of DeepSfT vs *Ours* on Kinect Paper dataset. Depth map error at each frame is reported in mm.

4.1. Evaluation on Kinect Paper Dataset

Table 1 compares quantitative results for traditional SfT [3], supervised SfT: DeepSfT [11] and TD-SfT [10], and unsupervised ϕ-SfT [17], with our method outperforming DeepSfT and traditional SfT. We did not compute results for ϕ-SfT [17] due to its high computation time. PGSFT [49] is designed to work with square meshes only. We could not evaluate Kinect-paper sequence on it. Selected reconstructed frames are shown in Figure 3. The code for TD-SfT [10] is not publicly available, so we cannot show visual results. While DeepSfT [11] and TD-SfT [10] have almost real-time performance, *Ours* takes almost 2 seconds to process each frame. The traditional SfT [3] takes almost 10 seconds per frame. The correspondences were computed using CoTracker v3 [18].

4.2. Evaluation on Synthetic ϕ-SfT Dataset

We evaluated our method on 4 sequences of this dataset and compared with ϕ-SfT and PGSFT. The results are shown in Table 1. The errors are computed as relative 3D error between the ground truth and the reconstructed mesh. We could not run the supervised method, DeepSfT on this

Table 2. Quantitative comparison on ϕ-SfT synthetic dataset. Average error between ground-truth and predited meshes is reported.

Seq.	Traditional SfT [3]	ϕ-SfT [17]	PGSfT [49]	*Ours*
S1	0.0328	0.0420	0.0298	**0.0229**
S2	0.0483	**0.0230**	0.0420	0.0254
S3	0.0481	**0.0330**	0.0823	0.0357
S4	0.0232	0.0050	0.0919	**0.0031**

sequence as it requires to be trained separately on each sequence and the training requires almost 100K samples of simulated data. These sequences each have 50 frames only, which are insufficient to train DeepSfT. In the papers, DeepSfT and TD-SfT report limitations with large occlusions and have not shown any experiments with complex geometries. A recent, weakly-supervised SfT [25] also requires more than 50 frames to train. Thus, we do not make further comparison with supervised methodologies.

4.3. Evaluation on Real ϕ-SfT Dataset

We evaluate our method on 9 real sequences in the ϕ-SfT dataset and compare our performance with ϕ-SfT and PGSfT, the best performing state of the art methods. We compute errors as average chamfer distance between ground truth point clouds and predicted point clouds (recovered from sampling on the reconstructed meshes the same number of points as the ground-truth) on each video sequence. Table 3 shows that our method outperforms ϕ-SfT [17] and PGSfT [49] in all sequences by large margins. Our vision-based adaptive data losses allow us to recover finer details of the object motion and geometry.

ϕ-SfT [17] is computationally expensive. On average, it processes a single frame in $2800s$. Our runtime performance is comparable to PGSfT [49], which is a self-supervised technique. Our frame-wise optimization strategy, detailed in Section 3.6 simplifies optimization scheme used in ϕ-SfT and PGSfT by large margins. Table 4 shows that our runtime in minutes, over ϕ-SfT real sequences matches PGSfT [49]. We refer to Appendix 8.1 for more details on complexity comparison of these methods with respect to *Ours*.

Sharp folds, strong motion and high perspectives. Figure 1 shows some visual results on a sample frame from R3 and R5 sequence. *Ours* closely match the ground-truth folds and contours, outperforming the other approaches. The traditional SfT[3] underestimates surface variation in certain regions (e.g., the upper corner in R3) due to noises in correspondences computed from CoTracker v3 [18]. The physics-based methods ϕ-SfT and PGSfT are slightly better but they still fail to capture subtle curvatures accurately.

Figure 4 shows the 3D reconstruction of our method from frontal and side perspectives, together with the ones

Table 3. Quantitative comparison ϕ-SfT real dataset. Average chamfer distance between ground-truth and predicted points is reported. All values are multiplied by 10^4 to improve readability.

Seq.	ϕ-SfT [17]	PGSfT [49]	*Ours*
R1	9.36	6.05	**0.66**
R2	11.60	4.11	**1.30**
R3	6.76	10.41	**4.49**
R4	11.59	13.19	**8.15**
R5	14.93	11.97	**8.04**
R6	9.95	15.46	**3.37**
R7	12.34	7.20	**4.62**
R8	3.23	10.21	**2.76**
R9	3.71	9.18	**2.12**

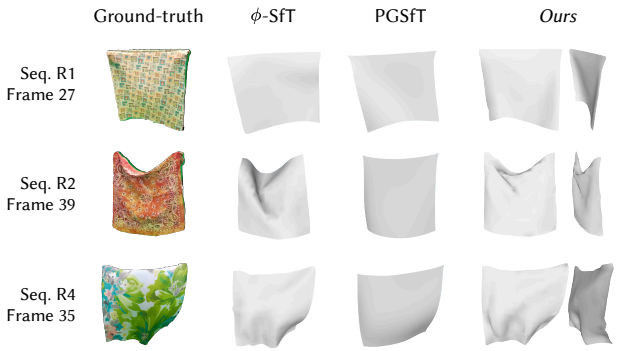

Figure 4. Qualitative comparison on ϕ-SfT real sequences.

from ϕ-SfT [17] and PGSfT [49]. In sequence R1, the deformation is relatively simple, with a straight upper boundary throughout the frames and only slight movement in the lower region. Hence, our approach achieves an accurate reconstruction with minimal difficulty. Although the rendered frame lacks shading effects, the side view reveals that our method successfully captures the surface folds, which are absent in the other two methods. The sequences R2 and R4 contain high-frequency movement, due to manual folding at the corners and strong wind turbulence, respectively. By using low-resolution templates in ϕ-SfT dataset [17], *Ours* captures fine-winkle and high-curvature details very well. Moreover, our reconstructions are smooth and physically plausible without using any smoothness regularization. These detailed reconstructions are attributed to deformation modeling, adaptive loss implementation strategy and efficient frame-wise optimisation, as discussed in Section 3. Compared to *Ours*, ϕ-SfT could reconstruct some folds but they are not well aligned with the object shapes. PGSfT, on the other hand, over-smoothed the surfaces and cannot capture any much detail.

Self-occlusions. Among ϕ-SfT real sequences, we observe that R3 and R6 have strongest self-occlusion in their final

Table 4. Runtime comparison with PGSfT [49]. All numbers represent the runtime (in minutes) for the optimization loop until convergence.

Method	R1	R2	R3	R4	R5	R6	R7	R8	R9
PGSfT [49]	2.62	2.37	2.39	2.32	2.61	2.36	2.41	2.38	2.31
Ours	3.58	2.88	2.75	2.69	2.21	2.91	2.83	2.76	2.63

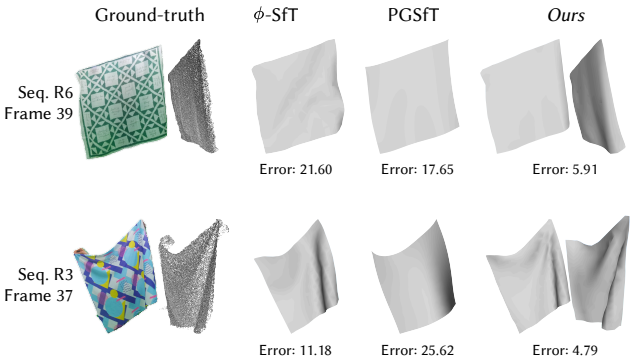

Figure 5. Comparison on frames with strongest self-occlusions on R6 and R3 sequences of the ϕ-SfT real dataset.

frames. When such behaviors occur, especially over a large number of frames, the surface reconstruction task becomes more challenging for most methods. An image-guided approach typically benefits from temporal constraints to handle the invisible regions. ϕ-SfT[17] shows that their physics-based model could provide reasonable results for self-occluded parts. Indeed, their incremental learning has built-in temporal consistency along with Newtonian dynamics. This is also incorporated into the pre-trained neural cloth model proposed by PGSFT [49]. However, training such a general-purpose model in self-supervised approach restricts the performance to having low-frequency wrinkles. In contrast, our frame-wise strategy employs no explicit temporal constraint yet still handles self-occlusion effectively. As illustrated in Figure 5, our reconstructed surfaces for both R3 and R6 align more closely to the ground-truth point clouds than the other physics-based methods. We attribute this capability to our initialization scheme in Section 3.6, in which each frame's deformation network starts from the suboptimal parameters of the preceding frame.

4.4. Ablation Study

Loss Functions. The RGB and silhouette loss, together with the mesh inextensibility is quite crucial for every SfT method. We have additionally imposed image gradient loss. In addtion, we have enforced all losses in an adaptive format. Thus, we evaluate importance of gradients and adaptive data loss in our default setting. We compare three configurations of our loss functions on ϕ-SfT real sequences [17], and report error as chamfer distance between ground truth and prediction. As shown in Table 5, removing the image gradient loss sometimes yields marginal improvements as in R2 and R3, but generally increases the error on other sequences. Similarly, omitting the adaptive data loss can help certain cases such as R7-R9 but often leads to higher overall error, especially in sequences R3 and R4. In contrast, our default setting, which employs both gradient and adaptive data losses, tends to strike a balanced compromise, achieving better or comparable results in most sequences. We study more about the adaptive data loss, silhouette loss and mesh inextensibility loss in Appendix 7.

Table 5. Ablation study evaluating the impact of removing image gradient and adaptive data loss terms on our performance. Average chamfer distance ($\times 10^4$) on ϕ-SfT real dataset is reported.

Seq.	No image grad.	No adaptive data loss	Default
R1	0.71	**0.52**	0.66
R2	**1.15**	1.44	1.30
R3	**4.23**	9.73	4.49
R4	8.61	11.1	**8.15**
R5	7.47	8.05	**8.04**
R6	4.09	5.85	**3.37**
R7	5.23	**4.37**	4.62
R8	2.36	**2.22**	2.76
R9	1.71	**1.56**	2.12
Avg.	3.95	4.98	**3.91**

Deformation Network. To evaluate the deformation model proposed in Section 3.1, we run our optimization on the ϕ-SfT real sequences under the default configuration, but replacing our deformation network with a vertex-offset parameterization as illustrated in first column of Figure 6. The resulting meshes contain noticeable non-smoothness compared to the deformation network outputs. We then incorporate the discrete thin-shell bending energy from [46] to examine whether this additional regularization can rectify the artifacts. Although it does yield smoother surfaces, the high-frequency folds are lost, indicating that the bending term enforces a trade-off during optimization. As mentioned in Section 3.1, our deformation network is naturally a continuous latent model that can capture the essential deformations and leave out redundant degrees of freedom that can cause unrealistic shapes.

Network Architecture. In addition to our base (8-layer,

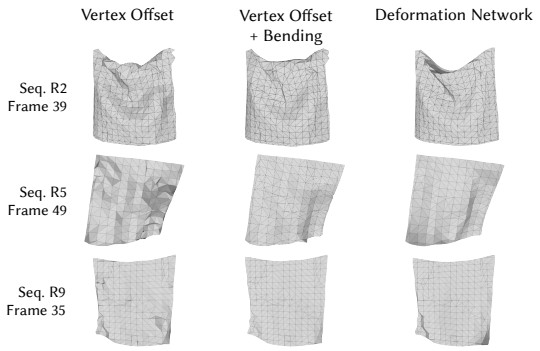

Figure 6. Qualitative comparison on modeling formulation.

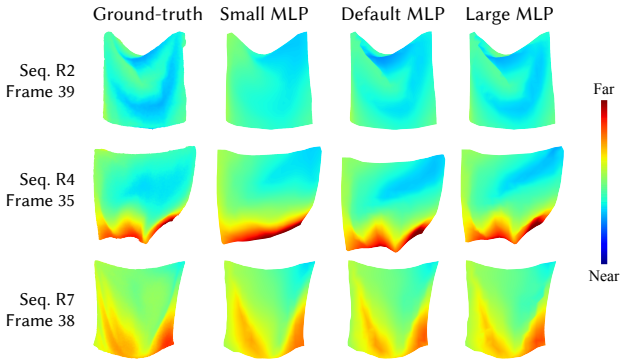

Figure 7. Qualitative comparison on network architecture.

256-width) deformation network, we analyze the network architecture with a small (4-layer, 64-width) and a large (12-layer, 512-width) version. We report the average reconstruction error in terms of chamfer distance between ground truth and predicted data of the three architectures on ϕ-SfT real sequences in Table 6. We can see that the larger networks generally give better performance due to their greater capacity, which might be able to reach a suboptimal solution more quickly. The small one, on the other hand, struggles to capture high-frequency wrinkles as shown in Figure 7 on sequences R2 and R4. For sequence R7, smaller networks perform much better quantitatively, where, in Table 6, we have 4.95 and 4.62 compared to 6.00 of the large one. We visually perceive that the motion of the cloth in this sequence is relatively simpler than others. Therefore, these observations suggests us to select the network architecture which aligns with the complexity of the object deformation within the video sequence. Moreover, to handle a more complex motion sequence, one can simply increase the network capacity without incurring a computational overhead. Hence, in our experiemnts, we have chosen between base and large MLP, depending on the complexity of the sequence. However, in sequences where large MLP outperforms our base one, the performance gap is not huge. It is important to note that learning on these three networks has almost no difference in runtime.

4.5. Limitations

We noticed that the performance on Kinect Paper dataset is slightly worse than TD-SfT [25] due to some dips while reconstructing frames with relatively high inter-image motion, as shown at frame 170 in Figure 3. In these cases, we need more than 200 iterations or other optimization strategies as discussed in Appendix 8. Moreover, although the adaptive data loss structure is designed from an intuition about shading, our formulation cannot reconstruct textureless or specular surfaces, which requires for light source and reflectance modeling in shape-from-shading literature [54].

Table 6. Quantitative comparison on network architecture. Average chamfer distance ($\times 10^4$) on ϕ-SfT real dataset is reported.

Seq.	Small	Base	Large
R1	0.75	0.69	**0.63**
R2	2.49	1.30	**0.96**
R3	4.00	4.49	**3.11**
R4	15.85	**8.15**	8.38
R5	9.97	**8.04**	9.11
R6	4.73	3.37	**3.02**
R7	4.95	**4.62**	6.00
R8	4.06	2.76	**2.15**
R9	3.74	2.12	**1.73**

5. Conclusion

In this paper, we presented an unsupervised approach to template-based 3D reconstruction using only vision cues (images, gradients and silhouettes) and mesh inextensibility regularization; thus providing the very first SfT solely guided by image observations. We introduced an adaptive strategy to enforce vision losses, a function-based deformation network to estimate deformations and an implicitly temporally-coherent frame-wise optimization strategy to devise a simple methodology which not only outperforms state-of-the-art methods by large margins but also achieves a $400\times$ faster computation than the best-performing counterpart, ϕ-SfT. We also tested several aspects of our proposed methodology and network architectures. Although our work focuses on triangle meshes, it is worth noting that other representations, such as parametric surfaces, implicit fields, or point-based structures, can also be adapted. We plan to incorporate them in future works. We also plan to extend our frame-wise to a window-wise approach, which allows us to constrain the object shapes with temporal consistency loss to improve the overall performance in terms of timing and accuracy. Moreover, we would like to discover other object categories that have servere self-occlusion.

Acknowledgments. This work is supported by the project RHINO (ANR-22-CE33-0014-01), a JCJC research grant. We also thank ϕ-SfT [17] and PGSfT [49] authors for releasing the dataset, the code, and their valuable discussions about the benchmarks.

References

[1] Antonio Agudo and Francesc Moreno-Noguer. Simultaneous pose and non-rigid shape with particle dynamics. In *CVPR*, 2015. 2

[2] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *SIGGRAPH*, 1998. 2

[3] Adrien Bartoli, Yan Gérard, Francois Chadebecq, Toby Collins, and Daniel Pizarro. Shape-from-template. *IEEE TPAMI*, 37(10):2099–2118, 2015. 1, 2, 5, 6

[4] F. Brunet, A. Bartoli, and R.I. Hartley. Monocular template-based 3d surface reconstruction: Convex inextensible and nonconvex isometric methods. *Computer Vision and Image Understanding*, 125:138–154, 2014. 2

[5] David Casillas-Perez, Daniel Pizarro, David Fuentes-Jimenez, Manuel Mazo, and Adrien Bartoli. Equiareal shape-from-template. *Journal of Mathematical Imaging and Vision*, 61(5):607–626, 2014. 2

[6] Ruochen Chen, Shaifali Parashar, and Liming Chen. Gaps: Geometry-aware, physics-based, self-supervised neural garment draping. In *International Conference on 3D Vision (3DV)*, 2024. 2, 3, 4

[7] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018. 1

[8] Ajad Chhatkuli, Daniel Pizarro, Adrien Bartoli, and Toby Collins. A stable analytical framework for isometric shape-from-template by surface integration. *IEEE TPAMI*, 39(5): 833–850, 2016. 1, 2

[9] Toby Collins and Adrien Bartoli. [poster] realtime shape-from-template: System and applications. In *2015 IEEE International Symposium on Mixed and Augmented Reality*, 2015. 1

[10] David Fuentes-Jimenez, Daniel Pizarro, David Casillas-Perez, Toby Collins, and Adrien Bartoli. Texture-generic deep shape-from-template. *IEEE Access*, 9:75211–75230, 2021. 1, 2, 5

[11] David Fuentes-Jimenez, Daniel Pizarro, David Casillas-Pérez, Toby Collins, and Adrien Bartoli. Deep shape-from-template: Single-image quasi-isometric deformable registration and reconstruction. *Image and Vision Computing*, 127: 104531, 2022. 2, 5

[12] Vladislav Golyanik, Soshi Shimada, Kiran Varanasi, and Didier Stricker. Hdm-net: Monocular non-rigid 3d reconstruction with learned deformation model. In *Virtual Reality and Augmented Reality*, 2018. 1, 2

[13] Jingfan Guo, Jie Li, Rahul Narain, and Hyun Soo Park. Inverse simulation: Reconstructing dynamic geometry of clothed humans via optimal control. In *CVPR*, 2021. 2

[14] Nazim Haouchine and Stephane Cotin. Template-based monocular 3d recovery of elastic shapes using lagrangian multipliers. In *CVPR*, 2017. 2

[15] Alec Jacobson, Daniele Panozzo, et al. libigl: A simple C++ geometry processing library, 2018. https://libigl.github.io/. 5

[16] Miguel Jaques, Michael Burke, and Timothy Hospedales. Physics-as-inverse-graphics: Unsupervised physical parameter estimation from video. In *ICLR*, 2020. 2

[17] Navami Kairanda, Edith Tretschk, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik. ϕ-sft: Shape-from-template with a physics-based deformation model. In *CVPR*, 2022. 2, 3, 4, 5, 6, 7, 9, 1

[18] Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024. 1, 5, 6

[19] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM TOG*, 39 (6):1–14, 2020. 3, 5

[20] Yue Li, Marc Habermann, Bernhard Thomaszewski, Stelian Coros, Thabo Beeler, and Christian Theobalt. Deep Physics-aware Inference of Cloth Deformation for Monocular Human Performance Capture . In *3DV*, 2021. 2

[21] Yifei Li, Tao Du, Kui Wu, Jie Xu, and Wojciech Matusik. Diffcloth: Differentiable cloth simulation with dry frictional contact. *ACM TOG*, 42(1), 2022. 2

[22] Junbang Liang, Ming Lin, and Vladlen Koltun. Differentiable cloth simulation for inverse problems. In *NeurIPS*, 2019. 2, 5

[23] Qi Liu-Yin, Rui Yu, Lourdes Agapito, Andrew Fitzgibbon, and Chris Russell. Better together: Joint reasoning for non-rigid 3d reconstruction with specularities and shading. In *BMVC*, 2016. 2

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[25] Sara Luengo-Sanchez, David Fuentes-Jimenez, Cristina Losada-Gutierrez, Daniel Pizarro, and Adrien Bartoli. Weakly-supervised deep shape-from-template. *IEEE Access*, 13:22868–22892, 2025. 1, 2, 6, 8

[26] Abed Malti and Cédric Herzet. Elastic shape-from-template with spatially sparse deforming forces. In *CVPR*, 2017. 2

[27] Abed Malti, Richard Hartley, Adrien Bartoli, and Jae-Hak Kim. Monocular template-based 3d reconstruction of extensible surfaces with local linear elasticity. In *CVPR*, 2013.

[28] Abed Malti, Adrien Bartoli, and Richard Hartley. A linear least-squares solution to elastic shape-from-template. In *CVPR*, 2015. 2

[29] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[30] Rahul Narain, Armin Samii, and James F. O'Brien. Adaptive anisotropic remeshing for cloth simulation. *ACM TOG*, 31 (6), 2012. 2

[31] Dat Tien Ngo, Sanghyuk Park, Anne Jorstad, Alberto Crivellaro, Chang D. Yoo, and Pascal Fua. Dense image registration and deformable surface reconstruction in presence of occlusions and minimal texture. In *ICCV*, 2015. 1

[32] Dat Tien Ngo, Jonas Östlund, and Pascal Fua. Template-based monocular 3d shape recovery using laplacian meshes. *IEEE TPAMI*, 38(1):172–187, 2016. 1, 2

[33] Shaifali Parashar, Daniel Pizarro, Adrien Bartoli, and Toby Collins. As-rigid-as-possible volumetric shape-from-template. In *ICCV*, 2015. 2

[34] Shaifali Parashar, Daniel Pizarro, and Adrien Bartoli. Local deformable 3d reconstruction with cartan's connections. *IEEE TPAMI*, 42(12):3011–3026, 2020. 2

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 5

[36] Mathieu Perriollat, Richard Hartley, and Adrien Bartoli. Monocular template-based reconstruction of inextensible surfaces. *IJCV*, 95(2):124–137, 2011. 2

[37] A. Pumarola, A. Agudo, L. Porzi, A. Sanfeliu, V. Lepetit, and F. Moreno-Noguer. Geometry-Aware Network for Non-Rigid Shape Prediction from a Single View. In *CVPR*, 2018. 1, 2

[38] Yi-Ling Qiao, Junbang Liang, Vladlen Koltun, and Ming C. Lin. Scalable differentiable physics for learning and control. In *ICML*, 2020. 2

[39] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 3, 5

[40] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 4

[41] Davis Rempe, Leonidas J. Guibas, Aaron Hertzmann, Bryan Russell, Ruben Villegas, and Jimei Yang. Contact and human dynamics from monocular video. In *ECCV*, 2020. 2

[42] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. Kornia: an open source differentiable computer vision library for pytorch. In *CVPR*, 2020. 5

[43] Mathieu Salzmann and Pascal Fua. Reconstructing sharply folding surfaces: A convex formulation. In *CVPR*, 2009. 2

[44] Mathieu Salzmann, Francesc Moreno-Noguer, Vincent Lepetit, and Pascal Fua. Closed-form solution to non-rigid 3d surface registration. In *ECCV*, 2008. 1, 2

[45] Igor Santesteban, Miguel A Otaduy, and Dan Casas. SNUG: Self-Supervised Neural Dynamic Garments. In *CVPR*, 2022. 2, 3

[46] Josua Sassen, Henrik Schumacher, Martin Rumpf, and Keenan Crane. Repulsive shells. *TOG*, 43(4):1–22, 2024. 7

[47] Mohammadreza Shetab-Bushehri, Miguel Aranda, Erol Özgür, Youcef Mezouar, and Adrien Bartoli. Robusft: Robust real-time shape-from-template, a c++ library. *Image and Vision Computing*, 141:104867, 2024. 1

[48] Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Didier Stricker. IsMo-GAN: Adversarial Learning for Monocular Non-Rigid 3D Reconstruction . In *CVPRW*, 2019. 1, 2

[49] David Stotko, Nils Wandel, and Reinhard Klein. Physics-guided shape-from-template: Monocular video perception through neural surrogate models. In *CVPR*, 2024. 2, 3, 5, 6, 7, 9, 1

[50] Dongliang Tan, Huamin Yang, Zhengang Jiang, Weili Shi, Jun Qin, and Feng Qu. Improved shape-from-template method with perspective space constraints for disappearing features. *Complex & Intelligent Systems*, 10(4):5475–5488, 2024. 1

[51] Aydin Varol, Mathieu Salzmann, Pascal Fua, and Raquel Urtasun. A constrained latent variable model. In *CVPR*, 2012. 5

[52] Sebastian Weiss, Robert Maier, Daniel Cremers, Rüdiger Westermann, and Nils Thuerey. Correspondence-free material reconstruction using sparse surface constraints. In *CVPR*, 2020. 2

[53] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *CVPR*, pages 20331–20341, 2024. 1

[54] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. *TPAMI*, 21 (8):690–706, 2002. 8

[55] Erol Özgür and Adrien Bartoli. Particle-sft: A provably-convergent, fast shape-from-template algorithm. *IJCV*, 123 (2):184–205, 2017. 2