SAFT: Shape and Appearance of Fabrics from Template via Differentiable Physical Simulations from Monocular Video

David Stotko Reinhard Klein

University of Bonn

Abstract

The reconstruction of three-dimensional dynamic scenes is a well-established yet challenging task within the domain of computer vision. In this paper, we propose a novel approach that combines the domains of 3D geometry reconstruction and appearance estimation for physically based rendering and present a system that is able to perform both tasks for fabrics, utilizing only a single monocular RGB video sequence as input. In order to obtain realistic and high-quality deformations and renderings, a physical simulation of the cloth geometry and differentiable rendering are employed. In this paper, we introduce two novel regularization terms for the 3D reconstruction task that improve the plausibility of the reconstruction by addressing the depth ambiguity problem in monocular video. In comparison with the most recent methods in the field, we have reduced the error in the 3D reconstruction by a factor of 2.64 *while requiring a medium runtime of* $30\,\mathrm{min}$ *per scene. Furthermore, the optimized motion achieves sufficient quality to perform an appearance estimation of the deforming object, recovering sharp details from this single monocular RGB video.*

1. Introduction

Shape-from-template (SfT) methods offer an elegant solution for reconstructing the 3D geometry of a scene by leveraging prior information in the form of a template and minimal setup requirements, such that a single RGB video is sufficient to tackle reconstruction tasks. We focus on the challenging task of reconstructing fabrics that deform due to external influences such as wind or human manipulation. While recent work has shown that incorporating physical simulations enhance the reconstruction process by enforcing realistic motion [27, 62], such simulations fall short in addressing depth ambiguity where multiple plausible deformations can result in identical RGB renderings. To address this challenge, we introduce two novel regularization terms that mitigate depth ambiguity, significantly improving the plausibility and quality of the reconstructed geometry. Our approach is evaluated against state-of-the-art methods, demonstrating superior reconstruction quality with a moderate runtime.

Furthermore, we demonstrate that, in contrast to previous methods, our precisely reconstructed geometry allows the estimation of a highly detailed Spatially Varying Bidirectional Reflectance Distribution Function (SVBRDF) together with the ambient lighting to reconstruct the appearance of the fabric. Correctly reconstructed dynamic deformations create slightly varying viewing angles and help to decouple diffuse, specular, and lighting contributions. In summary, we introduce a method that seamlessly combines 3D reconstruction of dynamic non-rigid objects with appearance and illumination reconstruction into a single pipeline, leveraging only a single monocular RGB video. We will present our results on a real dataset where no ground truth material properties are known, evaluate the quality using synthetic scenes and show the improvements compared to the blurred estimates when using the erroneous geometry of previous techniques [27, 62].

The source code is available at https://github.com/vc-bonn/saft.

2. Related Work

Non-rigid object reconstruction The presence of deformable objects introduces various challenges in the context of 3D object reconstruction tasks. A variety of algorithms are proposed based on the available data and the assumptions made. Non-rigid structure from motion (NRSfM) algorithms employ one or more cameras to reconstruct arbitrary deformed surfaces [60, 76] or specialize in human faces [6, 44] and thin objects like fabric [19, 59]. Higher quality can be achieved when using some template geometry, potentially including masks, material, textures, or other information to constrain the problem [54]. Such methods are called Shape-from-Template (SfT) [2, 58] and incorporate regularizing constraints, *e.g.* restrictions to isometric deformations resulting in a set of partial differential equations [2, 7]. More recent approaches model the deformation field by a neural network [11, 16, 47] and there-

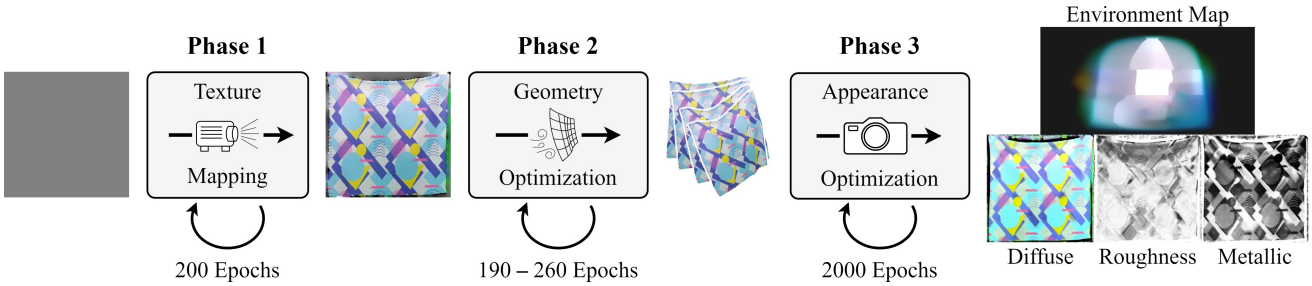


Figure 1. Overview of the three phases of our method. First, we perform projective texture mapping to obtain an accurate texture on the geometry template. Secondly, the precise geometry is reconstructed by optimizing the parameters of a physical simulation to fit rendered images to the target video sequence. Finally, the 3D deformation achieves sufficient quality that it can be used to reliably estimate fine details in appearance parameters and environment lighting from only one monocular RGB video by employing differentiable ray tracing.

fore get smooth and consistent motion. For a more detailed overview on non-rigid scene reconstruction, we refer to [67, 78].

Appearance estimation The task of optical material reconstruction heavily depends on the amount of information available. We want to focus on the estimation of SVBRDFs or parts of it for appearance reconstruction. Learning-based approaches are able to estimate the SVBRDF of a planar surface from a single image under various lighting conditions [13, 17, 34, 74, 75, 81]. Other methods additionally estimate a static object geometry at the same time but require very specific lighting conditions to work [5, 36, 38]. High-quality reconstructions can be achieved via inverse rendering techniques if multi-view data is available [24, 29, 30, 42, 79]. Some methods also consider to leverage multiple frames of dynamic scenes from a single camera to optimize the diffuse texture only [22, 61] or the full SVBRDF [23]. Note that all these approaches need RGBD data and are therefore able to work with ground truth geometry. A more detailed overview of SVBRDF acquisition can be found in the recent survey [31].

Physics simulation Physics simulations are an important part of various different tasks to get realistic dynamics with high precision. There are several frameworks that support general differentiable computations [14, 25, 40] for a variety of systems such as rigid bodies, soft bodies, and fluids. The simulation of cloth can be realized by using a mass-spring model [1, 9, 10, 20] or continuum mechanics [35, 37, 43, 48]. The idea of using a differentiable physics simulation *e.g.* as a regularization is already established and yields successful results in various tasks like the reconstruction of humans, animals, and objects [41, 68, 71, 72, 77] or estimating cloth parameters [20, 27, 65]. The major downside of these simulations is their computational complexity which often dominates the overall runtime.

Neural simulation techniques are able to yield high-quality simulations very quickly [8, 49]. Similar to classical simulations, there are general [46, 70] as well as specialized simulations for fluids [18, 69], soft bodies [46], and cloth in particular [3, 4, 28, 55, 57]. Neural networks with physics-based regularizations are used in reconstruction tasks of loose cloth [62], clothes on human bodies [12, 21, 63], and parameter estimation [15, 26, 52, 73] as well.

Similar methods Our approach of reconstructing the geometry first (ensuring physical plausibility) and estimating appearance parameters afterwards shares notable similarities with Gaussian Garments [53] and PhysAvatar [80], despite their reliance on multi-view data. Better Together [39] incorporates shading information directly into the geometry deformations of faces. The work of Tan *et al.* [66] tackles the SfT problem by restricting the template to align with the contours in image space and smoothing the geometry using a physics-based deformation regularization. ϕ-SfT [27] uses a sophisticated physics simulation to reconstruct the deformation of cloth. In contrast, a neural cloth simulation was introduced to reduce the runtime by a factor of 400 at the expense of slightly worse reconstruction quality and the lack of high-frequency details [62].

3. Method

Our goal is to perform a 3D reconstruction that produces highly accurate geometry, which facilitates subsequent tasks such as appearance estimation and ensuring the faithful capture of even fine visual details. We assume our target video sequence to consist of RGB images and corresponding masks, which may be obtained by systems like Segment Anything [32, 51]. Moreover, we require the initial geometry as a template mesh to start our reconstruction pipeline. Our approach can be split into three different phases for texture mapping, geometry reconstruction, and appearance estimation as shown in Figure 1. Within these phases, we make use of a physical cloth simulation and differentiable rendering techniques to define the pipeline depicted in Figure 2. First, Sections 3.1 and 3.2 describe the simulation and rendering techniques in more detail. Then,

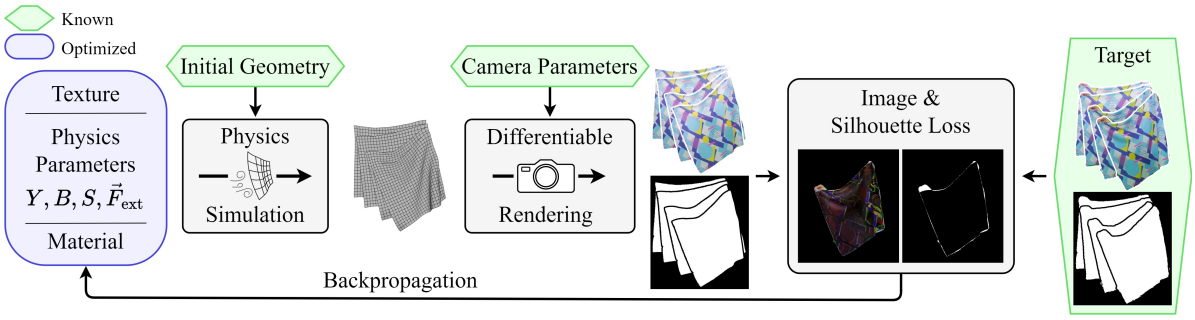


Figure 2. Overview of the pipeline during one epoch. Initially, we perform a physical simulation to obtain the geometry for all frames currently in consideration. Then, renderings and mask images of the fabric are created. Depending on the phase, we either use a texture file for a differentiable rasterizer or an SVBRDF model for a differentiable ray tracer. The images are then compared to the target sequence to define a loss function and perform a gradient-based optimization scheme.

Section 3.3 presents the phases including their requirements and their outcome. More details on the simulation and the optimization can be found in the supplementary material.

3.1. Physical Cloth Model

We employ a physics simulation as our underlying deformation model because it naturally restricts any dynamics to become physically plausible. In particular, the cloth is modeled as a mass-spring system, which is on one hand easy to implement and on the other hand still capable of generating highly realistic motion [9, 10, 20].

3.1.1. Equations of Motion

We represent the cloth as a quad-mesh containing several vertices and edges that also resemble the connections for the mass-spring system. Hence, each vertex is considered to be a point-mass $M = mI$ with position $\vec{x}(t)$, velocity $\vec{v}(t)$, and acceleration $\vec{a}(t)$. The relation between these quantities and the superposition of all forces $\vec{F}$ acting on a vertex is given by Newton's second law of motion:

$$M\vec{a}(t) = M\frac{\partial^2 \vec{x}(t)}{\partial t^2} = \vec{F}(t, \vec{x}, \vec{v}) \tag{1}$$

3.1.2. Numerical Integration Scheme

We employ the backward Euler integration scheme [1] to solve Equation (1) numerically due to its stability with relatively large time steps compared to other integration schemes. By separating the equations of motion into two first-order differential equations and applying the backward Euler method, we get the update rules

$$\vec{v}_{n+1} = \vec{v}_n + \Delta t \vec{a}_{n+1} = \vec{v}_n + \Delta t M^{-1} \vec{F}_{n+1} \tag{2}$$

$$\vec{x}_{n+1} = \vec{x}_n + \Delta t \vec{v}_{n+1} \tag{3}$$

All forces are extrapolated from the current time step by using a first order Taylor approximation to obtain $\vec{F}_{n+1}$.

3.1.3. Forces and Respective Energies

The movement of the mesh vertices is entirely described by their initial positions, velocities, and the forces acting on them. We separate the total forces into several parts, most of which we describe with analytical models. In particular, we model internal forces for the stretching, bending and shearing of the cloth with simple analytic expressions but also allow for arbitrary external forces $\vec{F}_{\text{ext}}$ [27, 62]. The internal forces can be modeled using energy terms $E(\vec{x})$ from which the forces acting on a vertex i at position $\vec{x}^i$ are calculated via $\vec{F} = -\frac{\partial E}{\partial \vec{x}^i}$. We rely on a simple mass-spring model for the energy terms similar to other simulations [9, 10, 20].

Energy Terms We defined the stretching energy as the Hookean energy term

$$E_Y = \frac{1}{2} Y \left(\|\vec{e}^{ij}\| - L_0^{ij} \right)^2 \tag{4}$$

that penalizes deviations of an edge length $\|\vec{e}^{ij}\|$ from its rest length L_0^{ij} scaled by the stretching stiffness Y. The bending and shearing energies model resistance against angular changes between pairs of edges that enclose angles θ^{ijk} or φ^{ijl}. Both energies are defined using the same analytical form

$$E_B = \frac{1}{2} B \left(\theta^{ijk} - \theta_0^{ijk} \right)^2 \text{ and } E_S = \frac{1}{2} S \left(\varphi^{ijl} - \varphi_0^{ijl} \right)^2 \tag{5}$$

with their own bending B or shearing S stiffness parameters, respectively. The stiffness parameters are shared for all stretching, bending, and shearing segments within the cloth to obtain a homogeneous simulation model.

3.2. Differentiable Rendering

3.2.1. Rasterization

During the texture mapping and the deformation optimization, we obtain rendered images via the fast differentiable rasterizer nvdiffrast [33]. We render a RGB image I_n^{rast} and

a mask M_n per frame n of the target video sequence using known camera intrinsic and extrinsic parameters. These images are used to indicate pixel-wise correspondences to guide the necessary motion and therefore do not need to be rendered perfectly. In fact, the texture mapping phase yields a texture with baked-in shadows and lighting such that we do not need to compute these contributions in the rendering. The renderings are used to define image-based loss functions for the color information and the silhouette.

3.2.2. Ray Tracing

In contrast to the deformation optimization, we switch to the high-quality differentiable ray tracing of nvdiffrecmc [24] to optimize the SVBRDF. This enables a decomposition into diffuse, roughness, and metallic textures as well as an estimation of the environment lighting. We intentionally omit the reconstruction of a normal map because of difficulties and ambiguities when optimizing it from only a single viewpoint and relying on approximate geometry. The reconstructed motion yields slightly different viewing angles of the fabric which enables the distinction between diffuse texture and specular highlights. However, this highly depends on the deformation relative to the viewing angle and the lighting conditions in the scene.

We rely on existing work that already dealt with the problem of separating the diffuse, roughness and metallic correctly into respective textures [30]. As we do not impose any illumination prior (*e.g.* illumination by a flashlight), we have to employ a general method that suits our use case. Therefore, we choose to generate the textures by a trainable decoder [29, 30]. It is part of an auto-encoder that is usually pre-trained on a large optical material database but we omitted the pre-training as it did not improve the results in our case. The decoder does not take any input and provides several texture patches that are blended together to achieve the desired texture resolution. We choose to generate 16 patches with 128×128 resolution and an overlap of 10 pixels yielding a resolution of 472×472 for the final output textures. We also estimate the illumination in the form of an environment map texture with a resolution of 512×256 pixels, which is independent of the decoder for the textures.

3.3. Optimization Phases

3.3.1. Texture Mapping

We assume to start with a template that represents a ground truth geometry for the first frame of our video sequence. Hence, the silhouette as well as the texture of a rendered cloth are supposed to align almost perfectly with the target. In the case of a template without (satisfactory) texture we can utilize this information to perform a projective texture mapping. In particular, we optimize the cloth texture T by minimizing the combined loss

$$\mathcal{L}_{\text{tex}} = \mathcal{L}_{0,\text{im}} + \lambda_T \mathcal{R}_T \tag{6}$$

with $\mathcal{L}_{0,\text{im}} = \frac{1}{N_\text{p}} \|I_0^{\text{rast}} - I_0^{\text{GT}}\|_2^2$ being the image loss for the initial frame only and N_p being the number of pixels in the images. The texture regularization $\mathcal{R}_T = \text{FD}(T)$ uses a simple finite difference formula (see supplementary material) to smooth the texture and interpolate unused texels with a small weight of $\lambda_T = 10^{-4}$. We observe that keeping this texture fixed throughout the deformation optimization works best since the cloth is dominantly diffuse even though the optical material and the environment lighting can change the observed colors slightly.

3.3.2. Shape-from-Template Optimization

The shape optimization uses all frames of the video sequence and the texture of the previous step to start from the given initial geometry and successively reconstruct later frames. Similar to previous work [27, 62], we start to optimize only the first three frames and regularly add another frame every 5 epochs. During this phase, we optimize the cloth parameters Y, B, S and external forces $\vec{F}_n^i$. The loss

$$\mathcal{L}_{\text{SfT}} = \mathcal{L}_{\text{im}} + \lambda_{\text{sil}} \mathcal{L}_{\text{sil}} + \lambda_E \mathcal{R}_E + \lambda_F \mathcal{R}_F \tag{7}$$

is composed of four parts, the RGB image loss, the silhouette loss, and our two novel regularization terms. Image and silhouette loss are defined as the mean of single-frame loss terms. We are using the squared L_2 norm on the images to obtain smoother gradients for optimizing the motion. Our key contribution to the 3D reconstruction task is the introduction of two regularization terms $\mathcal{R}_E, \mathcal{R}_F$ that deal with the depth ambiguity when trying to match the target images. Without regularization, the cloth is moved regardless of how meaningful the deformation may be. The gradients of the image and silhouette loss with respect to the vertex positions will dominantly be perpendicular to the viewing direction of the camera which results in crumpling of the cloth to match the target image instead of changing the depth. We will elaborate on this effect in Section 4.1.3.

Energy Regularization We observe that the optimization reduces the stiffness parameters as much as possible to get full control without soft restrictions from the simulation. This leads to unrealistic crumpling as the fabric does not create a sufficiently large restoring force to counteract. Hence, we guide the optimization towards a smoother surface by leveraging the internal cloth energy terms of all N_f currently used frames in a regularization

$$\mathcal{R}_E = \frac{1}{N_\text{f}} \sum_{n=1}^{N_\text{f}} E_{n,\text{SG}(Y)} + E_{n,\text{SG}(B)} + E_{n,\text{SG}(S)} \tag{8}$$

where SG is the stop-gradient operator applied to the stiffness parameters. Note that we have to stop these gradients because we want to regularize the geometry only. Otherwise, the stiffness parameters will be minimized again which is what we aim to prevent in the first place. In the

case of cloth, we naturally choose the deformation energy which could be used in a similar way in the context of other solid materials as well. The regularization weight is set to $\lambda_E = 2$ which results in a loss value that is 1–2 orders of magnitude lower than the image loss $\mathcal{L}_{\text{im}}$ but is still sufficient to prevent crumpling.

Force Regularization In addition to the observed crumpling behavior, we observed minimal displacement along the normalized viewing direction $\vec{d}_n^i$ (from the camera position towards vertex $\vec{x}_n^i$), underscoring the crucial issue of depth ambiguity inherent in monocular data. The reason lies in the information loss in the projection step during rendering. In the supplementary material we show that the gradients of mesh vertices obtained in the backward pass are always perpendicular to the viewing direction. Further contributions from the physical simulation may alter these gradients in the remaining backward pass but this information loss is inevitable and causes the depth ambiguity problem. The external forces are most affected parameters due to their direct and immediate influence on the vertex positions. Thus, we introduce the regularization

$$\mathcal{R}_F = \frac{1}{N_\text{f} N_\text{V}} \sum_{n=1}^{N_\text{f}} \sum_{i=1}^{N_\text{V}} \left(\vec{F}_n^i - \frac{\langle \vec{F}_n^i, \vec{d}_n^i \rangle}{\langle \vec{d}_n^i, \vec{d}_n^i \rangle} \vec{d}_n^i \right) \tag{9}$$

which penalizes external forces $\vec{F}_n^i$ for each vertex i and timestep n along the image direction such that motion along the viewing direction $\vec{d}_n^i$ is preferred for all N_V vertices. The regularization weight is set to $\lambda_F = 2 \cdot 10^{-4}$, producing a magnitude comparable to that of the energy regularization.

Both regularization terms are not meant to constrain the fabric's movements. They are supposed to gently reduce unfavorable motion (*e.g.* crumpling), guide the optimization towards more regular deformations and are thus similar to standard L_1 or L_2 regularizations. Hence, the regularizations only contribute a fraction to the total loss value. The effect of both regularizations will be discussed in detail in the ablation study (Section 4.1.3).

3.3.3. Appearance Estimation

We are optimizing the decoder weights that generate the diffuse, roughness, and metallic textures, together with the environment map. In order to separate the object color from the color in the lighting, we start with a gray environment map and only allow its brightness to change for the first 800 epochs of this phase but not its hue. Afterward, we slowly enable color information by using a linear interpolation between the mean of the color channels and the regular environment map. This interpolation lasts for 400 epochs after which we directly use the environment map with its RGB colors for another 800 epochs. Hence, we optimize the appearance and the illumination for 2000 epochs in total.

In order to disambiguate the brightness of the appearance textures (especially the diffuse color) and the environment map, we slightly regularize the amplitude of the environment map T_E by applying a standard L_1 loss with a weight of $\lambda_\text{env} = 10^{-5}$. Moreover, we apply a smoothness regularization $\mathcal{R}_\text{s_env} = \text{FD}(T_\text{E})$ to the environment map with a weight of $\lambda_\text{s_env} = 5 \cdot 10^{-5}$ similar to the texture smoothness regularization in Section 3.3.1. The full loss function for this phase is given by

$$\mathcal{L}_\text{Mat} = \mathcal{L}_\text{im} + \lambda_\text{env} \mathcal{R}_\text{env} + \lambda_\text{s_env} \mathcal{R}_\text{s_env}. \tag{10}$$

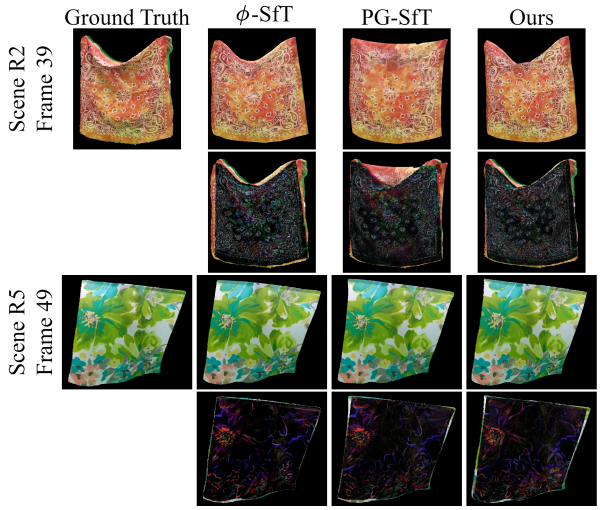


Figure 3. Qualitative comparison between our method and two competing methods for two example frames of different scenes.

4. Evaluation

We evaluate our approach on all real-world scenes of the ϕ-SfT dataset [27] due to its variety in motion and cloth sizes. The dataset provides RGB images, depth images and point clouds for all frames but only the RGB images are used for the reconstruction task. The videos were recorded in a capture stage with a green background and evenly distributed light sources. First, we evaluate the 3D reconstruction qualitatively and quantitatively. We compare our method to the reference implementations of ϕ-SfT [27] and PG-SfT [62]. Furthermore, we demonstrate the importance of our novel regularization terms in an ablation study (Section 4.1.3). Afterward, a qualitative evaluation of the appearance estimation is performed for the real-world scenes in Section 4.2.1 and for synthetic scenes in Section 4.2.2. At the end, we elaborate on the limitations of our method.

4.1. Motion Reconstruction

4.1.1. Qualitative Evaluation

We compare our method to ϕ-SfT [27] and PG-SfT [62] visually by presenting their reconstructions in late frames of scenes R2 and R5 that include significant deformations.

Method	R1	R2	R3	R4	R5	R6	R7	R8	R9	Mean
ϕ-SfT [27]	6.68	11.50	6.63	11.10	17.66	9.30	9.14	3.93	3.29	8.80
PG-SfT [62]	6.07	4.06	10.54	13.32	11.96	15.29	7.18	9.64	9.35	9.71
Ours	**1.01**	**1.55**	**2.71**	**8.29**	**7.12**	**2.18**	**4.84**	**1.23**	**1.04**	**3.33**

Table 1. Quantitative comparison using the L_2 Chamfer distance $[\times 10^{-4}\,\mathrm{m}^2]$.

Method	R1	R2	R3	R4	R5	R6	R7	R8	R9	Mean
ϕ-SfT [27]	1242	1406	1032	1231	836	1107	1064	1101	923	1105
PG-SfT [62]	**2.23**	**2.00**	**1.95**	**1.89**	**2.03**	**2.04**	**1.98**	**1.96**	**1.90**	**2.00**
Ours	34.1	25.2	29.6	22.7	49.0	24.8	29.5	26.2	28.5	29.9

Table 2. Runtime comparison for the geometry reconstruction [min].

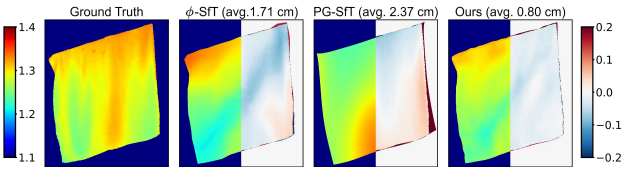


Figure 4. Depth images and differences for scene R6 frame 39. The average depth error during the whole video sequence is denoted in parenthesis.

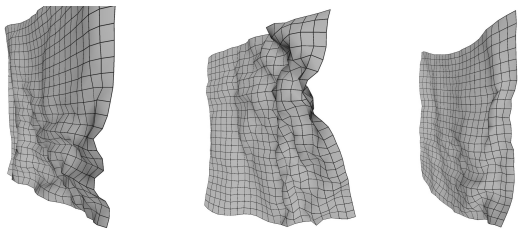


Figure 5. Crumpled fabric in scenes R4, R5, and R7 when using no regularization terms during the optimization.

The reconstructions, ground truth, and difference images for these scenes are depicted in Figure 3 and we present a full comparison of all scenes in a supplementary video. PG-SfT produces errors in the textures and misses to follow the sharp folds, *e.g.* in scenes R2. Both ϕ-SfT and our method yield comparable visual quality with only small mismatches in all scenes. However, note that our optimization finds a balance between a low image loss and a plausible geometry due to the regularization terms.

4.1.2. Quantitative Evaluation

We quantitatively measure the reconstruction quality and runtime of our method and the ones we compare with. The evaluations are performed on an Nvidia A100 GPU and an AMD Epyc 7713 CPU. We refer to the supplemental PDF for a more detailed evaluation including more metrics, comparisons for all scenes, and all formulas.

Figure 4 depicts depth images of all methods for the last frame in scene R6. We clearly see less error in our depth map compared to the baseline methods, which is also confirmed by the average depth error in parenthesis.

Table 1 summarizes the L_2 Chamfer distances for all three methods. Note that the results for ϕ-SfT differ from the values reported in their paper because we do not perform any rigid alignment at the end [27]. Our methods reliably produce geometries with the lowest Chamfer distance often beating the other methods by a factor of at least two. On average, we reduce the mean Chamfer distances of all scenes by $2.64\times$ compared to ϕ-SfT [27] and by $2.92\times$ compared to PG-SfT [62]. The table also demonstrates that the visual quality depicted in Figure 3 does not give evidence about the similarity between the ground truth and the reconstructed geometry due to the missing depth perception in monocular views.

We also evaluate the runtime of all methods. Table 2 lists the runtime to perform the geometry reconstruction (no subsequent appearance estimation for our method) in minutes for all scenes. ϕ-SfT is the slowest algorithm needing an average runtime of $18.4\,\mathrm{h}$ on the aforementioned hardware. Our method performs the motion reconstruction after $29.9\,\mathrm{min}$ on average which is 36 times faster than ϕ-SfT. Our runtime is dominantly affected by the number of frames in the scene resulting in a different number of optimization epochs. The fastest method is PG-SfT with a runtime of only $2\,\mathrm{min}$ per scene with some slight variations due to their fast neural physics simulation [62]. For the sake of completeness, we note that the appearance reconstruction part of our method takes additional $31.9\,\mathrm{min}$ on average to complete. However, this runtime is independent of the 3D reconstruction time and would be the same when using the geometry of the other methods.

Ablation	R1	R2	R3	R4	R5	R6	R7	R8	R9	Mean
w/o silhouette loss $\mathcal{L}_{\text{sil}}$	4.36	2.33	2.24	**6.97**	9.53	**1.60**	5.95	1.58	1.10	3.96
w/o force regularization $\mathcal{R}_F$	7.69	1.39	2.51	17.63	10.45	7.01	**3.41**	2.06	1.47	5.96
w/o energy regularization $\mathcal{R}_E$	2.54	1.47	**1.91**	28.84	128.15	3.68	37.09	3.66	4.96	23.59
w/o both regularizations	5.02	**1.13**	3.11	27.83	120.92	2.07	37.83	1.82	2.89	22.51
Full model	**1.01**	1.55	2.70	8.29	**7.12**	2.18	4.84	**1.23**	**1.04**	**3.33**

Table 3. L_2 Chamfer distances of ablated models compared to the full model $[\times 10^{-4}\,\mathrm{m}^2]$.

4.1.3. Ablation Study

We perform an ablation study by disabling some parts in our reconstruction pipeline. In particular, we evaluate the Chamfer distances of all scenes without using the silhouette loss $\mathcal{L}_{\text{sil}}$, the force regularization $\mathcal{R}_F$, the energy regularization $\mathcal{R}_E$, or both regularizations. The results are summarized in Table 3 and we will briefly elaborate on them.

The absence of the silhouette loss $\mathcal{L}_{\text{sil}}$ has minimal impact on the overall reconstruction quality. In some scenes, the quality improved when omitting the silhouette loss. A simple explanation comes from the RGB images, where the background is already removed. Thus, the mask information is already present in the RGB image loss.

Leaving out the force regularization $\mathcal{R}_F$ sometimes (*e.g.* in scenes R2 and R3) does not change the quality significantly while other scenes (R1, R4, and R6) are heavily affected. Especially the deformations in scenes R1 and R4 are induced by wind blowing approximately in the viewing direction of the camera. These results nicely show the imbalance between gradient information in and orthogonal to the viewing direction caused by depth ambiguity in monocular video. The largest variation in quality is present when disabling the energy regularization $\mathcal{R}_E$. The Chamfer distances of three scenes are massively increased and dominate the mean Chamfer distance of all scenes. Similar results are present when disabling both regularization losses. As already discussed in Section 3.3.2, the energy regularization prevents the optimization to crumple the cloth instead of moving it in a regular fashion. We present examples of crumpled cloth geometry in Figure 5 for the extreme cases in scenes R4, R5, and R7. The only two scenes that do not change significantly or even improve in quality are R2 and R3. Both of them show severe deformations that include a large fold which is slightly smoothed by the regularizations.

Our regularization terms are specifically designed to match the needs of our method and especially the physical simulation model. ϕ-SfT [27] uses a coarser mesh and a more complex but slower physical model, which may be more resistant to crumpling and other effects. Hence, we can not expect our regularization terms to improve its results in the same way. PG-SfT [62] already produces overly smooth reconstructions meaning that the regularizations will not be beneficial at all.

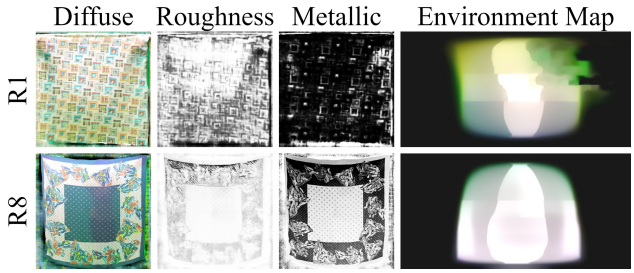


Figure 6. Two examples of estimated appearance of real-world scenes that are reconstructed only from one monocular view.

4.2. Appearance Reconstruction

4.2.1. Real-world Evaluation

The ϕ-SfT dataset does not contain any information about the fabric appearance. We only know that the scenes were captured in a light stage with multiple light sources and green screen background [27]. To the best of our knowledge, there is no method that reconstructs the SVBRDF using comparably low amount of information: The 3D reconstructed geometry of a deformable object captured by a monocular RGB camera without depth information. Therefore, we report and discuss our results for the real scenes of the dataset and evaluate our method on scenes with synthetic appearance parameters. Moreover, the supplemental material shows the lack of details when optimizing the appearance using the geometry reconstructed by PG-SfT.

In Figure 6, the estimated SVBRDFs and environment maps of two example scenes (R1 and R8) are depicted. The results of all scenes are summarized in the supplemental material and the corresponding renderings are given in the supplemental video. The diffuse textures of all scenes look very similar to the images in the target video and resolve sharp details. The roughness is quite high as expected for a diffuse-looking fabric but includes some spatial variations. The metallic textures show a variety of different outcomes which sometimes depend on the pattern of the fabric and again include spatial variations even within similar regions. Inside the environment map, only a large region in the center contains light information because only the light of that approximate hemisphere reflects off of the fabric and reaches the camera while the remaining parts approach zero

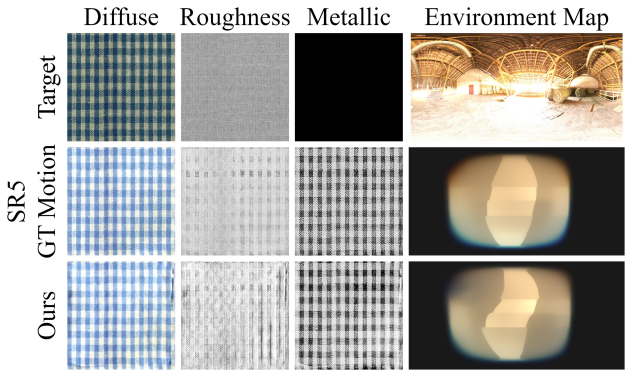


Figure 7. One example of estimating the appearance of the synthesized scene SR5. The first row shows the target textures used to render the images. The second and third rows display the estimated textures when employing the ground truth motion during the reconstruction or our reconstructed motion respectively.

due to the amplitude regularization. Furthermore, the dominantly diffuse fabric only makes it possible to recover very blurry illumination conditions. The environment map and diffuse color in scenes R1 and R8 are separated well and even a green border is visible. Such a dominantly green hue in the environment map is also visible in various other scenes which likely represents the green screen background.

4.2.2. Synthetic Evaluation

To conduct a more detailed evaluation of the appearance estimation, five synthetic scenes were created, each with renderings of different appearance parameters[1]. To ensure the highest degree of similarity with the ϕ-SfT dataset, we reuse the reconstructed motion of our algorithm and save renderings with novel appearance parameters. We call these scene SR1 to SR5 ("synthesized RX") with the motion taken from R1 to R5, respectively.

The results of reconstructing scene SR5 are given in Figure 7 while all remaining synthesized scenes are given in the supplemental PDF. The first row in the figure displays the ground truth appearance textures and the environment map used for creating the rendered target sequence. Since we know the data that generated the synthesized scenes, we are able to perform the appearance estimation using the ground truth motion and report the results in the second row. Hence, we get perfect correspondences for mapping the texture onto the geometry resulting in the appearance estimation possible with our method. We can see that despite some patterns in the roughness and metallic map we get reasonable results for the whole appearance including high-resolution details. Even the brown-colored light is correctly separated into the environment map. The last row depicts the optimized textures when using a reconstructed geometry similar to reconstructing the real scenes. Especially the scenes with non-metallic materials (*e.g.* the one given in Figure 7) yield almost the same results as for the ground truth motion.

[1] All textures were taken from PolyHaven https://polyhaven.com/

4.3. Limitations

4.3.1. 3D reconstruction

During our first phase, we perform a projective texture mapping that provides the main information for the 3D reconstruction. This approach does not reproduce specular highlights from shiny material that might be present when capturing *e.g.* silk fabric. However, estimating the appearance parameters reliably from a monocular view is a challenging task in itself, without knowing the deformation. One possibility could be to jointly reconstruct the motion and the SVBRDF either simultaneously or in an alternating fashion.

Furthermore, despite the high quality achieved in comparison to existing approaches, Figure 3 still shows discrepancies between our reconstructed motion and the target sequence. In this regard, we also showed that our ablated method yield higher quality in specific scenes. Thus, there is the potential to further improve on the quality by finding a universally accurate model for all scenes at the same time.

4.3.2. Appearance Estimation

The SVBRDF optimization is mainly limited by the amount of information available during the optimization. However, there is still room to improve the current results since in some cases the diffuse texture includes inpaintings or the environment map includes some diffuse color. Moreover, the roughness and metallic maps are inpainted as well with regions similar to the diffuse texture pattern.

5. Conclusion

We have presented a Shape-from-Template method to reconstruct the dynamic geometry of deforming fabric employing a physically simulated model. Two novel regularization terms have been introduced, which improve the reconstruction quality by guiding toward a distinct solution to the depth ambiguity problem in monocular videos. As a result, we are able to reduce the chamfer distance between the ground truth and the reconstructed geometry by factors of 2.64 and 2.92 compared to the state-of-the-art methods ϕ-SfT and PG-SfT respectively while requiring moderate runtime. Additionally, we show that it is possible to conduct a material estimation only using the precisely reconstructed geometry and the monocular RGB video data. We have demonstrated that even this limited information is sufficient to obtain detailed appearance textures. We hope that our approach will encourage future work in this domain towards the goal of reducing the overall assumptions for material estimation.

Acknowledgements

This work has been funded by the DFG project KL 1142/11-2 (DFG Research Unit FOR 2535 Anticipating Human Behavior), by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the Lamarr-Institute for Machine Learning and Artificial Intelligence and the InVirtuo 4.0 project, and additionally by the Federal Ministry of Education and Research under grant no. 01IS22094A WEST-AI.

References

[1] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *Annual Conference on Computer Graphics and Interactive Techniques*, page 43–54, 1998. 2, 3, 12

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

[3] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Pbns: Physically based neural simulator for unsupervised garment pose space deformation. *CoRR*, abs/2012.11310, 2020. 2

[4] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Neural cloth simulation. *ACM TOG*, 41, 2022. 2

[5] Mark Boss, Varun Jampani, Kihwan Kim, Hendrik P.A. Lensch, and Jan Kautz. Two-shot spatially-varying brdf and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[6] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR*, pages 690–696 vol.2, 2000. 1

[7] David Casillas-Perez, Daniel Pizarro, David Fuentes-Jimenez, Manuel Mazo, and Adrien Bartoli. The isowarp: the template-based visual geometry of isometric surfaces. *IJCV*, 129(7):2194–2222, 2021. 1

[8] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018. 2

[9] Gabriel Cirio, Jorge Lopez-Moreno, David Miraut, and Miguel A. Otaduy. Yarn-level simulation of woven cloth. *ACM TOG*, 33(6), 2014. 2, 3, 12

[10] Gabriel Cirio, Jorge Lopez-Moreno, and Miguel A Otaduy. Efficient simulation of knitted cloth using persistent contacts. In *Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 55–61, 2015. 2, 3

[11] Devikalyan Das, Christopher Wewer, Raza Yunus, Eddy Ilg, and Jan Eric Lenssen. Neural parametric gaussians for monocular non-rigid object reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10715–10725, 2024. 1

[12] Luca De Luigi, Ren Li, Benoit Guillard, Mathieu Salzmann, and Pascal Fua. Drapenet: Garment generation and self-supervised draping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1451–1460, 2023. 2

[13] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image svbrdf capture with a rendering-aware deep network. *ACM Trans. Graph.*, 37, 2018. 2

[14] Tao Du, Kui Wu, Pingchuan Ma, Sebastien Wah, Andrew Spielberg, Daniela Rus, and Wojciech Matusik. Diffpd: Differentiable projective dynamics. *ACM TOG*, 41(2), 2021. 2

[15] Xudong Feng, Wenchao Huang, Weiwei Xu, and Huamin Wang. Learning-based bending stiffness parameter estimation by a drape tester. *ACM TOG*, 41(6), 2022. 2

[16] 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

[17] Duan Gao, Xiao Li, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. Deep inverse rendering for high-resolution svbrdf estimation from an arbitrary number of images. *ACM Trans. Graph.*, 38(4):134–1, 2019. 2

[18] Nicholas Geneva and Nicholas Zabaras. Modeling the dynamics of pde systems with physics-constrained deep auto-regressive networks. *Journal of Computational Physics*, 403: 109056, 2020. 2

[19] 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*, pages 51–72. Springer International Publishing, 2018. 1

[20] D Gong, Z Zhu, A Bulpitt, and H Wang. Fine-grained differentiable physics: a yarn-level model for fabrics. In *ICLR*, 2022. 2, 3

[21] Artur Grigorev, Michael J. Black, and Otmar Hilliges. Hood: Hierarchical graphs for generalized modelling of clothing dynamics. In *CVPR*, pages 16965–16974, 2023. 2

[22] Kaiwen Guo, Feng Xu, Tao Yu, Xiaoyang Liu, Qionghai Dai, and Yebin Liu. Real-time geometry, albedo, and motion reconstruction using a single rgb-d camera. *ACM Trans. Graph.*, 2017. 2

[23] Hyunho Ha, Seung-Hwan Baek, Giljoo Nam, and Min H. Kim. Progressive acquisition of svbrdf and shape in motion. *Computer Graphics Forum*, 39(6):480–495, 2020. 2

[24] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, light, and material decomposition from images using monte carlo rendering and denoising. In *Advances in Neural Information Processing Systems*, pages 22856–22869. Curran Associates, Inc., 2022. 2, 4, 15

[25] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Fredo Durand. Difftaichi: Differentiable programming for physical simulation. In *ICLR*, 2020. 2

[26] Eunjung Ju and Myung Geol Choi. Estimating cloth simulation parameters from a static drape using neural networks. *IEEE Access*, 8:195113–195121, 2020. 2

[27] Navami Kairanda, Edith Tretschk, Mohamed Elgharib, Christian Theobalt, and Vladislav Golyanik. ϕ-SfT: Shape-from-Template with a Physics-Based Deformation Model. In *CVPR*, pages 3948–3958, 2022. 1, 2, 3, 4, 5, 6, 7, 13, 14, 15, 16, 17, 18

[28] Navami Kairanda, Marc Habermann, Christian Theobalt, and Vladislav Golyanik. Neuralclothsim: Neural deformation fields meet the kirchhoff-love thin shell theory. *arXiv:2308.12970*, 2023. 2

[29] Julian Kaltheuner, Patrick Stotko, and Reinhard Klein. Unified shape and appearance reconstruction with joint camera parameter refinement. *Graphical Models*, 129, 2023. 2, 4

[30] Julian Kaltheuner, Patrick Stotko, and Reinhard Klein. Rosa: Reconstructing object shape and appearance textures by adaptive detail transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025. 2, 4

[31] Behnaz Kavoosighafi, Saghi Hajisharif, Ehsan Miandji, Gabriel Baravdish, Wen Cao, and Jonas Unger. Deep svbrdf acquisition and modelling: A survey. *Computer Graphics Forum*, 43(6):e15199, 2024. 2

[32] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 2

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

[34] Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. *ACM Trans. Graph.*, 36(4), 2017. 2

[35] 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

[36] Zhengqin Li, Zexiang Xu, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Learning to reconstruct shape and spatially-varying reflectance from a single image. *ACM Trans. Graph.*, 37, 2018. 2

[37] Junbang Liang, Ming Lin, and Vladlen Koltun. Differentiable cloth simulation for inverse problems. In *NeurIPS*. Curran Associates, Inc., 2019. 2

[38] Daniel Lichy, Jiaye Wu, Soumyadip Sengupta, and David W. Jacobs. Shape and material capture at home. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6123–6133, 2021. 2

[39] 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. *arXiv preprint arXiv:1708.01654*, 2017. 2

[40] Miles Macklin. Warp: A high-performance python framework for gpu simulation and graphics. https://github.com/nvidia/warp, 2022. NVIDIA GPU Technology Conference (GTC). 2

[41] Mariem Mezghanni, Théo Bodrito, Malika Boulkenafed, and Maks Ovsjanikov. Physical simulation layer for accurate 3d modeling. In *CVPR*, pages 13514–13523, 2022. 2

[42] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting Triangular 3D Models, Materials, and Lighting From Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8280–8290, 2022. 2

[43] Rahul Narain, Armin Samii, and James F. O'Brien. Adaptive anisotropic remeshing for cloth simulation. *ACM TOG*, 31 (6), 2012. 2, 14

[44] Shaifali Parashar, Mathieu Salzmann, and Pascal Fua. Local non-rigid structure-from-motion from diffeomorphic mappings. In *CVPR*, 2020. 1

[45] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. 14

[46] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *ICLR*, 2020. 2

[47] Albert Pumarola, Antonio Agudo, Lorenzo Porzi, Alberto Sanfeliu, Vincent Lepetit, and Francesc Moreno-Noguer. Geometry-aware network for non-rigid shape prediction from a single view. In *CVPR*, 2018. 1

[48] Yi-Ling Qiao, Junbang Liang, Vladlen Koltun, and Ming C Lin. Scalable differentiable physics for learning and control. In *ICML*, pages 7847–7856, 2020. 2

[49] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019. 2

[50] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d, 2020. 14

[51] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2

[52] Carlos Rodriguez-Pardo, Melania Prieto-Martin, Dan Casas, and Elena Garces. How will it drape like? capturing fabric mechanics from depth images. *Computer Graphics Forum*, 42(2):149–160, 2023. 2

[53] Boxiang Rong, Artur Grigorev, Wenbo Wang, Michael J Black, Bernhard Thomaszewski, Christina Tsalicoglou, and Otmar Hilliges. Gaussian garments: Reconstructing simulation-ready clothing with photorealistic appearance from multi-view video. *arXiv preprint arXiv:2409.08189*, 2024. 2

[54] Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Deformable surface tracking ambiguities. In *CVPR*, pages 1–8, 2007. 1

[55] Igor Santesteban, Miguel A. Otaduy, and Dan Casas. Snug: Self-supervised neural dynamic garments. In *CVPR*, pages 8140–8150, 2022. 2

[56] Prem Seetharaman, Gordon Wichern, Bryan Pardo, and Jonathan Le Roux. Autoclip: Adaptive gradient clipping for source separation networks. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2020. 14

[57] Yidi Shao, Chen Change Loy, and Bo Dai. Towards multi-layered 3d garments animation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14361–14370, 2023. 2

[58] 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

[59] Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Didier Stricker. Ismo-gan: Adversarial learning for monocular non-rigid 3d reconstruction. In *CVPRW*, 2019. 1

[60] Vikramjit Sidhu, Edgar Tretschk, Vladislav Golyanik, Antonio Agudo, and Christian Theobalt. Neural dense non-rigid structure from motion with latent space constraints. In *ECCV*, 2020. 1

[61] Miroslava Slavcheva, Maximilian Baust, and Slobodan Ilic. Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[62] David Stotko, Nils Wandel, and Reinhard Klein. Physics-guided Shape-from-Template: Monocular Video Perception through Neural Surrogate Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11895–11904, 2024. 1, 2, 3, 4, 5, 6, 7, 14, 15, 16, 17, 18

[63] Zhaoqi Su, Liangxiao Hu, Siyou Lin, Hongwen Zhang, Shengping Zhang, Justus Thies, and Yebin Liu. Caphy: Capturing physical properties for animatable human avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14150–14160, 2023. 2

[64] John M. Sullivan. *Curves of Finite Total Curvature*, pages 137–161. Birkhäuser Basel, Basel, 2008. 12

[65] Priya Sundaresan, Rika Antonova, and Jeannette Bohgl. Diffcloud: Real-to-sim from point clouds with differentiable simulation and rendering of deformable objects. In *IROS*, pages 10828–10835, 2022. 2

[66] 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*, pages 1–14, 2024. 2

[67] Edith Tretschk, Navami Kairanda, Mallikarjun B R, Rishabh Dabral, Adam Kortylewski, Bernhard Egger, Marc Habermann, Pascal Fua, Christian Theobalt, and Vladislav Golyanik. State of the art in dense monocular non-rigid 3d reconstruction. *Comput. Graph. Forum*, 42(2):485–520, 2023. 2

[68] Shashank Tripathi, Lea Müller, Chun-Hao P. Huang, Omid Taheri, Michael J. Black, and Dimitrios Tzionas. 3d human pose estimation via intuitive physics. In *CVPR*, pages 4713–4725, 2023. 2

[69] Nils Wandel, Michael Weinmann, and Reinhard Klein. Unsupervised deep learning of incompressible fluid dynamics. *CoRR*, abs/2006.08762, 2020. 2

[70] Nils Wandel, Stefan Schulz, and Reinhard Klein. Metamizer: a versatile neural optimizer for fast and accurate physics simulations. *arXiv preprint arXiv:2410.19746*, 2024. 2

[71] Sirui Xu, Zhengyuan Li, Yu-Xiong Wang, and Liang-Yan Gui. Interdiff: Generating 3d human-object interactions with physics-informed diffusion. In *ICCV*, pages 14928–14940, 2023. 2

[72] Gengshan Yang, Shuo Yang, John Z. Zhang, Zachary Manchester, and Deva Ramanan. Ppr: Physically plausible reconstruction from monocular videos. In *ICCV*, pages 3914–3924, 2023. 2

[73] Shan Yang, Junbang Liang, and Ming C. Lin. Learning-based cloth material recovery from video. In *ICCV*, 2017. 2

[74] Wenjie Ye, Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Single image surface appearance modeling with self-augmented cnns and inexact supervision. In *Computer Graphics Forum*, pages 201–211. Wiley Online Library, 2018. 2

[75] Wenjie Ye, Yue Dong, Pieter Peers, and Baining Guo. Deep reflectance scanning: Recovering spatially-varying material appearance from a flash-lit video sequence. In *Computer Graphics Forum*, pages 409–427. Wiley Online Library, 2021. 2

[76] Rui Yu, Chris Russell, Neill D. F. Campbell, and Lourdes Agapito. Direct, dense, and deformable: Template-based non-rigid 3d reconstruction from rgb video. In *ICCV*, 2015. 1

[77] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. In *ICCV*, pages 16010–16021, 2023. 2

[78] Raza Yunus, Jan Eric Lenssen, Michael Niemeyer, Yiyi Liao, Christian Rupprecht, Christian Theobalt, Gerard Pons-Moll, Jia-Bin Huang, Vladislav Golyanik, and Eddy Ilg. Recent trends in 3d reconstruction of general non-rigid scenes. In *Computer Graphics Forum*, page e15062. Wiley Online Library, 2024. 2

[79] Kai Zhang, Fujun Luan, Zhengqi Li, and Noah Snavely. Iron: Inverse rendering by optimizing neural sdfs and materials from photometric images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5565–5574, 2022. 2

[80] Yang Zheng, Qingqing Zhao, Guandao Yang, Wang Yifan, Donglai Xiang, Florian Dubost, Dmitry Lagun, Thabo Beeler, Federico Tombari, Leonidas Guibas, and Gordon Wetzstein. Physavatar: Learning the physics of dressed 3d avatars from visual observations. In *Computer Vision – ECCV 2024*, pages 262–284. Springer Nature Switzerland, 2025. 2

[81] Xilong Zhou and Nima Khademi Kalantari. Adversarial single-image svbrdf estimation with hybrid training. In *Computer Graphics Forum*, pages 315–325. Wiley Online Library, 2021. 2

Supplementary Material

This supplementary material contains further information about the physical simulation and the rendering gradients in Section 6. Further implementation details and additional formulas are summarized in Section 7. The complete evaluation of the appearance estimation as one possible downstream task for the 3D reconstruction is given in Section 8.

6. Method

6.1. Numerical Integration Scheme

For simulation, we discretize the time into constant steps of $\Delta t = 5\,\mathrm{ms}$ resulting in all time-dependent quantities to be discretized accordingly, *e.g.* vertex positions $\vec{x}_n = \vec{x}(n\Delta t)$. The update rules when applying the backward Euler schemes to Newton's law are given by

$$\vec{v}_{n+1} = \vec{v}_n + \Delta t \vec{a}_{n+1} = \vec{v}_n + \Delta t M^{-1} \vec{F}_{n+1} \tag{11}$$

$$\vec{x}_{n+1} = \vec{x}_n + \Delta t \vec{v}_{n+1}. \tag{12}$$

However, these rules require information about the time step that we are trying to compute in the first place. This problem can be solved by introducing the differences $\Delta v = \vec{v}_{n+1} - \vec{v}_n$ and $\Delta x = \vec{x}_{n+1} - \vec{x}_n = \Delta t \vec{v}_{n+1}$ and approximating the force by a first-order Taylor series [1]:

$$\begin{aligned} \vec{F}_{n+1} &= \vec{F}(\vec{x}_n + \Delta\vec{x}, \vec{v}_n + \Delta\vec{v}) \\ &\approx \vec{F}(\vec{x}_n, \vec{v}_n) + \frac{\partial \vec{F}(\vec{x}_n, \vec{v}_n)}{\partial \vec{x}} \Delta\vec{x} + \frac{\partial \vec{F}(\vec{x}_n, \vec{v}_n)}{\partial \vec{v}} \Delta\vec{v} \\ &= \vec{F}_n + \Delta t \frac{\partial \vec{F}_n}{\partial \vec{x}} \vec{v}_{n+1} + \frac{\partial \vec{F}_n}{\partial \vec{v}} \Delta\vec{v} \end{aligned} \tag{13}$$

Inserting this approximation into Equation (11) yields the expression

$$M\vec{v}_{n+1} = M\vec{v}_n + \Delta t \vec{F}_n + (\Delta t)^2 \frac{\partial \vec{F}_n}{\partial \vec{x}} \vec{v}_{n+1} + \Delta t \frac{\partial \vec{F}_n}{\partial \vec{v}} \Delta\vec{v} \tag{14}$$

which can be transformed to get

$$\left(M - (\Delta t)^2 \frac{\partial \vec{F}_n}{\partial \vec{x}} - \Delta t \frac{\partial \vec{F}_n}{\partial \vec{v}} \right) \vec{v}_{n+1} = M\vec{v}_n + \Delta t \vec{F}_n + \Delta t \frac{\partial \vec{F}_n}{\partial \vec{v}} \vec{v}_n \tag{15}$$

for the velocity update. This linear system of equations can be solved using only quantities from the current time step.

6.2. Energy Terms

The forces acting on the cloth can be split into internal forces (stretching, bending, shearing) and external forces (gravity, wind, and other contributions). We model the internal forces explicitly by imposing corresponding deformation energies and obtaining the forces using the derivatives $\vec{F} = -\frac{\partial E}{\partial \vec{x}}$. The following paragraphs will describe each contribution in more detail.

Stretching A stretching energy limits the individual changes in length for edges $\vec{e}^{ij}$ between two connected vertices i and j. We impose a Hookean energy term

$$E_Y = \frac{1}{2} Y \left(\|\vec{e}^{ij}\| - L_0^{ij} \right)^2 \tag{16}$$

that penalizes deviations from the rest length L_0^{ij}. The parameter Y is called stretching stiffness or Young's modulus and scales the amount of stress required to create a certain strain.

Bending We also define two resistances against the deformation of angles within the geometry. The bending energy describes the resistance against a curvature relative to the rest state. In the case of a quad mesh, we use pairs $(\vec{e}^{ij}, \vec{e}^{jk})$ of connected edges that form a straight path on the geometry (see Figure 8). For an enclosed angle $\theta^{ijk} = \sphericalangle(\vec{e}^{ij}, \vec{e}^{jk})$ we define the bending energy

$$E_B = \frac{1}{2} B \left(\theta^{ijk} - \theta_0^{ijk} \right)^2 \tag{17}$$

with the rest angle θ_0^{ijk} and a bending stiffness B [9, 64].

Shearing Furthermore, an energy term is required that penalizes shearing deformations of the geometry, as these do not necessarily result in stretching or bending of the geometry. In real cloth, such deformations will push different yarns together, resulting in a repulsive force from the collisions. An energy term can again be modeled using the angles $\varphi^{ijl} = \sphericalangle(\vec{e}^{ij}, \vec{e}^{jl})$ of all remaining pairs of connected edges that were not used for bending (Figure 8). The functional form of the shearing energy

$$E_S = \frac{1}{2} S \left(\varphi^{ijl} - \varphi_0^{ijl} \right)^2 \tag{18}$$

is the same as for the bending energy but contains a different scaling parameter S, the shearing stiffness. It would also easily be possible to allow for independent stiffness parameters for each edge or pair of edges. However, for the sake of simplicity, we opted to use a homogeneous cloth model.

External Forces All other forces acting on the cloth are summarized in external forces $\vec{F}_{\mathrm{ext}}$. In general, these forces are not conservative and, thus, can not be modeled as the gradient of an energy term. Therefore we treat them as arbitrary vectors $\vec{F}_n^i$ for each vertex $i = 1, \ldots, N_{\mathrm{V}}$ at each

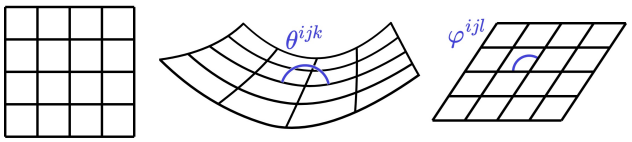


Figure 8. Schematic view of a fabric in its rest state, a bend fabric, and a sheared one.

frame $n = 1, \dots, N_\mathrm{f}$. Two important examples in our scenario are the presence of wind and manipulations by people. Due to their many degrees of freedom, the external forces are able to model the intricate internal cloth forces as well. However, previous work has demonstrated that this approach is much more challenging to optimize and often results in inferior reconstructions [27]. We can observe similar effects, as the forces begin to crumple the fabric together when our regularization terms are omitted, see the ablation study of the main paper (Section 4.1.3). For more control we split the external forces $\vec{F}_n^i = \vec{C} + \vec{D}_n^i$ into a spatially and temporally constant part $\vec{C}$ and a dynamic part $\vec{D}_n^i$.

6.3. Simulation Details

Time Discretization We perform a physical simulation using four constant steps of $\Delta t = 5\,\mathrm{ms}$ between frames to match the $20\,\mathrm{ms}$ video data. For the sake of simplicity, we keep the external forces constant for time steps in between frames, *i.e.* we reuse the same external forces $\vec{F}_n^i$ four times until we switch to the next forces $\vec{F}_{n+1}^i$. Hence, there is no difference between simulation steps and frames other than the simulation error and stability. Individual forces per simulation time step are easy to implement but may require some smoothness regularization to get reasonable interpolation steps due to missing target frames to compare to.

Mass In order to apply Equation (15), we need a mass matrix $M = mI$ associated to each vertex of the cloth. To do so, we define an area density coefficient ρ. We triangulate our template mesh, calculate the mass of each triangle face, and distribute a third of the face mass to each of its vertices. This leads to a relatively homogeneous mass distribution inside the mesh and lower masses at the mesh border. Note that the cloth motion is determined by the acceleration $\vec{a} = M^{-1}\vec{F}$ only and scaling force as well as the mass by the same amount will result in the same motion. Hence, we keep the area density and therefore the masses constant using $\rho = 0.1\,\frac{\mathrm{kg}}{\mathrm{m}^2}$ while optimizing the forces.

Damping We also include an explicit damping term due to the large air resistance present for cloth. We consider the minimalistic damping model

$$\vec{v}_n = \delta\,\vec{v}'_n \tag{19}$$

that scales down the vertex velocities by a factor of δ after the update steps in Equations (12) and (15) are applied. The damping is kept constant at $\delta = 0.9$ as well, since small changes will affect the whole movement and make optimization difficult.

6.4. Orthogonality of Gradients

The gradient of the loss function with respect to some arbitrary (physical) parameter q can be decomposed using the chain-rule. We assume the parameter q to have some influence on a vertex at position $(x\ \ y\ \ z)$ that will be projected on pixel $(p_x\ \ p_y)$ during rendering. An illustration of the projection step is depicted in Figure 9. We now decompose the gradient

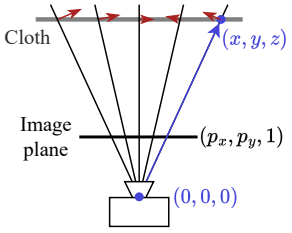


Figure 9. Projection step in camera coordinates.

$$\frac{\partial \mathcal{L}}{\partial q} = \frac{\partial \mathcal{L}}{\partial (p_x\ \ p_y)} \frac{\partial (p_x\ \ p_y)}{\partial (x\ \ y\ \ z)} \frac{\partial (x\ \ y\ \ z)}{\partial q} \tag{20}$$

into three parts denoting contributions for the projection step itself as well as the calculations before and afterwards. Our renderer makes use of the OpenCV camera convention such that the camera intrinsic parameters c_x, c_y, f_x, f_y determine on which pixel a vertex will be projected:

$$\begin{pmatrix} p_x \\ p_y \end{pmatrix} = \begin{pmatrix} \frac{f_x x}{z} + c_x \\ \frac{f_y y}{z} + c_y \end{pmatrix} \tag{21}$$

Its derivative with respect to the vertex coordinates is therefore

$$\frac{\partial (p_x\ \ p_y)}{\partial (x\ \ y\ \ z)} = \begin{pmatrix} \frac{f_x}{z} & 0 & -\frac{f_x x}{z^2} \\ 0 & \frac{f_y}{z} & -\frac{f_y y}{z^2} \end{pmatrix} \tag{22}$$

Similar calculations can be done for other camera conventions as well. An arbitrary image gradient $\frac{\partial \mathcal{L}}{\partial (p_x\ \ p_y)} = (a\ \ b)$ now yields the following 3D gradient for the vertex position:

$$\frac{\partial \mathcal{L}}{\partial (x\ \ y\ \ z)} = (a \quad b) \begin{pmatrix} \frac{f_x}{z} & 0 & -\frac{f_x x}{z^2} \\ 0 & \frac{f_y}{z} & -\frac{f_y y}{z^2} \end{pmatrix} \tag{23}$$

$$= a \begin{pmatrix} \frac{f_x}{z} & 0 & -\frac{f_x x}{z^2} \end{pmatrix} + b \begin{pmatrix} 0 & \frac{f_y}{z} & -\frac{f_y y}{z^2} \end{pmatrix} \tag{24}$$

$$= \begin{pmatrix} a\frac{f_x}{z} & b\frac{f_y}{z} & -a\frac{f_x x}{z^2} - b\frac{f_y y}{z^2} \end{pmatrix} \tag{25}$$

The (unnormalized) viewing direction towards the vertex in camera space is exactly $\vec{d} = (x\ \ y\ \ z)$ and the orthogonality follows directly:

$$\left\langle \frac{\partial \mathcal{L}}{\partial (x\ \ y\ \ z)}, \vec{d} \right\rangle = \begin{pmatrix} a\frac{f_x}{z} & b\frac{f_y}{z} & -a\frac{f_x x}{z^2} - b\frac{f_y y}{z^2} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

$$= 0 \tag{26}$$

Parameter	Init	Min	Max	lr
$\log_{10}(Y)$	$\log_{10}(200)$	1	3	0.02
$\log_{10}(B)$	-3	-4	-2	0.02
$\log_{10}(S)$	-4	-5	-2	0.02
$\vec{C}$	$(0, -1, 0)$	-	-	0.1
$\vec{D}_n^i$	$\vec{0}$	-	-	0.2
Texture T	0.5	0	1	0.05
Diffuse T_{D}	(0.5)	0	1	$2 \cdot 10^{-4}$
Roughness T_{R}	(0.5)	10^{-3}	1	$2 \cdot 10^{-4}$
Metallic T_{M}	(0.5)	0	1	$2 \cdot 10^{-4}$
Environment T_{E}	1.5	0	-	0.01

Table 4. Initial values, limits, and learning rates of all optimized parameters.

As a consequence, the orthogonality of the final gradients $\frac{\partial \mathcal{L}}{\partial q}$ can only be altered by calculations before the projection step happens. In our case, the physical simulation introduces complex interaction between vertices such that the term $\frac{\partial (x \;\; y \;\; z)}{\partial q}$ in Equation (20) breaks the perfect orthogonality. Nonetheless, the loss of information during the projection causes the depth ambiguity problem and heavily influences all gradients with explicit implications for large portions of the gradient computation.

While this property of the rendering gradients is not new, our regularization terms are designed to utilize this knowledge and enhance the gradients in order to improve on 3D reconstruction tasks in monocular setups. While the regularizations are closely related to our specific physical simulation, the presented concepts are applicable in a variety of scenarios.

7. Implementation Details

7.1. Simulation Scheme

As a consequence of the spatial discretization into discrete vertices, the energy terms dependent not only on a single vertex but also on neighboring vertices. In order to get an accurate update step for all vertices at once, we have to include their influence as well. Hence, the derivatives $\frac{\partial \vec{F}_n}{\partial \vec{x}}$ in Equation (15) have to be computed for all vertices participating in the calculation. All derivatives are computed automatically using the automatic differentiation abilities of PyTorch [45] such that only the simple energy functions have to be implemented.

It is possible to infer the physical parameters Y, B, S as well as the external forces acting on the cloth. However, we are only able to infer the net forces that determine the motion of the cloth. Canceling contributions such as gravity and the counteracting holding forces can not be reconstructed individually. Nonetheless, the reconstructed physical parameters and forces can be used to recreate or even change the given dynamics.

7.2. Stable Gradients

Some functions or derivatives introduce numerically instable or undesired expressions in the forward or backward pass of the optimization. We mitigate this issue by limiting all such values to always maintain some distance to all mathematical singularities. However, there may be dramatic variations in the gradients that remain, and we address this issue through two methods. The first is to clip the gradients to a maximum length of 1000. The second is to employ an automatic gradient clipping algorithm [56] that clips gradients based on their length in previous epochs.

7.3. Initial Values, Limits, and Learning Rates

We always start with the same parameters to reconstruct the cloth. Similarly, some parameters are limited to lie between a minimum and maximum value to ensure a stable simulation. All these values are summarized in Table 4. Note, that we actually optimize the logarithm of the three stiffness parameters (when using SI units), *e.g.* $\log_{10}(Y)$, due to their large range of reasonable values of which we allow at least two orders of magnitude. With our hyperparameters the stiffness parameters do not reach their limits during optimization. The different textures and the environment map are initialized with a uniform gray color. In the case of the material textures (diffuse, roughness, and metallic textures), the decoder starts the optimization with gray textures. Note, that the roughness values have to be strictly positive due to singularities at zero during the rendering.

7.4. Comparison of all evaluated Methods

All three methods use ground truth geometry in the first frame but remeshing introduces minor differences between all of them. ϕ-SfT [27] uses irregular meshes with approximately 400 vertices per scene. They employ arcsim [43] to perform a sophisticated cloth simulation based on continuum mechanics and PyTorch3D [50] for rendering, both of which having high computational costs. PG-SfT [62] needs regular rectangular meshes with $32 \times 32 = 1024$ vertices but does not apply any texture mapping resulting in distorted textures compared to ground truth. The physical simulation is performed by a fast physics-informed neural network that was trained in a self-supervised way. Although the neural network computes the simulation results very quickly, its quality is limited to low-frequency motion due to its simple convolutional architectures. The rendering is done using nvdiffrast [33] for fast and differentiable rendering. Our approach combines the benefit of both predecessors: we employ a fast yet accurate simulation based on a mass-spring system. It produces versatile results, is easy to extend and to implement. For the reconstruction task we use regular

meshes with $25 \times 25 = 625$ vertices but arbitrary meshes can be simulated as well. We make use of nvdiffrast as well for fast rendering but also include nvdiffrecmc [24] for high-quality rendering based on ray tracing.

7.5. Additional Formulas

7.5.1. Finite Differences

Within the texture mapping phase, the texture T is smoothed using the regularization $\mathcal{R}_T = \mathrm{FD}(T)$ with

$$\begin{aligned}\mathrm{FD}(T) := \quad & \frac{1}{(w-1)h} \sum_{i=1}^{w-1} \sum_{j=1}^{h} \|T_{i+1,j} - T_{i,j}\|_1 \\ + & \frac{1}{w(h-1)} \sum_{i=1}^{w} \sum_{j=1}^{h-1} \|T_{i,j+1} - T_{i,j}\|_1\end{aligned} \tag{27}$$

being the formula for the mean of all neighboring finite differences in the texture. The same regularization is employed to smooth the environment map T_E during the appearance estimation phase.

7.5.2. Depth Differences

We are able to render depth images D_R of the reconstructed mesh and compare them with ground truth D_GT. We only consider pixels i, j with depth values within the region of interest (ROI) in which both the ground truth as well as the reconstructed geometry yield valid depth values:

$$\delta(D_\mathrm{GT}, D_\mathrm{R}) = \frac{1}{|\mathrm{ROI}_n|} \sum_{i,j \in \mathrm{ROI}} \|D_{ij,\mathrm{R}} - D_{ij,\mathrm{GT}}\|_2 \tag{28}$$

We report the average depth error for all frames in a scene as the final metric.

7.5.3. Chamfer Distance

Similar to previous work [27, 62], we use the symmetric Chamfer distance to quantify the quality of our 3D reconstructed geometry.

$$\begin{aligned}\mathbf{CD}_p(P_\mathrm{GT}, P_\mathrm{R}) = \quad & \frac{1}{|P_\mathrm{R}|} \sum_{\vec{r} \in P_\mathrm{R}} \min_{\vec{t} \in P_\mathrm{GT}} \|\vec{r} - \vec{t}\|_2^p \\ + & \frac{1}{|P_\mathrm{GT}|} \sum_{\vec{t} \in P_\mathrm{GT}} \min_{\vec{r} \in P_\mathrm{R}} \|\vec{r} - \vec{t}\|_2^p\end{aligned} \tag{29}$$

The chamfer distance is computed by comparing a ground truth point cloud P_GT from a depth sensor with a sampled point cloud P_R on the reconstructed fabric mesh and averaging these values over all frames in a scene. The point cloud of the reconstructed mesh P_R will always have the same number of points as the ground truth point cloud P_GT. We use the L_1 Chamfer distance $\mathbf{CD}_1$ as well as the L_2 Chamfer distance $\mathbf{CD}_2$ and compare both of them. The use of linear and quadratic weighting facilitates the identification of outliers.

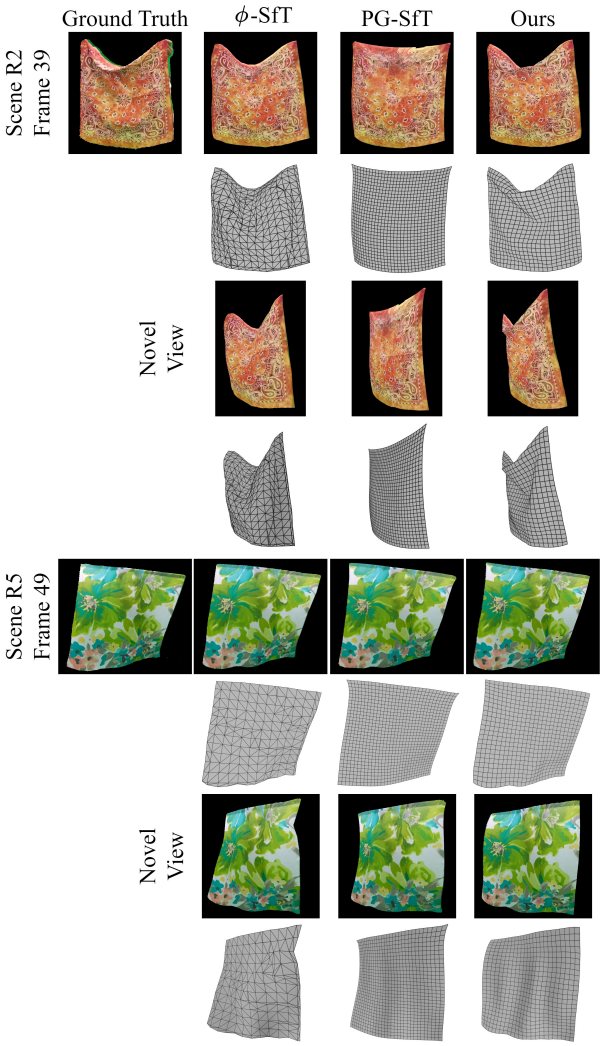


Figure 10. Renderings and mesh views of all methods for two different scenes.

7.5.4. Point to Surface Distance

We also report the symmetric point-to-surface (p2s) distance between the ground truth point cloud P_GT and the reconstructed triangulated surface T_R

$$\begin{aligned}\mathrm{p2s}_p(P_\mathrm{GT}, T_\mathrm{R}) = \quad & \frac{1}{|T_\mathrm{R}|} \sum_{\vec{r} \in T_\mathrm{R}} \min_{\vec{t} \in P_\mathrm{GT}} \mathrm{p2t}(t, r)^p \\ + & \frac{1}{|P_\mathrm{GT}|} \sum_{\vec{t} \in P_\mathrm{GT}} \min_{\vec{r} \in P_\mathrm{R}} \mathrm{p2t}(t, r)^p\end{aligned} \tag{30}$$

using the point-to-triangle distances $\mathrm{p2t}$. Note that in our case the point cloud is the ground truth data and the mesh is the reconstructed surface. Thus, a one-sided metric that averages the minimal distances from each point in the point cloud to the closest point of the mesh surface would not account for overly large reconstructed surfaces. Again, we av-

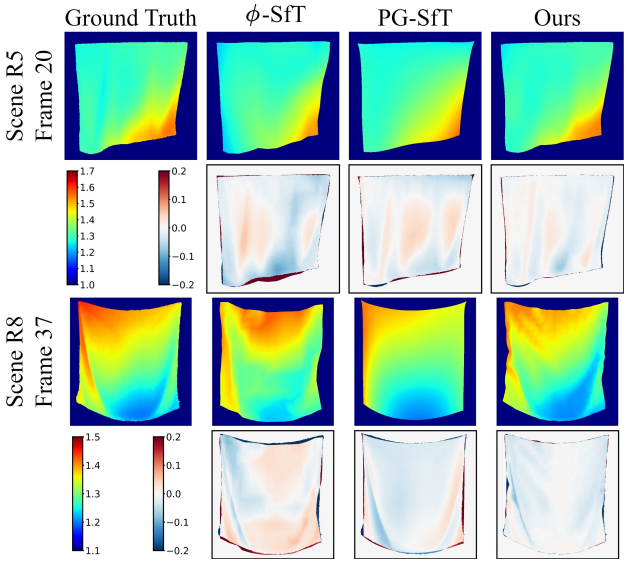


Figure 11. Qualitative comparison between depth images for two example frames of different scenes.

erage the point-to-surface distances for all frames in a scene for the final metric. Similar to the Chamfer distance, we report the L_1 and L_2 p2s distance and compare both of them.

8. Evaluation

8.1. Qualitative Comparison

In addition to the qualitative comparison in the main paper, we present mesh renderings and a novel view for the same scenes in Figure 10 and for all scenes in the supplemental video. The novel view camera is always rotated by $45°$ to the left for all scenes and translated to get a good view on the fabric (with individual translations per scene).

We observe that PG-SfT [62] produces overly smooth geometries that do not capture high-frequency details (*e.g.* in scenes R2, R3, and R4) and distorted textures due to erroneous uv-maps. ϕ-SfT [27] seems to follow the motion well with minor errors. Especially in scene R2 the renderings of ϕ-SfT and our method look similar, however, the novel view and the mesh view reveal significant differences on how the large fold looks like. The fold reconstructed by our method not only looks more realistic, Section 8.2 shows that the 3D error for our method is much lower in the this scenes compared to the other two approaches indicating the correct folding behavior. In scene R5 ϕ-SfT also creates an unrealistic kink close to the top right corner of the mesh which is visible in the novel view in Figure 10 while our method remains smooth as expected for a hanging cloth. Moreover, the supplemental video displays that our method produces visibly the best accordance with the ground truth video almost all the time.

Figure 11 depicts depth values and depth differences with respect to the ground truth depth data for two scenes. The depth values show nicely that PG-SfT does not include any fine wrinkles in the reconstructed mesh despite having the highest mesh resolution. ϕ-SfT and our method both include such details but our results agree better with the ground truth values than ϕ-SfT's results. The supplemental video also points out that our approach yields the best depth values almost all the time.

8.2. Quantitative Comparison

We extend our evaluation by comparing several different metrics for each method. First, we evaluate the depth images quantitatively by averaging the mean depth errors of all images per scene (see Section 7.5.2). We only use the depth values where ground truth and reconstruction overlap (*i.e.* there are valid depth values to compare). This excludes small regions in the images in which only one geometry, either the ground truth or the reconstructed one, is present. We summerize the results in Table 5 and observe that our approach beats ϕ-SfT [27] and PG-SfT [62] in all scenes by up to $4.63\times$ and $2.96\times$ respectively. Both baseline methods have similar performance with mean depth errors of $1.67\,\mathrm{cm}$ and $1.83\,\mathrm{cm}$ respectively. We are able to reduce the average errors by a factor of 1.84 compared to ϕ-SfT and 2.01 compared to PG-SfT.

In addition to the L_2 Chamfer distance CD_2, we also report the L_1 Chamfer distance CD_1 in Table 6 (Section 7.5.3). We use the same point cloud as for the L_2 Chamfer distance, that is created by sampling the same number of points as the ground truth point cloud randomly distributed on the reconstructed mesh. The linear weight for each pair of points reduces the contribution of outliers and focuses on the mean distance between the point clouds instead. We observe that our approach yields the best result in all scenes again. On average the L_1 Chamfer distance improves by a factor of 1.64 compared to ϕ-SfT and by a factor of 1.99 compared to PG-SfT.

Lastly, we compare the L_1 and L_2 p2s distance for all methods (see Section 7.5.4 for the definitions). This metric does not include any sampling but only considers the closest point on a triangle instead of including the whole interior. In 8 out of 9 scenes our method achieves the best results often beating the other approaches by a factor of more than 2.5 for the L_1 distance and a factor of more than 4 for the L_2 distance. Only in scene R5 PG-SfT performs best. This may come from the uniform motion in this scene which does not expose the weaknesses of their method. However, on average our reconstruction outperform ϕ-SfT [27] and PG-SfT [62] by factors of 3.29 and 2.53 for the L_1 p2s distance and by factors of 3.94 and 3.43 for the L_2 p2s distance.

Method	R1	R2	R3	R4	R5	R6	R7	R8	R9	Mean
ϕ-SfT [27]	2.50	1.48	1.29	1.99	2.27	1.71	1.66	1.13	1.01	1.67
PG-SfT [62]	1.60	1.13	2.03	2.08	2.00	2.37	1.54	2.00	1.70	1.83
Ours	**0.54**	**0.71**	**0.76**	**1.64**	**1.31**	**0.81**	**1.28**	**0.63**	**0.51**	**0.91**

Table 5. Average depth difference δ for all methods $[\times 10^{-2}\,\mathrm{m}]$.

Method	R1	R2	R3	R4	R5	R6	R7	R8	R9	Mean
ϕ-SfT [27]	2.21	2.63	2.34	3.56	4.31	3.22	3.04	2.14	1.95	2.82
PG-SfT [62]	2.97	2.14	3.69	4.02	3.64	4.38	3.01	3.79	3.24	3.43
Ours	**1.08**	**1.24**	**1.44**	**2.92**	**2.57**	**1.57**	**2.38**	**1.22**	**1.04**	**1.72**

Table 6. Quantitative comparison using the L_1 Chamfer distance $[\times 10^{-2}\,\mathrm{m}]$.

8.3. Fixed Physical Parameters

We perform a second ablation study that concentrates on the reconstruction quality and stability when the physical material parameters Y, B, and S are not optimized but constant at their initial values. On one hand, these stiffness parameters can not decrease such that the external forces are less likely to become strong enough to crumple the cloth. On the other hand, the optimization loses its ability to adapt to different fabrics automatically. We evaluate the L_2 Chamfer distances in this scenario with and without our regularization terms and collect the results in Table 10. We observe that fixing the physical parameters mainly prevents complete failures without regularizations. Similar to the paper's ablation study, some scenes improve in quality due to the variety in deformations but the full model still reaches the best mean error.

8.4. Real-world Appearance Estimation

We complete the evaluation of the optical material estimation by depicting our results for all real-world scenes in Figure 12a. As a comparison, Figure 12b shows the results when using the imprecisely reconstructed geometry of PG-SfT [62]. We emphasize notable differences between the estimates with red rectangles within the diffuse texture in both figures. Similar characteristics are visible in the roughness and metallic textures. As described in the main paper, the enhanced geometry reconstruction leads to visibly sharper textures in most of the scenes. Especially fine structures like the red striped ovals in scene R3 and the flower stamens in scenes R4 and R5 can not be resolved using the geometry of PG-SfT. The largest difference is visible in the optical material for scene R2. PG-SfT is not able to reconstruct the deformation at all, causing the appearance optimization to fake the appearance by making the fabric highly reflective. The environment maps do not show any significant differences because the dominantly diffuse fabrics only make it possible to recover low-frequency features which are very insensitive to the geometry.

Scene	SR1	SR2	SR3	SR4	SR5
CD_2	1.34	1.56	13.7	0.83	0.99

Table 7. 3D Reconstruction quality of the synthesized scenes $[\times 10^{-4}\,\mathrm{m}^2]$.

8.5. Synthetic Appearance Estimation

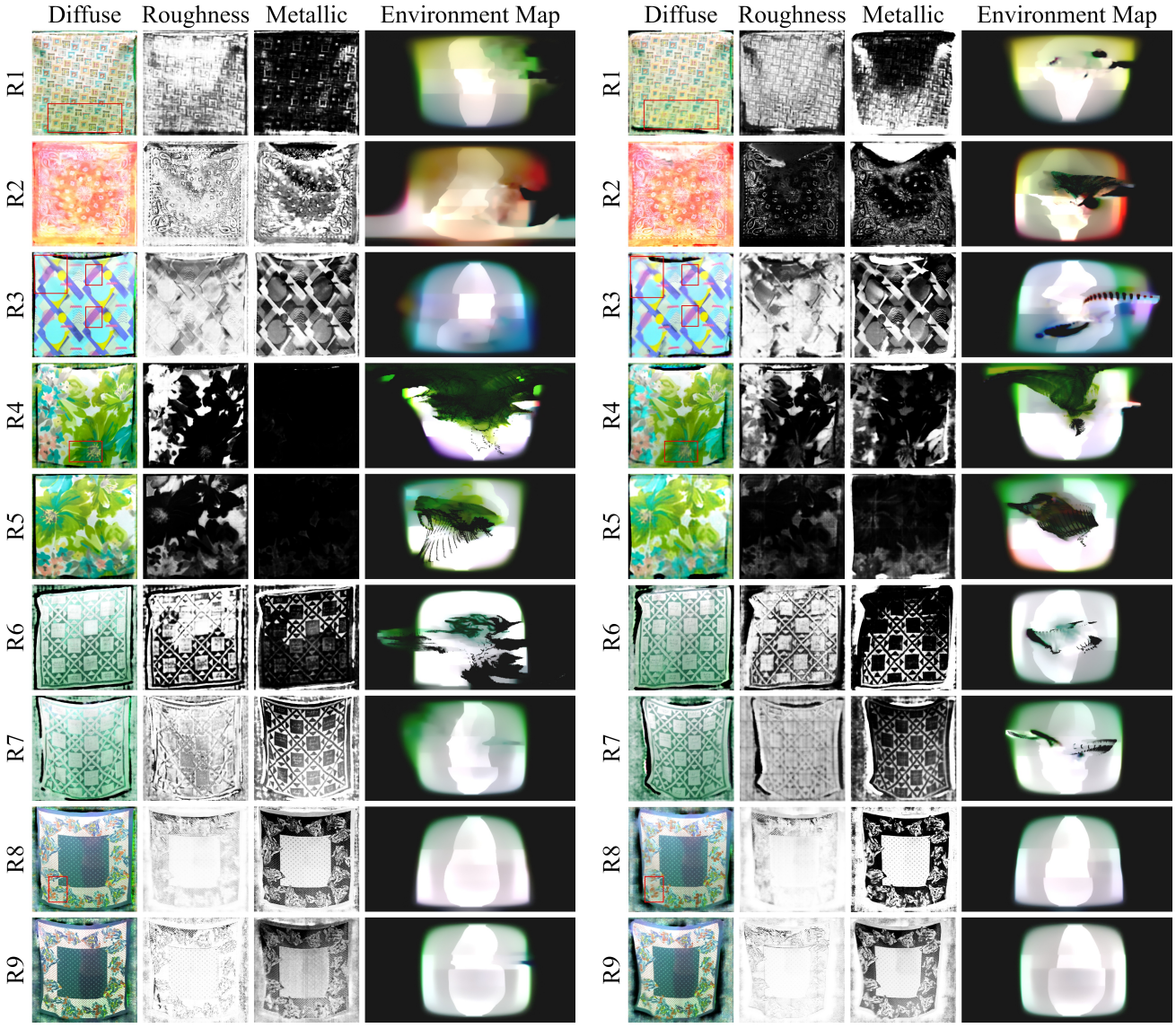
We report the results for the appearance estimation of the synthesized scenes in Figure 13. For all five scenes, we depict the target textures, the estimated textures employing the ground truth motion, and the estimated textures using the reconstructed geometry of our method. The precision of all corresponding geometry reconstructions is given in Table 7. All estimates suffer from the ambiguity that different distributions of color and brightness between diffuse texture and environment map may create the same appearance when observing only limited data, such as a single monocular view. Hence, we mainly compare our results with the ideal case of an estimate based on perfect geometry. The scenes SR2 and SR3 pose a difficult problem to the geometry reconstruction due to their metallic material causing some specular highlights within the video sequence. As a consequence, the geometry does not fit the target well enough for scene SR3 (see Table 7), resulting in blurred textures at the bottom. Nonetheless, despite little color variations, lots of fine details in the optimized optical material are resolved for scene SR2. The three non-metallic materials (SR1, SR4, and SR5) show very similar results to the estimates that use the ground truth geometry. Especially the two fabric materials in scenes R4 and R5 show sharp textures with very little differences.

Method	R1	R2	R3	R4	R5	R6	R7	R8	R9	Mean
ϕ-SfT [27]	7.73	3.96	4.45	12.44	21.87	14.08	11.85	8.25	7.20	10.20
PG-SfT [62]	2.46	5.70	4.94	7.61	**4.52**	20.12	4.05	15.26	5.84	7.83
Ours	**0.70**	**1.42**	**1.58**	**6.80**	6.94	**2.87**	**3.90**	**1.97**	**1.76**	**3.10**

Table 8. L_1 point-to-surface distance $\mathrm{p2s}_1$ between the ground truth point cloud and the reconstructed mesh $[\times 10^{-3}\,\mathrm{m}]$.

Method	R1	R2	R3	R4	R5	R6	R7	R8	R9	Mean
ϕ-SfT [27]	22.65	15.21	9.71	27.38	62.88	30.26	23.78	10.81	8.85	23.50
PG-SfT [62]	3.86	8.24	11.21	17.51	**20.59**	73.12	6.82	30.55	12.38	20.48
Ours	**0.34**	**1.14**	**2.49**	**14.32**	25.26	**2.55**	**4.42**	**1.27**	**1.93**	**5.97**

Table 9. L_2 point-to-surface distance $\mathrm{p2s}_2$ between the ground truth point cloud and the reconstructed mesh $[\times 10^{-5}\,\mathrm{m}^2]$.



(a) Estimation using the geometry of our 3D reconstruction.

(b) Estimation using the geometry of PG-SfT [62].

Figure 12. Estimated appearance parameters of all real scenes. Major differences are highlighted with a red box. Please zoom in for details.

Experiment	R1	R2	R3	R4	R5	R6	R7	R8	R9	Mean
w/o phys. param.	1.64	4.18	2.31	15.28	7.32	5.24	**3.15**	2.12	1.33	4.73
w/o phys. param. & $\mathcal{R}_F$	5.12	5.47	2.29	14.56	**6.83**	5.45	3.29	2.89	1.18	5.23
w/o phys. param. & $\mathcal{R}_E$	4.77	1.56	4.02	9.79	25.65	3.53	4.73	1.72	1.54	6.37
w/o phys. param. & $\mathcal{R}_F$ & $\mathcal{R}_E$	4.52	4.44	**2.30**	15.70	7.42	5.74	3.24	1.58	1.42	5.15
Full model	**1.01**	**1.55**	2.70	**8.29**	7.12	**2.18**	4.84	**1.23**	**1.04**	**3.33**

Table 10. L_2 Chamfer distances without optimization of physical material parameters compared to the full model $[\times 10^{-4}\,\mathrm{m}^2]$.

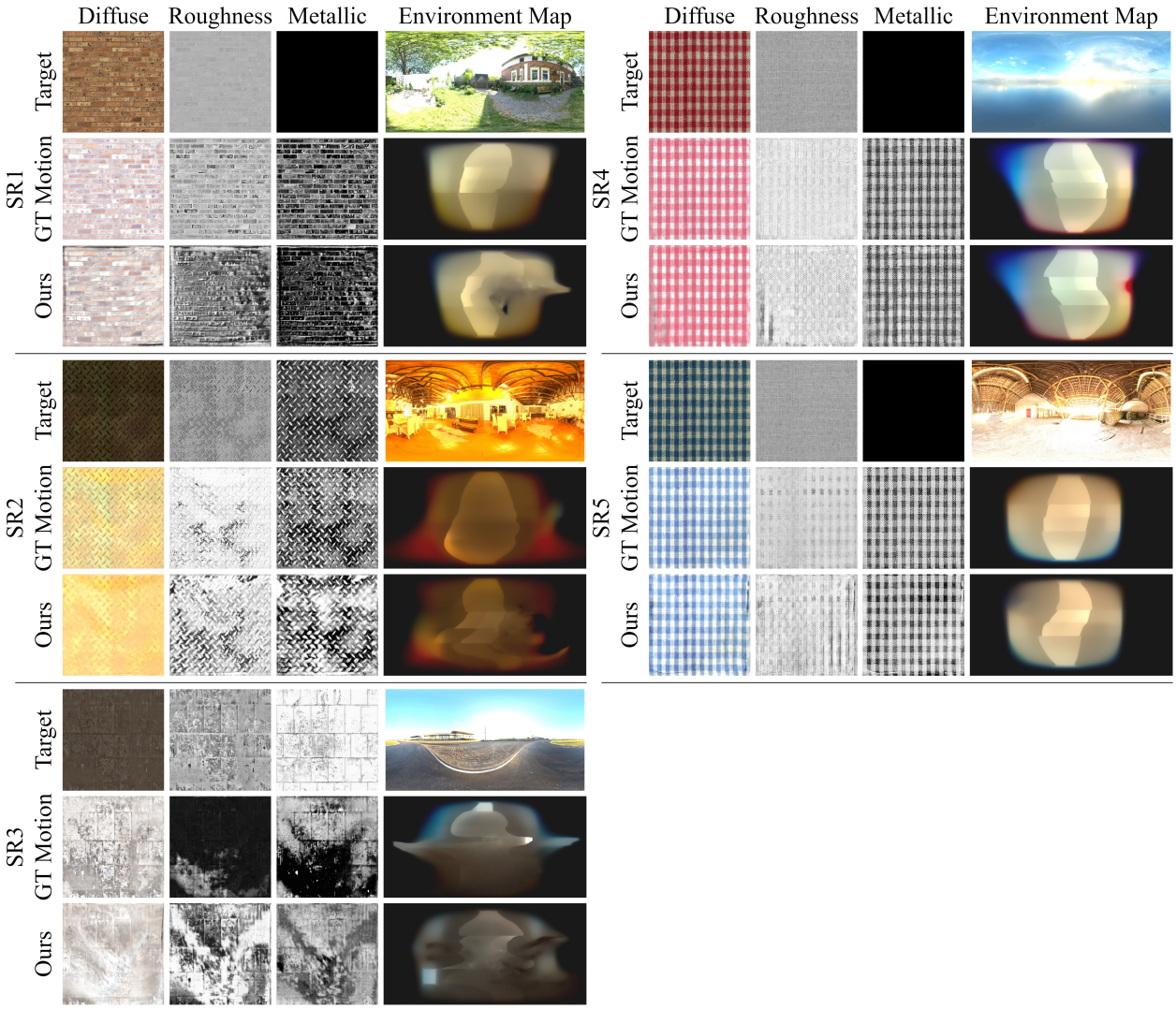


Figure 13. Appearance parameters of all synthesized scenes. The first row depicts the input material textures or the environment map. The second and third row show the estimated appearance maps when using the ground truth geometry or our reconstructed geometry, respectively.