Correspondence-Free Material Reconstruction using Sparse Surface Constraints

Sebastian Weiss
sebastian13.weiss@tum.de

Robert Maier
robert.maier@tum.de

Daniel Cremers
cremers@tum.de

Rüdiger Westermann
westermann@tum.de

Nils Thuerey
nils.thuerey@tum.de

All authors are with Technical University of Munich, Germany

Abstract

We present a method to infer physical material parameters and external boundaries from the scanned motion of a homogeneous deformable object via the solution of an inverse problem. Parameters are estimated from real-world data sources such as sparse observations from a Kinect sensor without correspondences. We introduce a novel Lagrangian-Eulerian optimization formulation, including a cost function that penalizes differences to observations during an optimization run. This formulation matches correspondence-free, sparse observations from a single-view depth image with a finite element simulation of deformable bodies. In a number of tests using synthetic datasets and real-world measurements, we analyse the robustness of our approach and the convergence behavior of the numerical optimization scheme.

1. Introduction

Our goal is to reconstruct material parameters of real-world objects in scenarios where we cannot rely on complex multi-camera setups, but instead only have access to single RGB-D scans from handheld devices. In this setting, we assume that the object's initial shape is known a-priori, and we aim for reconstructing this object with collisions by finding the parameterization of an elasticity simulation that best explains the observations. The central parameters that describe the object's physical behaviour are the Young's modulus, Poisson ratio, as well as damping parameters and boundary conditions, e.g., gravity, initial velocity, and collision planes. While previous works have likewise targeted reconstructing materials from measured deformations and visual observations, they typically rely either on carefully controlled lab settings [2, 28, 50], or on dense observations [9, 44, 19]. In contrast, we focus on real-world interactions, such as falling and colliding objects, that are recorded from a single viewpoint without an explicit feature tracking step.

To match observations with the simulated object without pre-computed correspondences, we propose a novel formulation of implicit constraints that we combine with a differentiable simulation method for soft bodies. We introduce a fully differentiable physics solver that provides gradients for solving a non-linear inverse problem using the adjoint method, to match the depth measurements over time with a physical simulation. Due to the regularization properties of the adjoint optimization, improved robustness wrt to measurement noise over, e.g., the finite difference method, can be achieved. In addition, the adjoint method allows optimizing for multiple parameters with a negligible increase in computations, as only a single backward pass is required to simultaneously compute gradients for all parameters. We further shed light on the inclusion of observed collisions into the optimization via an implicit formulation. To reconstruct collision events even if the actual point of impact is obstructed from view, we leverage the recorded collision response of materials and use it for the inference of a suitable object parameterization.

Specifically, our work targets the image-based reconstruction of material parameters via inverse elasticity problems. In this context, we propose:

- A novel formulation for sparse and correspondence-free surface constraints, e.g. measurements from RGB-D cameras.
- A hybrid Lagrangian-Eulerian formulation that yields gradients for solving an inverse elasticity problems via the adjoint method.
- A detailed robustness and convergence analysis of the proposed optimization scheme using observed real-world deformations and synthetic deformations of objects with ground truth behavior.

We see our work as a step towards improved computational schemes for estimating material parameters and boundary conditions in real-world situations using simple hardware setups. We also believe that this approach can be an interesting alternative to incorporate soft body
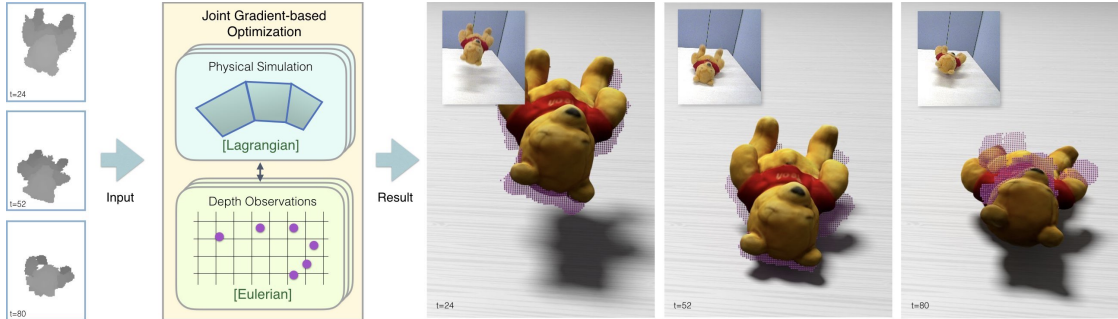
Figure 1: We jointly match a sparse sequence of single-view depth observations (left) with a simulation model and perform an end-to-end gradient-based optimization, to estimate a wide range of material parameters and collision geometries (shown here is a material and damping reconstruction for a plush toy). Ground truth images are shown in the insets.

physics into network-based inference methods, by employing the differentiability of the solver for unsupervised back-propagation[1].

2. Related Work

For reconstructing deforming objects geometrically, a variety of powerful methods were proposed to align templates with measured data [3, 36, 5], or to obtain static reconstructions of scenes [8, 34, 46]. It is also possible to rely on visual information to compute geometry, e.g., via shape from shading [47, 25], or alternatively to estimate surface reflectance models [20, 27, 1]. Deforming objects and characters are inherently difficult to capture, and methods were proposed to capture characters via locally rigid parts [37], or via template based capturing of freely deforming objects [22]. Template-free reconstruction algorithms employing formulations with space-time optimizations were also proposed [29, 51, 10].

Many non-rigid reconstruction algorithms make use of depth data in the form of a video stream to obtain geometric representations [32, 42, 52, 31, 19]. More recent approaches propose level set evolutions for handling topology changes [40, 41], employ articulated skeletons [49], or compute a dense 3D flow around the object [33]. Combinations of such algorithms to capture geometry, motions and surface properties were, e.g., proposed by Guo et al. [14]. A thorough overview of 3D reconstruction algorithms can, e.g., be found in Zollhöfer et al. [53]. While many of these works share our goal to capture deforming objects, they target recovering geometric information, while our method focuses on the reconstruction of physical material parameters.

For modeling an object's behavior, inverse elasticity simulation has been used to compute an alignment between an elastic FE-mesh and the captured point cloud [44]. The alignment procedure is split into the computation of a reference shape that best matches the observed shape, and the estimation of the deformation parameters of the captured shape. In our approach, instead of using a gradient-free downhill simplex method, we formulate the inverse problem as a constrained minimization problem that is solved using the adjoint method [26]. For this, we demonstrate the efficient calculation of the gradient of the cost function with respect to the optimization parameters, which allows us to use only a single camera and include effects such as collisions. Inverse elasticity simulation has also been used in medical imaging to estimate single physical parameters like stiffness and elastic properties[12, 21]. Yet these approaches require dense constraints and rely on full correspondences between observations and the geometry.

We employ an Eulerian FEM model for the elasticity simulation. As an alternative, the Material Point Method recently gained interest for simulating physical models using a Lagrangian object representation [18, 17].

In recent years, also first attempts have been made to replace the elasticity model itself by neural networks [43, 23], an avenue that would yield interesting benefits in conjunction with algorithms for object reconstruction. E.g., several methods employ physical data sets [45, 11], for which our approach could provide additional information in the form of gradients.

3. Overview and Physical Model

At the core of our method, we iteratively optimize the physical parameters that govern the object's deformation behavior, starting from a known reference configuration $\Omega^r \subset \mathbb{R}^3$ and a displacement at time t given by $u : \Omega^r \times t \to \mathbb{R}^3$. The linear *Green strain tensor* $E(u) := \frac{1}{2} \left(\nabla u + (\nabla u)^T \right)$ is used to compute the second order *Piola-Kirchoff stress tensor* as

$$P(u) := 2\mu E(u) + \lambda \operatorname{tr}(E(u))\mathbf{1} \qquad (1)$$

with the *Lamé coefficients* μ and λ derived from the Young's modulus k and the Poisson ratio ρ. The dynamic behavior

---

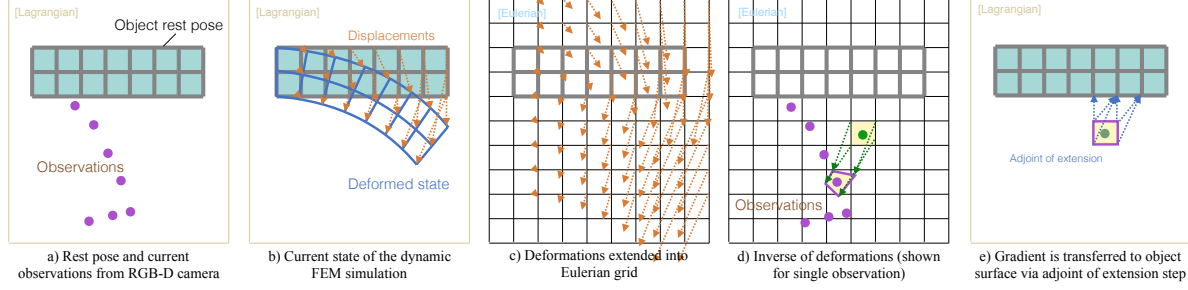| [Lagrangian]	[Lagrangian]	[Eulerian]	[Eulerian]	[Lagrangian]
a) Rest pose and current observations from RGB-D camera	b) Current state of the dynamic FEM simulation	c) Deformations extended into Eulerian grid	d) Inverse of deformations (shown for single observation)	e) Gradient is transferred to object surface via adjoint of extension step

Figure 2: Method overview: Given an object in rest pose and sparse point observations, elastic displacements are computed in a Lagrangian framework and resampled to an Eulerian grid. For each observation, the inverse displacement is computed, and displaced observations are matched to grid cells. A cost function penalizes the distances of displaced observations to the object's surface. Finally, the gradient of this function is transferred to the object surface via the adjoint of the extension step.

of the deformable object is then governed by the system of ordinary differential equations

$$m\ddot{u} - \operatorname{div} P(u) = f_B \quad \text{in } \Omega^r \times \mathbb{R}_0^+ \tag{2a}$$

$$u = u_D \quad \text{on } \Gamma_D^r \times \mathbb{R}_0^+ \tag{2b}$$

$$P(u) \cdot n = f_S \quad \text{on } \Gamma_N^r \times \mathbb{R}_0^+. \tag{2c}$$

with mass m, unit outer normal n, including body forces such as gravity in f_B, as well as Dirichlet and Neumann boundary conditions that prescribe respectively the displacement on the boundary Γ_D^r by u_D and external surface forces on Γ_N^r by f_S.

For the dynamic case, this leads to an initial value problem, using the initial shape Ω^r and material parameters. To discretize problem (2) in space, we use a hexahedral simulation grid [39, 48] with implicitly embedded boundaries [15] and a rotational invariant formulation of the strain tensor using the corotated strain formulation [30, 16]. A Newmark scheme is used for time integration [13].

As input to the reconstruction, we utilize the depth images of an RGB-D camera, giving rise to sparse points in world space. Color information is omitted for simplicity. Furthermore, we assume that a scanned representation of the observed object in the rest pose exists, e.g., from an initial scan.

In a forward step, a Lagrangian elasticity simulation is performed with the current parameter estimates. Consistent with the forward simulation, this step utilizes a Cartesian grid as Eulerian representation. The subsequent matching step, constrained by the physical deformation properties, provides a cost function as similarity measure and enables the calculation of gradients of this function that is to be minimized. These gradients are back-propagated through the simulation using the adjoint method, to obtain updates for the physical parameters via a gradient-based optimization. The reconstruction process is illustrated in Fig. 2.

For the reference configuration of the observed object, a signed distance function (SDF) $\phi^{(0)} : \mathbb{R}^3 \to \mathbb{R}$ is computed. In the object's interior, $\phi^{(0)}$ is negative: $\Omega^r := \{\mathbf{x} \in \mathbb{R}^3 :$

$\phi^{(0)}\mathbf{x}) < 0\}$. For given displacements of material points at time $t \geq 0$ ($t \in \mathbb{R}_0^+$), $u : \Omega^r \times t \to \mathbb{R}^3$, the dynamic behavior of the deformable object is simulated.

The deformed configuration $\phi^{(t)}$ is obtained via an advection step $\phi^{(t)}(x' + u^{(t)}(x')) = \phi^{(0)}(x') \ \forall x' \in \Omega^r$, i.e. $u^{(t)}$ gives the displacement from the point x' in reference configuration to the displaced point x, $x := x' + u^{(t)}(x')$. We define the *inverse displacement field* $u^{(t)}(x)^{-1}$ as the mapping from the displaced point x to the reference point x':

$$u^{(t)}(x)^{-1} := x - x' \iff x' = x - u^{(t)}(x)^{-1}. \tag{3}$$

This allows us to compute the deformed configuration as $\phi^{(t)}(x) = \phi^{(0)}(x - u^{(t)}(x)^{-1})$.

4. Sparse Surface Constraints

To estimate unknown material parameters from observed object deformations with an optimization algorithm, we need to use a cost function J that can reliably and efficiently penalize differences between the observed and simulated object. We propose an SDF-based formulation of J that is able to incorporate sparse constraints without requiring any explicit feature matching. This is different to cost functions considering squared differences of per-vertex displacements and derivatives over time [7, 35], which require ground truth vertex displacements that are not available in our case. Feature tracking methods [38, 44] can circumvent this requirement by explicitly matching observations to vertices in the simulation mesh. However, the matching step does not provide derivatives, and as such cannot be used in combination with gradient-based optimization schemes. Furthermore, explicit matching approaches typically only couple to a simulation via external force estimates, and are thus decoupled from the actual parameter estimation step.

Another cost function variant uses squared differences of the SDF values per domain point [26]. This approach, however, requires a full SDF representation, and while guesses about the complete 3D shape of the observation could be

made, erroneous estimates can easily mislead the optimization procedure. Furthermore, the current SDF needs to be calculated with an advection step from the last to the current timestep, requiring highly non-linear gradient evaluations for inverting the displacement field (see (3)). This often leads to diverging optimizations in practice.

In our setting, we assume that in time step t, $N^{(t)}$ points—with world space positions $\mathbf{x}_{t,i} \in \mathbb{R}^3$—are observed via depth images. Since observed points are located at the object boundary where $\phi = 0$, our *sparse surface constraint* (SSC) cost function aims for minimizing the SDF values at these locations via

$$J_{\text{SSC}}(\phi) := \sum_{t=1}^{T} \sum_{i=0}^{N^{(t)}} \frac{1}{2} \left(\phi^{(t)}(\mathbf{x}_{t,i}) \right)^2. \quad (4)$$

To avoid advecting the full SDF, the SSC formulation encapsulates the underlying sparseness assumption by solving for a point-wise inversion of the body motion. We build upon the assumption that the simulated displacements $u^{(t)}$ do not destroy the signed distance property of ϕ. Then, the deformed SDF $\phi^{(t)}$ can be computed by evaluating the initial SDF at the images of displaced locations, i.e., by moving along the *inverse* displacement field (3): $\phi^{(t)}(\mathbf{x}) \approx \phi^{(0)}\big(\mathbf{x} - u^{(t)}(\mathbf{x})^{-1}\big)$, However, this requires the inverse displacement field, which cannot be computed by simple back-tracing. Since every point $\mathbf{x}$ of the deformed object can be associated with a matching point $\mathbf{x}'$ of the undeformed object, the problem can be reformulated in the following way: Since we know the point $\mathbf{x}$ with $\mathbf{x} = \mathbf{x}' + u^{(t)}(\mathbf{x}')$ (and hence $\mathbf{x}' = \mathbf{x} - u^{(t)}(\mathbf{x})^{-1}$), we can compute the (yet unknown) index (i, j, k) of the hexahedral simulation cell containing $\mathbf{x}'$. With $\mathcal{N} = \{i + [0, 1], j + [0, 1], k + [0, 1]\}$ denoting the eight corners of the cell (i, j, k), and $\mathbf{x}'_{l \in \mathcal{N}}$ their reference locations, let $\mathbf{x}_{l \in \mathcal{N}}$ be the displaced locations of these corners, i.e., $\mathbf{x}_l = \mathbf{x}'_l + \mathbf{u}_l^{(t)}$. Then, the location of point $\mathbf{x}'$ can be computed by tri-linear interpolation of the eight reference corner locations, with the cell-wise interpolation weights α, β, γ.

By further assuming that the interpolation weights don't change during the advection, i.e., $\mathbf{x}'$ is interpolated from $\mathbf{x}'_l$ with the same weights as $\mathbf{x}$ is interpolated from $\mathbf{x}_l$, the same weights for interpolating positions can be used to interpolate the SDF values. This allows us to formulate Alg. 1 for computing $\phi^{(t)}(\mathbf{x})$:

The key step here is solving for the unknown tri-linear interpolation weights. This requires finding a solution within the cell space $[0, 1]^3$ of a non-linear system of equation in three variables. For this, we employ a Newton solve (see Supp. B) that typically converges within a few iterations.

4.1. Extension of Displacements

To ensure that J_{SSC} can be evaluated for all cells that possibly contain an observed point, the displacements that

---

Algorithm 1 Compute $\phi^{(t)}(\mathbf{x})$ based on $\phi^{(0)}$ and $u^{(t)}$

    Input: The observed point $\mathbf{x}$
1: **for each** cell i, j, k **do**
2:     Compute α, β, γ with Newton solve of
3:         $\mathbf{x} = \text{interpolate}(\mathbf{x}_{l \in \mathcal{N}}, \alpha, \beta, \gamma)$
4:     **if** $(\alpha, \beta, \gamma) \in [0, 1]^3$ **then**
5:         **return** $\phi^{(t)}(\mathbf{x}) = \text{interpolate}(\boldsymbol{\phi}_{l \in \mathcal{N}}^{(0)}, \alpha, \beta, \gamma)$
6:     **end if**
7: **end for**

---

are provided by the FE solver only at locations covered by the object, need to be extended into the ambient space around the object. For this purpose, the displacements around the rest pose on the Eulerian grid are extended via a Poisson-based diffusion process. All cells that receive displacements are implicitly matched with observations via the SSC. Once observations and displacements are brought together, the adjoint method ensures that the information travels back to the relevant nodes in the FE mesh. This process is illustrated in Fig. 2 c) to e).

For the SDF extension, we restrict to a narrow band around the surface. The width of the narrow band $\phi_{\max}$ naturally defines an upper bound per point for the value of the cost function. The width specifies the maximum allowed distance of a matched point to the surface. Thus, points that are further away from the surface can be ignored and induce a constant cost of $\frac{1}{2}\phi_{\max}^2$. A detailed quantitative evaluation of how this parameter influences the accuracy of the solutions can be found in Supp. E.3.

To summarize, with one pass over the computational grid, we can compute the inverse mappings (as interpolation weights) for all observed points $\mathbf{x}$, so that J_{SSC} can be evaluated. All steps in the evaluation of J_{SSC}, as well as the extension step, can be efficiently differentiated and incorporated into an inverse elasticity solver for optimizing the material parameters.

5. Inverse Elasticity Solver

By using a forward solver for the elasticity PDE, Eq. (3) in Supp. A, in combination with our proposed cost function J_{SSC}, the adjoint method [26] can be used to optimize for the unknown material parameters. Let $\mathbf{u} \in \mathbb{R}^U$ be the $U \in \mathbb{N}$ states of the system, i.e., the output variables such as the computed displacements $u^{(t)}$ and velocities $\dot{u}^{(t)}$ for each timestep. Let $\mathbf{p} \in \mathbb{R}^P$ be the P control parameters of the system, i.e., the estimated material parameters that are used as input variables in the forward pass. The general optimization problem is then defined as

$$\text{minimize } J(\mathbf{u}, \mathbf{p}) \,, \; J : \mathbb{R}^U \times \mathbb{R}^P \to \mathbb{R} \quad (5a)$$

$$\text{subject to } \mathbf{E}(\mathbf{u}, \mathbf{p}) = \mathbf{0} \,, \; \mathbf{E} : \mathbb{R}^U \times \mathbb{R}^P \to \mathbb{R}^U \quad (5b)$$
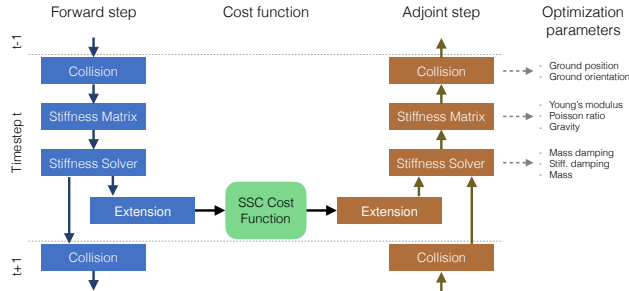
Figure 3: Outline of the forward and adjoint steps.

with a problem-specific function $\mathbf{E}(\mathbf{u}, \mathbf{p})$ that relates the control parameters to the state variables and a cost function $J(\mathbf{u}, \mathbf{p})$. The gradient $\frac{dJ}{dp}$, which is needed in the optimization, is computed by solving for the adjoint state y and using it in the chain rule:

$$\frac{\partial \mathbf{E}}{\partial \mathbf{u}} y = \frac{\partial J}{\partial \mathbf{u}}, \qquad \frac{dJ}{d\mathbf{p}} = -y^T \frac{\partial \mathbf{E}}{\partial \mathbf{p}} + \frac{\partial J}{\partial \mathbf{p}}. \qquad (6)$$

The advantage of the adjoint method is that only a single linear system needs to be solved initially for y. Then, arbitrary control parameters can be added to the final vector-matrix multiplication, making the computational cost of a single gradient evaluation mostly independent of the number of control parameters.

An overview of the different steps that are considered in the adjoint method is shown in Fig. 3. In each time step, the system matrix E in Eq. (5b) captures collision handling (Sect. 5.1), stiffness matrix assembly with corotation (Supp. A.1), stiffness solve via a Conjugate Gradient solver and displacement extension. In the adjoint pass, the order of operations is reversed. Starting from the last frame, the adjoint variables of the displacements and velocities are computed with the derivatives of the operations $\frac{\partial \mathbf{E}}{\partial \mathbf{u}}$, see Supp. C. This gives the adjoint state y in Eq. (6). To improve the performance, we assemble the gradients of the control parameters directly within the respective adjoint operations.

5.1. Collision Embedding

To include collisions in the inverse solver framework, we employ the penalty method [4, 7] (see the accompanying video for the interplay between collisions and the dynamic simulation). Wherever the object penetrates another object, a virtual spring is attached to it that generates a repulsive force that is added as a Neumann boundary in the next timestep.

Let $x = \text{dist}(\mathbf{x})$ be the penetration depth of point $\mathbf{x}$. Then, the force of a spring is described by Hooke's Law: $\mathbf{f} = -kx\mathbf{n}$ with the stiffness factor k and outer normal vector $\mathbf{n}$. In our case, the spring must not exert an attractive force towards the surface when the objects

are not penetrating. Therefore, the force is clamped with $\mathbf{f}_c = -k \min(0, x)\mathbf{n}$.

To obtain a stable simulation, we replace the hard minimum by a soft minimum [6, 24]

$$\mathbf{f}_c = -k \, \text{softmin}(0, x)\mathbf{n}$$

$$\text{with softmin}_\alpha(a, b) := -\ln\left(e^{-a\alpha} + e^{-b\alpha}\right)/\alpha. \qquad (7)$$

This makes the minimum differentiable, which is necessary to consider the collision response in the adjoint method. Furthermore, the collision forces have to be included implicitly in the Newmark time integrator. Since the collision force at the next timestep $\mathbf{f}_c^{(n)}$ is not known, it is approximated using the time derivative of Eq. (7) as

$$\mathbf{f}_c^{(n)} \approx \mathbf{f}_c^{(n-1)} + \Delta t \frac{\partial}{\partial t} \mathbf{f}_c^{(n-1)}. \qquad (8)$$

6. Results and Evaluation

We analyze the accuracy, robustness and performance of our approach. All experiments were performed on a desktop system equipped with an Intel Xeon W-2123 CPU, 64 GB RAM and a Nvidia RTX 2070 GPU. We analyze both synthetic datasets, to compare to ground truth material parameters, as well as several live captures. We first demonstrate robust reconstructions for a bouncing ball made from a homogeneous material, before estimating material parameters of a bending plate, a falling teddy bear and a pillow. Setup parameters as well as concrete timings and model statistics for all experiments are given in Table 4 in the Supplemental Material. For all of our examples, both the forward and backward solve take less than one second per timestep, resulting in a total runtime of roughly 40 minutes for objects comprised of 4000/3000 active nodes/simulation elements. Additional tests are given in Supp. E and F.

6.1. Synthetic Datasets

We consider a bouncing ball with prescribed gravity, Young's modulus and stiffness damping (Fig. 4). The ball's dynamics is forward simulated using the finite-element solver (Fig. 4a), and depth images at resolution 50x50 pixels are rendered and provided as sparse constraints. Multiple optimization runs using different initial conditions (Fig. 4b,d) are performed to reconstruct the material parameters. They are then used in forward simulations to validate against the ground truth simulation (Fig. 4c,e). As shown, the ground truth dynamics is extremely well captured. Gravity optimization converges for all start values, yet this doesn't seem to hold for Young's modulus and stiffness damping. However, Fig. 4f reveals that a hyperbolic relation between these parameters for dynamic motion is in fact exactly reconstructed.

The cost function can also effectively handle complex geometry (see Fig. 5). The soft body Stanford dragon is
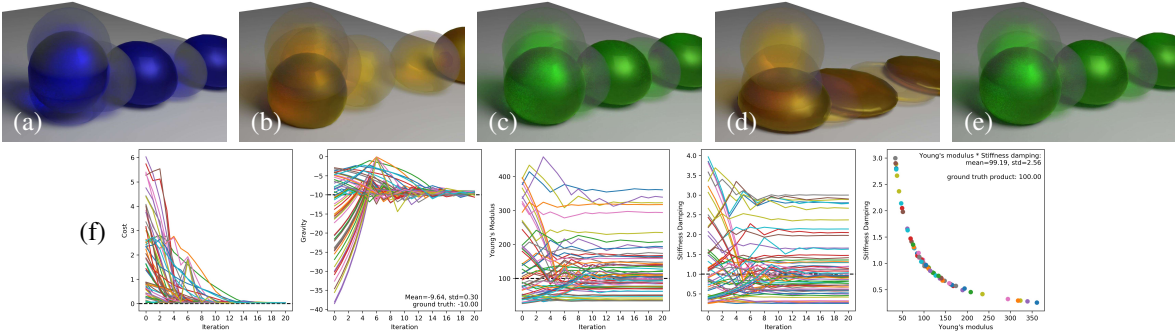
Figure 4: Our algorithm robustly reconstructs material parameters which explain the dynamics of the ground truth (a) very well (c,e), even when starting from very different initial conditions (b,d). Convergence plots show a single optimum for the gravity, and indicate perfect adherence of physical dependency between Young's modulus and stiffness damping. Each line of the same color represents an optimization run.

fixed at its head and pulled down due to gravity. From the observed forward simulation, the Young's modulus is reconstructed and then used in a new simulation which accurately mimics the observed one.
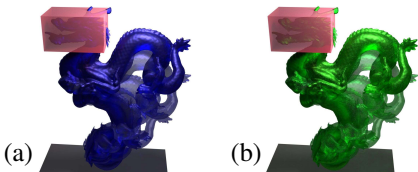

Figure 5: Observed forward simulation (a) and simulation using reconstructed material parameters (b) for the hanging deformable dragon.

To evaluate whether the optimization improves when more cameras or higher camera resolutions are used, different configurations, each with 20 different initial conditions, are tested (Table 1). The adjoint method can robustly reconstruct the parameters, almost always converging with an error of less than 5%. If finite differences are used, the reconstruction quality degrades substantially, with more than 50% of the runs not converging.

6.2. Real-World Scans

We evaluate our approach on a number of live captures from a commodity RGB-D camera, an Asus Xtion ProLive. The reconstruction is performed in a virtual, unit-less system, and results are later scaled to physical units using the object's size and mass (see Supp. D).

Single-Parameter Reconstruction A thin plate made of acoustic foam is fixed at one edge and bends due to gravity. Optimization is for the Young's modulus, the simulation is calibrated with the real-world mass, object size and gravity, and low stiffness damping. The experiment is performed three times with slightly different positions of the plate (Fig. 6), 100 timesteps are recorded each and used in the optimization. The convergence plots indicate that the optimizer always converges quickly to a single optimum,

Increasing # cameras $r = 50^2$			Increasing resolution $n = 1$		
n	error	conv.	r	error	conv.
1	2.888%	95.00%	20^2	2.709%	90.00%
2	8.646%	70.00%	50^2	4.621%	95.00%
4	5.084%	95.00%	100^2	3.046%	100.00%
8	4.104%	95.00%	200^2	2.763%	95.00%
Gradient computation $n = 1, r = 50^2$					
method		error		conv.	
Adjoint Method		2.621%		100.00%	
FD ($\Delta x = 5$)		2.499%		8.33%	
FD ($\Delta x = 100$)		9.238%		41.67%	

Table 1: Statistics for a falling torus (Supp. E.4) for varying numbers of cameras and camera resolutions, and using finite differences versus the adjoint method for gradient computation. Young's modulus is reconstructed. "error": average L_1 error of converged runs ($< 10\%$ error), relative to the ground truth, "conv.": percentage of converged runs.

i.e., 11327Pa, 8960Pa, and 9488Pa even for far-off initial conditions. Due to the approximation of the initial shape and the Dirichlet boundaries, as well as camera noise, an exact match of the reconstructed values cannot be expected.

Multi-Parameter Reconstruction We use a plush teddy, with a high-quality initial pose obtained from a 3D scanner, and record a depth image sequence of the falling teddy (Fig. 7a,b). From 18 random initial configuration, the optimizer estimates gravity, the Young's modulus, the mass- and stiffness damping parameters. Each optimization run performs 50 iterations with 80 timesteps each (see plots in Fig. 7d). The material parameters estimated by the five runs with the lowest reconstruction costs closely match the bouncing behaviour of the teddy. However, the teddy tilts to the side after the first bounce. We attribute this behaviour to inhomogeneities in the material composition, which our solver approximates by assuming homogeneous material. For the Young's modulus, the initial value is several magnitudes higher than the reconstructed values, showing that the optimization is stable over a wide range of values.
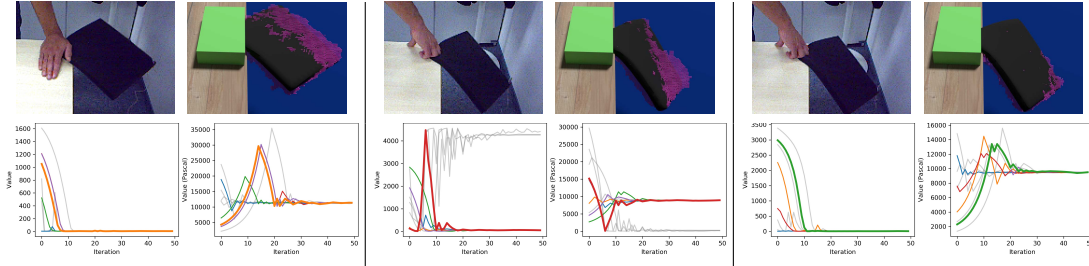
Figure 6: Reconstruction of Young's modulus for recordings of a bending plate made of acoustic foam, starting from different positions. Original and reconstruction are shown side by side.

For the best run, Table 2 shows the reconstructed values with physical units (see Table 1 in the Supplemental Material for all recorded parameter values), and Fig. 7c shows the reconstructed configuration. The gravity is plausible but slightly too high with $11.64\,\mathrm{m/s^2}$, presumably due to effects of the Rayleigh damping. The reconstructed Young's modulus (590Pa) indicates about five times softer material than e.g. polystyrene foam (2500Pa), which seems in good agreement with the observed dynamics.

Testcase	Teddy	Pillow-Flat	Pillow-Ramp
Initial Cost	143.9	249.9	589.3
Recon. Cost	8.461	21.2	59.816
camera framerate	60Hz	60Hz	60 Hz
object size	0.33x0.22x0.18m	0.46x0.46x0.15m	0.46x0.46x0.15m
object mass	0.256kg	0.340kg	0.340kg
grid resolution	29x28x22	27x47x18	30x38x18
Gravity	$11.9\frac{m}{s^2}$	$6.98\frac{m}{s^2}$	$5.68\frac{m}{s^2}$
Young's Modulus	590Pa	151Pa	3430Pa
Mass Damping	0.240	0.078	0.068
Stiffness Damping	0.027	0.015	0.044
Ground Height	-	-	0.127m
Ground Theta	-	-	21.9°
Ground Phi	-	-	4.8°

Table 2: Estimated parameters in real-world units.

Lastly, for a falling pillow bouncing off a skewed ramp, we use 20 perturbed initial configurations (Fig. 8) to let the optimizer simultaneously reconstruct gravity, Young's modulus, mass- and stiffness damping parameters, as well as the collision geometry. (In Supp. F we show this for collisions with a flat surface.) We reconstruct the ground plane height, and its orientation in polar and azimuthal angle. Reconstructed values are given in Table 2. Our method recovers both a plausible orientation of the ramp and the object's material parameters solely from the sequence of depth images. To our knowledge, this is the first simultaneous physical reconstruction of a deformable object and its environment from a single depth video. Since we do not consider friction in the underlying physical model, however, the simulation cannot accurately match the speed of the sliding pillow, and it underestimates gravity. The plots in Fig. 8d indicate the difficulties arising in the current scenario. The parametric ambiguities, e.g., the same contact point can be obtained with high/steep or low/flat ground planes, lead to noticeable differences in the reconstructions. The reconstructed values indicate that the optimizer tries to compensate for the miss-

ing physical phenomena by increasing the material stiffness significantly. Nonetheless, the final result yields a realistic reconstruction of the initial impact and partially matches the observed sliding behavior.

7. Discussion and Outlook

Our formulation provides a method for gradient-based inverse parameter estimation using sparse constraints and physical priors. We see huge potential of this formulation for computer vision tasks to improve unseen or occluded motion, such as the backside of an object, via physical priors. It will also be particularly interesting to investigate the incorporation of soft body physics into deep learning methods via our differentiable formulation. By shifting the workload to a physics-based training process, it is potentially possible to train neural networks in a fully or partially unsupervised manner.

In the real-world experiments, the accuracy of the estimated material parameters is affected by measurement noise as well as the non-physical damping distribution that is assumed in the simulation. Especially the latter introduces forces that reduce the gravity. Similarly, since the soft collision model repulses the object already before the contact point is reached, it can require a stronger gravity force to compensate this effect. Hence, the estimated material parameters are typically less accurate than those obtained from more specialized laboratory experiments [28, 50].

In contrast to previous work [44], we require an initial object pose. It will be interesting to combine our method with the estimation of the rest pose proposed there. Furthermore, we consider the extension region only in proximity to the domain covered by the object. Thus, the cost function cannot be evaluated at points that are observed far outside this region. To handle such cases, we will investigate multi-scale approaches that can efficiently propagate deformations into a wider region around the object. To avoid multiple solutions with different parameter values, it will be interesting to introducing additional priors or domain-specific knowledge about the observed materials.
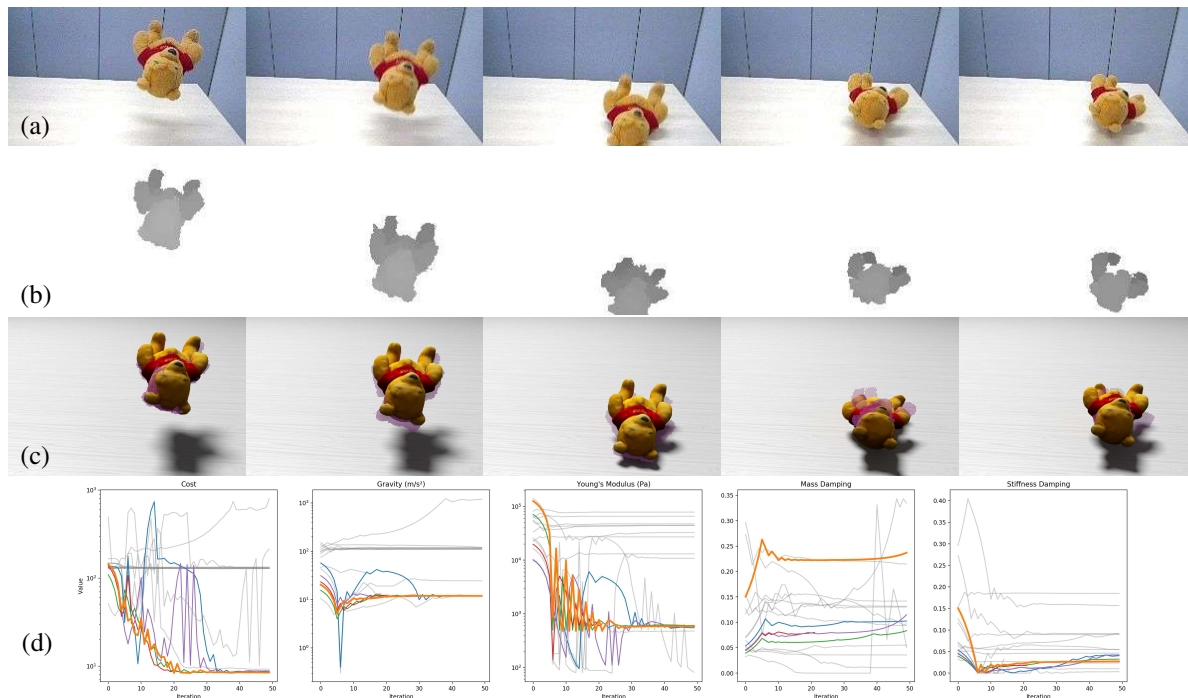
Figure 7: First three rows from top to bottom: Observed colors, observed depths, reconstructed model (purple dots indicate observations). Each row shows a sequence of steps over time. Last row shows the convergence plots for teddy, using 18 randomly selected initial parameter sets. Plots of best 5 runs drawn in color, plot of best run thickened.
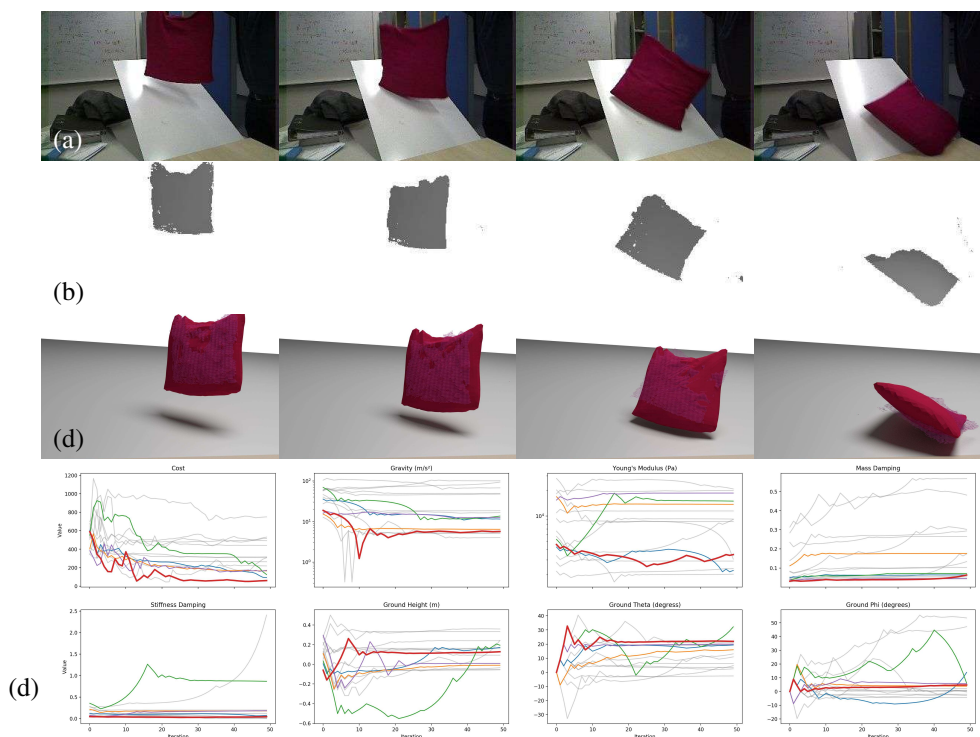

Figure 8: First three rows from top to bottom: Observed colors, observed depths, reconstructed model (purple dots indicate observations). Each row shows a sequence of steps over time. The last row shows the plots of the optimization runs.

References

[1] D. Azinović, T.-M. Li, A. Kaplanyan, and M. Nießner. Inverse path tracing for joint material and lighting estimation. In *CVPR*, 2019.

[2] Bernd Bickel, Moritz Bächer, Miguel A Otaduy, Wojciech Matusik, Hanspeter Pfister, and Markus Gross. Capture and modeling of non-linear heterogeneous soft tissue. In *ACM Transactions on Graphics (TOG)*, volume 28(3), page 89. ACM, 2009.

[3] Volker Blanz, Thomas Vetter, et al. A morphable model for the synthesis of 3d faces. In *Siggraph*, volume 99, pages 187–194, 1999.

[4] Robert Bridson, Ronald Fedkiw, and John Anderson. Robust treatment of collisions, contact and friction for cloth animation. *Proc. Conf. Computer Graphics and Interactive Techniques*, pages 594–603, 2002.

[5] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences*, 103(5):1168–1172, 2006.

[6] John. D. Cook. Soft maximum function. HTTPS://WWW.JOHNDCOOK.COM/BLOG/2010/01/13/SOFT-MAXIMUM/, 2010. Accessed 07/26/2018.

[7] Stelian Coros, Sebastian Martin, Bernhard Thomaszewski, Christian Schumacher, Robert Sumner, and Markus Gross. Deformable objects alive! *ACM Transactions on Graphics*, 31(4):1–9, 2012.

[8] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. 1996.

[9] Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. *ACM Trans. Graph.*, 27(3):98:1–98:10, 2008.

[10] Mingsong Dou, Jonathan Taylor, Henry Fuchs, Andrew Fitzgibbon, and Shahram Izadi. 3d scanning deformable objects with a single rgbd sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 493–501, 2015.

[11] Lawson Fulton, Vismay Modi, David Duvenaud, David IW Levin, and Alec Jacobson. Latent-space dynamics for reduced deformable simulation. In *Computer Graphics Forum*, volume 38, pages 379–391, 2019.

[12] Nachiket H Gokhale, Paul E Barbone, and Assad A Oberai. Solution of the nonlinear elasticity imaging inverse problem: the compressible case. *Inverse Problems*, 24(4):045010, 2008.

[13] Chris Greenough. Newmark's method of direct integration. HTTP://WWW.SOFTENG.RL.AC.UK/ST/PROJECTS/FELIB3/DOCS/HTML/INTRO/INTRO-NODE52.HTML, 2001. Accessed: 05/17/2018.

[14] K. Guo, F. Xu, T. Yu, X. Liu, Q. Dai, and Y. Liu. Real-time geometry, albedo, and motion reconstruction using a single RGB-D camera. *ACM ToG*, 36(3):32, 2017.

[15] Peter Hansbo, Mats G. Larson, and Sara Zahedi. A cut finite element method for a stokes interface problem. *Applied Numerical Mathematics*, 85:90 – 114, 2014.

[16] Michael Hauth and Wolfgang Strasser. Corotational simulation of deformable solids. *Journal of WSCG*, 12(1-3), 2003.

[17] Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. Difftaichi: Differentiable programming for physical simulation. *arXiv:1910.00935*, 2019.

[18] Yuanming Hu, Jiancheng Liu, Andrew Spielberg, Joshua B Tenenbaum, William T Freeman, Jiajun Wu, Daniela Rus, and Wojciech Matusik. Chainqueen: A real-time differentiable physical simulator for soft robotics. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6265–6271, 2019.

[19] Matthias Innmann, Michael Zollhöfer, Matthias Niessner, Christian Theobald, and Marc Stamminger. Volumedeform: Real-time volumetric non-rigid reconstruction. volume 9912, pages 362–379.

[20] K. Kim, J. Gu, S. Tyree, P. Molchanov, M. Nießner, and J. Kautz. A lightweight approach for on-the-fly reflectance estimation. In *ICCV*, 2017.

[21] Martin Kroon and Gerhard A Holzapfel. Estimation of the distributions of anisotropic , elastic properties and wall stresses of saccular cerebral aneurysms by inverse analysis. In *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2008.

[22] Hao Li, Bart Adams, Leonidas J Guibas, and Mark Pauly. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics (TOG)*, 28(5):175, 2009.

[23] Ran Luo, Tianjia Shao, Huamin Wang, Weiwei Xu, Kun Zhou, and Yin Yang. Deepwarp: Dnn-based nonlinear deformation. *arXiv:1803.09109*, 2018.

[24] Martin Mächler. Accurately computing log(1 - exp(.)). HTTPS://CRAN.R-PROJECT.ORG/WEB/PACKAGES/RMPFR/VIGNETTES/LOG1MEXP-NOTE.PDF, 2012. Accessed: 07/15/2019.

[25] R. Maier, K. Kim, D. Cremers, J. Kautz, and M. Nießner. Intrinsic3D: High-quality 3D reconstruction by joint appearance and geometry optimization with spatially-varying lighting. In *ICCV*, 2017.

[26] Antoine McNamara, Adrien Treuille, Zoran Popović, and Jos Stam. Fluid control using the adjoint method. *ACM Trans. Graph.*, 23(3):449–456, Aug. 2004.

[27] A. Meka, M. Maximov, M. Zollhoefer, A. Chatterjee, H.-P. Seidel, C. Richardt, and C. Theobalt. LIME: Live Intrinsic Material Estimation. In *CVPR*, 2018.

[28] Eder Miguel, David Miraut, and Miguel A Otaduy. Modeling and estimation of energy-based hyperelastic objects. In *Computer Graphics Forum*, volume 35(2), pages 385–396. Wiley Online Library, 2016.

[29] Niloy J Mitra, Simon Flöry, Maks Ovsjanikov, Natasha Gelfand, Leonidas J Guibas, and Helmut Pottmann. Dynamic geometry registration. In *Symposium on geometry processing*, pages 173–182, 2007.

[30] Matthias Müller, Julie Dorsey, Leonard McMillan, Robert Jagnow, and Barbara Cutler. Stable real-time deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 49–54. ACM, 2002.

[31] R.A. Newcombe, D. Fox, and S.M. Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *CVPR*, pages 343–352, 2015.

[32] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136. IEEE, 26.10.2011 - 29.10.2011.

[33] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *International Conference on Computer Vision*, 2019.

[34] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3D reconstruction at scale using voxel hashing. *ACM ToG*, 32(6):169, 2013.

[35] Zherong Pan and Dinesh Manocha. Active animations of reduced deformable models with environment interactions. *ACM Transactions on Graphics (TOG)*, 37(3):36, 2018.

[36] Mark Pauly, Niloy J Mitra, Joachim Giesen, Markus H Gross, and Leonidas J Guibas. Example-based 3d scan completion. In *Symposium on Geometry Processing*, number CONF, pages 23–32, 2005.

[37] Yuri Pekelny and Craig Gotsman. Articulated object reconstruction and markerless motion capture from depth video. In *Computer Graphics Forum*, volume 27(2), pages 399–408. Wiley Online Library, 2008.

[38] John Schulman, Alex Lee, Jonathan Ho, and Pieter Abbeel. Tracking deformable objects with point clouds. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1130–1137, 2013.

[39] Eftychios Sifakis and Jernej Barbič. Finite element method simulation of 3d deformable solids. *Synthesis Lectures on Visual Computing: Computer Graphics, Animation, Computational Photography, and Imaging*, 1(1):1–69, 2015.

[40] Miroslava Slavcheva, Maximilian Baust, Daniel Cremers, and Slobodan Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1395, 2017.

[41] M. Slavcheva, M. Baust, and S. Ilic. SobolevFusion: 3D reconstruction of scenes undergoing free non-rigid motion. In *CVPR*, pages 2646–2655, 2018.

[42] Jing Tong, Jin Zhou, Ligang Liu, Zhigeng Pan, and Hao Yan. Scanning 3d full human bodies using kinects. *IEEE transactions on visualization and computer graphics*, 18(4):643–650, 2012.

[43] Bin Wang, Paul G. Kry, Yuanmin Deng, Uri M. Ascher, Hui Huang, and Baoquan Chen. Neural material: Learning elastic constitutive material and damping models from sparse data. *arXiv*, 1808.04931, 2018.

[44] Bin Wang, Longhua Wu, KangKang Yin, Uri Ascher, Libin Liu, and Hui Huang. Deformation capture and modeling of soft objects. *ACM Transactions on Graphics (TOG)*, 34(4):94, 2015.

[45] Z. Wang, S. Rosa, B. Yang, S. Wang, N. Trigoni, and A. Markham. 3d-physnet: Learning the intuitive physics of non-rigid object deformations. In *IJCAI*, 2018.

[46] T. Whelan, S. Leutenegger, R.F. Salas-Moreno, B. Glocker, and A.J. Davison. Elasticfusion: Dense slam without a pose graph. In *RSS*, 2015.

[47] C. Wu, M. Zollhöfer, M. Nießner, M. Stamminger, S. Izadi, and C. Theobalt. Real-time shading-based refinement for consumer depth cameras. *ACM Transactions on Graphics (TOG)*, 2014.

[48] Jun Wu, Rüdiger Westermann, and Christian Dick. A survey of physically based simulation of cuts in deformable bodies. *Computer Graphics Forum*, 34(6):161–187, 2015.

[49] Tao Yu, Kaiwen Guo, Feng Xu, Yuan Dong, Zhaoqi Su, Jianhui Zhao, Jianguo Li, Qionghai Dai, and Yebin Liu. Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 910–919, 2017.

[50] Jonas Zehnder, Espen Knoop, Moritz Bächer, and Bernhard Thomaszewski. Metasilicone: design and fabrication of composite silicone with desired mechanical properties. *ACM Transactions on Graphics (TOG)*, 36(6):240, 2017.

[51] Ming Zeng, Jiaxiang Zheng, Xuan Cheng, and Xinguo Liu. Templateless quasi-rigid shape modeling with implicit loop-closure. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 145–152, 2013.

[52] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, and M. Stamminger. Real-time non-rigid reconstruction using an RGB-D camera. *ACM Transactions on Graphics (TOG)*, 33(4):156, 2014.

[53] Michael Zollhöfer, Patrick Stotko, Andreas Görlitz, Christian Theobalt, Matthias Nießner, Reinhard Klein, and Andreas Kolb. State of the art on 3d reconstruction with rgb-d cameras. In *Computer graphics forum*, volume 37, pages 625–652, 2018.