Constraining Dense Hand Surface Tracking with Elasticity

BREANNAN SMITH, Facebook Reality Labs Research
CHENGLEI WU, Facebook Reality Labs Research
HE WEN, Facebook Reality Labs Research
PATRICK PELUSE, Facebook Reality Labs Research
YASER SHEIKH, Facebook Reality Labs Research
JESSICA K. HODGINS, Facebook AI Research
TAKAAKI SHIRATORI, Facebook Reality Labs Research

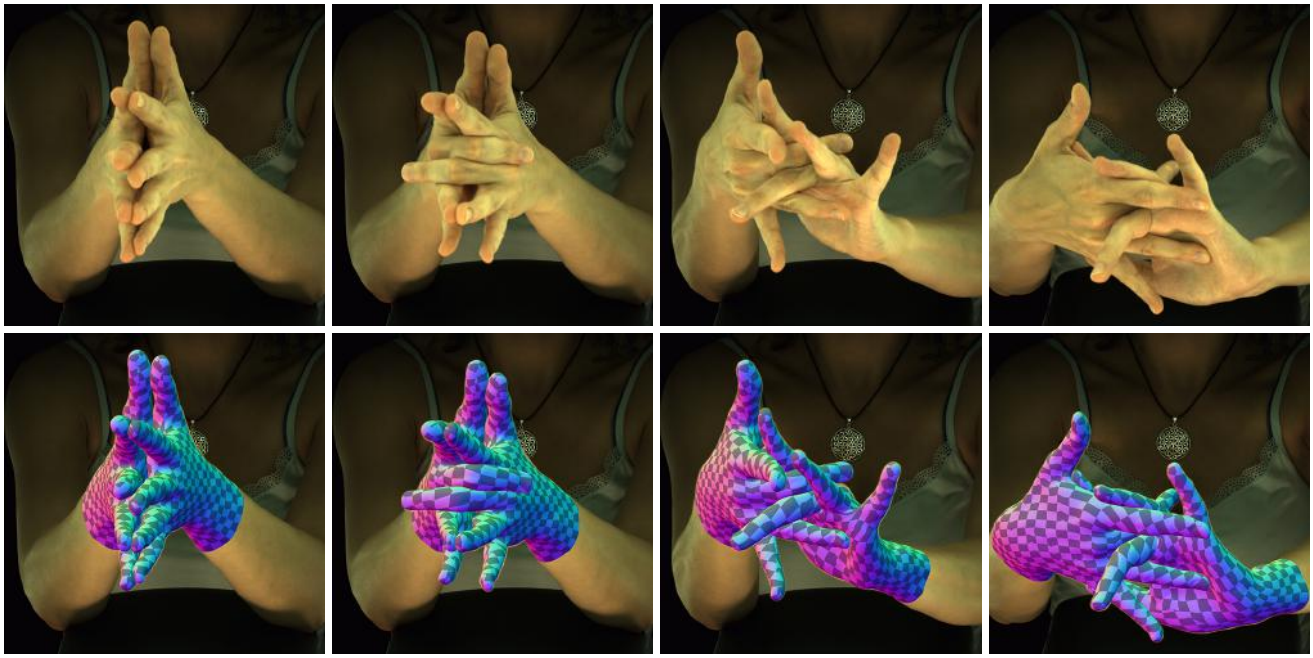

Fig. 1. A subject brings her hands together, bends her middle fingers, pivots her hands around this region of contact, intertwines her remaining fingers, and wiggles her middle fingers. Top row: Input images. Bottom row: Our tracking results. Our approach is able to track through the significant amount of self-contact and and self-occlusion induced by this two-handed performance.

Many of the actions that we take with our hands involve self-contact and occlusion: shaking hands, making a fist, or interlacing our fingers while thinking. This use of of our hands illustrates the importance of tracking hands through self-contact and occlusion for many applications in computer vision and graphics, but existing methods for tracking hands and faces are not designed to treat the extreme amounts of self-contact and self-occlusion exhibited by common hand gestures. By extending recent advances in vision-based tracking and physically based animation, we present the first algorithm capable of tracking high-fidelity hand deformations through highly self-contacting and self-occluding hand gestures, for both single hands and two hands. By constraining a vision-based tracking algorithm with a physically based deformable model, we obtain an algorithm that is robust to the ubiquitous self-interactions and massive self-occlusions exhibited by common hand gestures, allowing us to track two hand interactions and some of the most difficult possible configurations of a human hand.

CCS Concepts: • **Computing methodologies → Motion capture**.

Additional Key Words and Phrases: hand tracking, simulation, elasticity

ACM Reference Format:
Breannan Smith, Chenglei Wu, He Wen, Patrick Peluse, Yaser Sheikh, Jessica K. Hodgins, and Takaaki Shiratori. 2020. Constraining Dense Hand Surface Tracking with Elasticity. *ACM Trans. Graph.* 39, 6, Article 219 (December 2020), 14 pages. https://doi.org/10.1145/3414685.3417768

Authors' addresses: Breannan Smith, Facebook Reality Labs Research, breannan@fb.com; Chenglei Wu, Facebook Reality Labs Research, chenglei@fb.com; He Wen, Facebook Reality Labs Research, hewen@fb.com; Patrick Peluse, Facebook Reality Labs Research, ppeluse@fb.com; Yaser Sheikh, Facebook Reality Labs Research, yasers@fb.com; Jessica K. Hodgins, Facebook AI Research, jkh@fb.com; Takaaki Shiratori, Facebook Reality Labs Research, tshiratori@fb.com.

0730-0301/2020/12-ART219
https://doi.org/10.1145/3414685.3417768

1 INTRODUCTION

Hands are essential in our daily life: we use our hands to manipulate and interact with the world around us, and we also communicate with our hands, using *non-verbal gestures* to transmit, clarify, and emphasize our ideas and thoughts during conversation. Our hands are suited to both these functions due to their high degree of articulation, which leads to dexterity for manipulation and a high bandwidth for communicating information. However, it is precisely due to this high degree of articulation that hands exhibit frequent incidence of occlusion. This occlusion may be caused by contact with other objects, other parts of the body, and often with other parts of the hand itself. Indeed, if you consider where your hands are right now, they are almost certainly in contact with something. It is rare that hands are in a state where they are not in significant contact with another object.

It is for this reason that precisely tracking hand motion in the presence of significant contact and occlusion is a core—perhaps *the* core—challenge in analyzing and synthesizing hand behavior. Technically, this challenge presents as the problem of precisely tracking two deformable surfaces undergoing dynamic contact. Visual observations do not fully constrain the deformation and obtaining precise ground truth in these interactions is nearly impossible.

Addressing this problem will enable the production of more immersive and engaging digital doubles, will help add lifelike realism to characters in movies, and will unlock new forms of interaction modalities in virtual and augmented reality. The ability to track hands with a high degree of fidelity under self-contact and self-occlusion will also help in understanding subtle non-verbal gestures and human-human interactions in the social sciences, as well as enable new applications in medicine and physical rehabilitation. It will provide data for ideation, training, and validation for dexterous humanoid manipulators.

Many efforts have targeted hand tracking via methods that achieve varying degrees of visual and skeletal fidelity. Most existing approaches utilize generic skeleton-based models, *e.g.*, linear blend skinning (LBS) approaches [Taylor et al. 2016; Tan et al. 2016; Taylor et al. 2017] and MANO-based approaches [Romero et al. 2017; Mueller et al. 2019; Hasson et al. 2019; Baek et al. 2019]. While these approaches can replicate a hand's geometry by estimating a skeletal pose, they are not able to reproduce the true detail and fine-grained surface deformations of a hand, particularly when the fingers and palm deform as they interact with one another. While this issue can be alleviated by developing higher-fidelity hand models, the lack of a high-fidelity hand surface tracker impedes progress. At the same time, image-based tracking techniques have delivered high-fidelity dense correspondences for face tracking [Beeler et al. 2011; Wu et al. 2018]. These techniques are difficult to apply directly to hands, however, because unlike the face, hands significantly self-interact and self-occlude in many common poses, making image- and depth-based approaches less effective. These problems grow even more apparent when we consider two interacting hands.

Fortunately, the deformation of the human hand is governed by the laws of physics, suggesting that a physically based approach has the potential to circumvent these challenges and allow us to achieve highly detailed and high fidelity deformations for dense hand surface tracking under self-contact and self-occlusion.

In this paper, we present a new method to track dense hand surfaces to a high degree of fidelity from multi-view image sequences using a physically based model. Specifically, we constrain the solution space of a vision-based tracking algorithm with an elastic volume deformation model and a collision response model, regularizing the entire hand geometry and deforming occluded regions of the hand stably and plausibly. The remaining visible regions are tracked based on visual data from multi-view cameras. To the best of our knowledge, our method is the first to track details such as creasing, bulging, and deformations under extreme self-contact and self-occlusion for one and two hand motion sequences.

To constrain a vision-based tracking method with a deformable physics model, we employ two representations of the hand: a template surface mesh and a volumetric tetrahedral mesh. Given an initialized template mesh, our tracking optimization minimizes photometric and geometric errors using the vertices of the surface mesh, while the entire hand geometry is regularized using an elastic deformation energy and a penetration avoidance term defined on the tetrahedral mesh. To enable communication between these representations, we impose a coupling term during the optimization procedure. We employ an optimization method that alternates [Bezdek and Hathaway 2003] between minimizations of the vision-based terms with the physics terms frozen, and minimizations of the physics terms with the vision-based terms frozen. This alternating procedure allows us to employ state-of-the-art optimization techniques for vision-based tracking and physically based simulation, thus effectively minimizing the total energy. Our method successfully tracks challenging poses and motion sequences for a single hand and for two interacting hands, even with large occluded areas, over multiple individuals with varying appearances.

2 RELATED WORK

We survey the hand tracking literature and summarize the current state of the art in tracking with deformable elastic simulations, highlighting the techniques most closely related to ours.

Hand Surface Tracking. Human hands, due to their structure and range of motion, are surprisingly difficult to densely track, demonstrating self-similarity in both geometry and appearance and experiencing self-occlusions by the fingers and the palm. A majority of hand tracking works focus on estimating skeletal poses with or without geometry. We refer readers interested in pose tracking alone (*i.e.*, estimating 3D joint positions or angles from visual data) to [Yuan et al. 2018]. We will focus primarily on hand surface tracking here. We further delineate our work from the existing dense hand tracking literature by noting that we focus on tracking hands to the highest possible fidelity with a *multi-view* capture system. For recent, state-of-the-art results on dense hand tracking with a *monocular* system, we refer readers to [Ge et al. 2019].

With a surface deformation model, a low-dimensional parameterized space is estimated so that the hand geometry can be obtained from the deformation model. Many existing approaches incorporate linear blend skinning, which deforms a mesh based on a linear combination of rigid transformations of associated bones, on top of

various skeletal representations, including geometric primitives [Iason Oikonomidis and Argyros 2011], a sphere-mesh [Tkach et al. 2016, 2017], and a set of 3D Gaussians [Sridhar et al. 2013]. This representation can produce reasonable deformations for articulated objects if skinning weights are carefully designed, making it suitable for hand surface tracking [Taylor et al. 2016; Tan et al. 2016; Taylor et al. 2017]. More recently, Romero *et al.* augmented an LBS-based hand model with statistical identity- and pose-dependent geometric correctives, leading to a system they name MANO [Romero et al. 2017]. The MANO model has been used to track hands in scenarios including two-hand interactions [Mueller et al. 2019], hand-object interactions [Hasson et al. 2019] and in-the-wild single color images [Baek et al. 2019]. These existing approaches are fully constrained by the underlying models, however, and often fail to replicate subtle details of hand geometry like creases and bulging.

Different than these works, our method takes the first step towards densely estimating hand surface correspondences in scenarios with heavy self-contact and self-occlusion. Dense correspondence techniques have made great recent progress in tracking faces [Beeler et al. 2011; Fyffe et al. 2015; Wu et al. 2018; Fyffe et al. 2017], often targeted at driving visual effects for films [Beeler et al. 2014]. These methods are specifically designed for capturing facial performances, however, which is a simpler setting than hands as the face encounters minimal self-occlusions and self-collisions. Our method takes the first strides towards using physically based laws to estimate dense correspondences for highly occluded and heavily self-colliding objects like hands, where existing approaches would fail.

Applying physics to hand surface tracking to preserve physical correctness, conserve volume, and avoid penetration is a challenging task. Some hand tracking approaches tackle the problem of penetration avoidance by detecting collisions using a sparse set of proxy spheres [Oikonomidis et al. 2011], approximate proxy geometries [Tzionas et al. 2016], sphere-meshes [Tkach et al. 2017] and 3D Gaussians [Mueller et al. 2019]. Unlike existing approaches we do not approximate penetration testing, instead using the full degrees of freedom of a tetrahedralized hand model to avoid penetration. We also consider physical properties of human tissue, including incompressibility, which existing tracking approaches neglect. These considerations allow us to reproduce physically plausible deformations under contact, which existing methods can not achieve.

Tracking with Deformable Elastic Simulations. Deformable elastic simulations have been employed in both reconstruction and tracking applications. Notable early works simulate 'symmetry-seeking' 3D elastic models embedded in force fields defined by image intensity gradients to reconstruct [Terzopoulos et al. 1987] and track [Terzopoulos et al. 1988] cylindrically-shaped objects. A similar technique simulates 2D quasistatic elastic splines to track image contours using forces derived from user constraints and image intensity gradients [Kass et al. 1988]. In the context of reconstruction, Szeliski *et al.* [1991] derive conditions under which minimizing a quasistatic elastic model subject to constraints from a sensor is equivalent to imposing the elastic model as a prior on a probabilistic model. This work further derives a sequential tracking algorithm by designing a Kalman filter where a deformable elastic simulation is used as the system model. Schulman *et al.* [2013] derive a similar probabilistic approach to track rods and flat sheets of material. Subsequent work expands on the Kalman filter approach [Metaxas and Terzopoulos 1993], tracking a torso and arms using a low resolution deformable elastic surface. In the context of full body tracking, de Aguiar *et al.* [2008] use a volumetric tetrahedral mesh with an as-rigid-as-possible (ARAP) deformation model [Sorkine and Alexa 2007] and constraints derived from SIFT features and silhouettes to obtain an initialization for a surface-based tracking algorithm. This work does not handle self-collisions and recent work [Smith et al. 2018] has shown that ARAP and its co-rotational extension produce unacceptable artifacts when used to model human flesh, making this technique less than ideal for modeling a hand. A similar method instead performs an initial pass of surface tracking with a data term and surface regularizer followed by a solve with a volumetric tetrahedral mesh and a linear elastic model to improve tracking quality in unobserved regions [Wuhrer et al. 2015]. Linear elasticity is not suited for modeling large deformations [Müller et al. 2002], however, and this work is thus forced to reinitialize the mesh's rest state at each frame, leading to artificial plasticity and limiting the method to small deformations. Barrielle *et al.* [2016] derive forces from and drive simulations with linear combinations of blendshapes and optimize for sets of blendshape weights that make the resulting simulation most closely match face tracking results computed from sparse surface correspondences. While blendshapes are well suited for animating faces [Lewis et al. 2014], even with rigid transforms factored off hands experience massive nonlinear local changes in shape, making direct applications of linear blendshapes difficult.

We are heavily inspired by and draw from the rich literature on tracking with deformable simulations, but to track a highly articulated and deformable object like a hand through self-occlusion and self-contact, we will necessarily need to treat a degree of nonlinearity not considered in previous works. These non-linearities are present in both our vision-based tracking formulation, where we directly enforce a photometric consistency term, and in our deformation formulation, where we model the hand as a volume-preserving material that seeks to avoid self-penetration.

3 TRACKING OVERVIEW

In this section, we briefly outline our hand surface tracking algorithm before discussing each component in depth in Section 4. We visualize the flow of data through our pipeline in Figure 2.

Data Capture. We first capture a hand in motion using our multi-view camera system that consists of 124 calibrated cameras with hardware synchronization, capable of capturing 2668×4096 RGB images at 30 frames per second. We perform all image-based operations at a base down-sampled resolution of 1334×2048 to conserve memory. A hand is captured at the center of the cameras and lit uniformly from static LED light sources. After capturing images, we perform PatchMatch-based multi-view stereo [Galliani et al. 2015] to obtain a 3D scan mesh for every frame. The images and 3D scan serve as input data to our tracking algorithm. Figure 2 shows example images from our capture system and a resulting 3D scan.

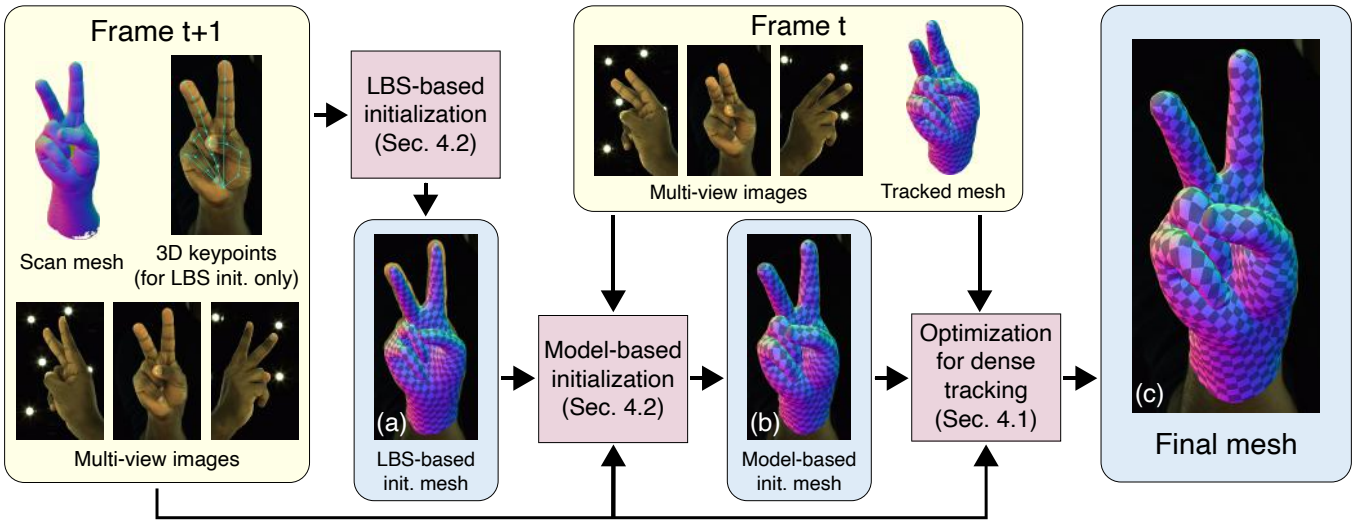

Fig. 2. Overview of our dense hand surface tracking method. Starting with 124 images from our multi-view capture system for the current frame we compute keypoints and a 3D scan mesh with PatchMatch multi-view stereo [Galliani et al. 2015]. (a) We then use these keypoints and the 3D scan to predict the pose of a subject's hand using a generic hand rig that is skinned through linear blend weights (Section 4.2). (b) We then use this pose estimate, along with a linear deformation model computed from tracking results in previous frames (if available) to obtain an initial mesh for warm starting our dense tracking optimization (Section 4.2). (c) We finally feed the initial surface estimate, the captured images at the previous and current frame, and the 3D scan into our model-free mesh tracking algorithm, which outputs our final mesh (Section 4.1).

Hand Representation. Similar to existing work on 3D tracking, we track a hand in motion using a template mesh. In our setting, the template mesh is a generic surface triangle mesh.

For every subject, we capture both the neutral pose of the hand as well as multiple motion sequences. Given a 3D scan of the neutral pose, we manually register the template surface to the scan using the commercial ZBrush[1] modeling package, producing a personalized surface mesh. This step could be further automated by fitting to an initial scan [Beeler et al. 2011; Wu et al. 2018] or by leveraging existing commercial solutions[2].

After registering the surface mesh, we create a volumetric tetrahedral mesh by feeding the personalized surface mesh to the TetGen library [Si 2015]. Across all subjects we obtain high quality constrained Delauny terahedralizations using TetGen, and we are thus able to directly read off exact correspondences from the personalized surface mesh to the tetrahedral mesh. We have also successfully tracked sequences using tetrahedral meshes computed with the TetWild [Hu et al. 2018] algorithm and inexact correspondences, but given the similar quality of output from both terahedralization routines for our meshes, we report results in this work with the output of TetGen alone.

Tracking Initialization. With personalized surface and tetrahedral meshes in hand, given a sequence of images and 3D scans, we next initialize the template meshes for these sequences using a two-stage, model-based tracking optimization (Algorithm 1). In the first stage, we deform the template using an LBS model that applies a skeleton-based transformation to local regions of the hand mesh. We use a generic embedded skeleton model and skinning weights designed once by an artist. Per tracking sequence, this generic model is personalized based on non-rigid registration to several frames in the beginning of the input sequence. This personalization step minimizes a geometric error based on the input 3D scan and the landmark error in a similar fashion to Taylor *et al.*'s method [2016] (Figure 2a).

In the second stage, we use a region-based linear deformation model [Tena et al. 2011; Wu et al. 2016] to further match the template mesh to the image and 3D scan data by optimizing the region-based model's parameters and the rigid transformations estimated in the first stage.

Algorithm 1 `Initialize_Surface` $\left(\mathbf{X}^t, \mathbf{X}^{t-1}, \ldots, \Theta^t, \Theta^{t-1}\right)$

1: $\mathbf{X}^{t+1} \leftarrow$ `Skeletal_Transformation` $\left(\mathbf{X}^t, \Theta^t, \Theta^{t-1}\right)$
2: $\mathbf{X}^{t+1} \leftarrow$ `PCA_Deformations` $\left(\mathbf{X}^{t+1}, \mathbf{X}^t, \mathbf{X}^{t-1}, \ldots\right)$
3: **return** $\mathbf{X}^{t+1}$

With the surface initialization complete, we warp the tetrahedral mesh so its surface exactly corresponds to the initialized, personalized surface triangle mesh (Figure 2b). Section 4.2 describes the initialization step in detail.

Constraining Tracking with an Elastic Physical Model. After initializing the template, we track across a sequence by solving an optimization at each frame that includes vision-based energies, a deformable elastic energy, and an energy to couple the two representations. This optimization alternates between refinements of surface vertex positions using the input data and updates of all

[1]https://pixologic.com/
[2]https://www.russian3dscanner.com/

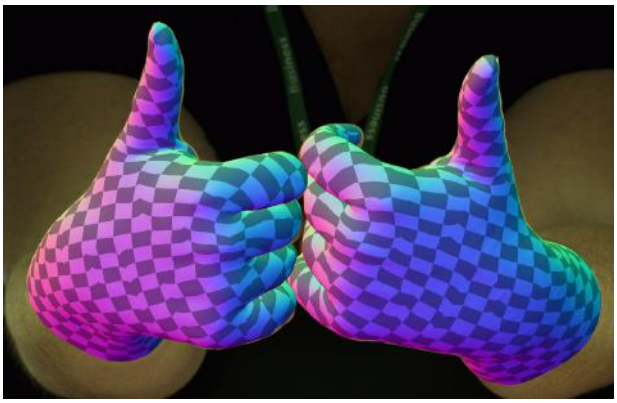

Fig. 3. Tracking results for a two-hand American Sign Language sequence. Throughout this sequence, the subject's hands execute fast and sudden motions, change shape dramatically, and come in and out of contact. Our algorithm successfully tracks the hands in the face of these challenges.

tetrahedral vertex positions using the physics term (Algorithm 2). The former optimization ensures pixel-accurate locations for visible vertices of the template mesh, while the latter optimization deforms invisible vertices in a physically feasible manner while resolving self-collisions (Figure 2c). The optimization is structured in a manner that ensures a decrease in total combined energy across each alternating iteration. Section 4.1 describes the details of the energies and optimization.

Algorithm 2 `Track_Frame` $\left(\mathbf{X}^t, \mathbf{X}^{t-1}, \ldots, \Theta^t, \Theta^{t-1}, \mathbf{V}^t\right)$

$\mathbf{X}^{t+1} \leftarrow$ `Initialize_Surface` $\left(\mathbf{X}^t, \mathbf{X}^{t-1}, \ldots, \Theta^t, \Theta^{t-1}\right)$
$\mathbf{V}^{t+1} \leftarrow \underset{\mathbf{V}}{\operatorname{argmin}} \left(E_{\text{nh}}(\mathbf{V}) + E_{\text{link}}\left(\mathbf{X}^{t+1}, \mathbf{V}\right)\right)$ ▷ Init. from $\mathbf{V}^t$
for $h = 1, 2, \ldots, 5$ **do** ▷ Image res. hierarchy, coarse-to-fine
  for $a = 1, 2$ **do** ▷ Alternating optimization
    $\mathbf{X}^{t+1} \leftarrow \underset{\mathbf{X}}{\operatorname{argmin}}(E_{\text{vision}}(\mathbf{X}, h) + E_{\text{link}}\left(\mathbf{X}, \mathbf{V}^{t+1}\right))$
    $\mathbf{V}^{t+1} \leftarrow \underset{\mathbf{V}}{\operatorname{argmin}}(E_{\text{physics}}(\mathbf{V}) + E_{\text{link}}\left(\mathbf{X}^{t+1}, \mathbf{V}\right))$
  end for
end for
return $\mathbf{X}^{t+1}, \mathbf{V}^{t+1}$

4 METHOD

In this section, we describe our initialization and optimization strategies for hand surface tracking. Note that two hand tracking does not require special treatment with our algorithm. For two-handed sequences, we simply concatenate meshes for the left and right hands and run the same algorithm as the single hand case.

4.1 Optimization for Dense Hand Surface Tracking

We formulate the combination of vision-based tracking and physically based simulation as an energy minimization problem. Denoting the surface mesh's vertices as $\mathbf{X}$ and the tetrahedral mesh's vertices as $\mathbf{V}$, the total energy is expressed as:

$$E_{\text{total}} = E_{\text{vision}}(\mathbf{X}) + E_{\text{physics}}(\mathbf{V}) + E_{\text{link}}(\mathbf{X}, \mathbf{V}). \quad (1)$$

Here E_{vision} is an image data term, which evaluates the degree to which the input multi-view data is explained by the 3D surface estimates. This term is computed over the surface template mesh only. E_{physics} is a term that enforces physical plausibility for volumetric hand deformations. This term is computed over the tetrahedral representation. Owing to the use of two separate representations for our vision and physics terms, we impose an additional term E_{link} that relates 3D surface positions to their corresponding vertices in the tetrahedral mesh. In the following, we describe each term in detail and explain our final optimization strategy for tracking from frame t to frame $t+1$ assuming that tracking for frame t and initialization for frame $t+1$ are completed. Note that the initialization step is described in Section 4.2.

4.1.1 Vision Tracking Term. Inspired by Wu *et al.*'s method [2018], we seek to directly optimize vertex positions $\mathbf{X}^{t+1}$ and their surface orientations $\mathbf{R}^{t+1}$ on the surface mesh by minimizing the vision-based energy E_{vision} that consists of photo-consistency, geometric consistency, and surface regularization terms, namely E_{pho}, E_{geo}, and E_{reg}, respectively:

$$E_{\text{vision}} = w_{\text{pho}} E_{\text{pho}} + E_{\text{geo}} + E_{\text{reg}}.$$

Photo-consistency Terms E_{pho}. Using a 3D local tangent plane for each surface vertex as a proxy, we use a homography to transform an image patch around the 2D projection of $\mathbf{X}$ in image I between t and $t+1$ to compute E_{pho}. Unlike faces, the hand can easily execute rapid rotational motions, influencing the visibility of the vertices for each view. To compensate for this issue, we develop an adaptive view selection strategy to appropriately match image patches and incorporate the selected camera parameters when computing the homography. Specifically, for vertex $\mathbf{X}^t$ and camera c at frame t, we compute the viewing angle θ_c^t based on the vertex normal and the camera view direction. At frame $t+1$, given initialized vertex $\mathbf{X}^{t+1}$, we compute the viewing angle in the same way for all the cameras, and choose the camera c_o that has the closest viewing angle to θ_c^t (see Section 4.2 for initialization). We then compute the homography $\mathbf{H}^{c,c_o}$ from the parameters of cameras c and c_o and from $\{\mathbf{X}^t, \mathbf{X}^{t+1}\}$ and $\{\mathbf{R}^t, \mathbf{R}^{t+1}\}$ at frames t and $t+1$. This approach selects image patches that share similar projective distortion and thus enables more precise computation of E_{pho}. Using this view selection strategy together with enhanced correlation coefficients [Evangelidis and Psarakis 2008] for robust image error, E_{pho} is formulated as

$$E_{\text{pho}} = \sum_v \sum_{c \in C(\mathbf{X}_v^t)} \psi\left(\left\| \frac{I_{c_o}^{t+1}(\mathbf{P}_{c_o}\mathbf{X}_v^{t+1})}{\|I_{c_o}^{t+1}(\mathbf{P}_{c_o}\mathbf{X}_v^{t+1})\|} - \frac{I_c^t(\mathbf{H}_v^{c,c_o}(\mathbf{P}_c\mathbf{X}_v^t))}{\|I_c^t(\mathbf{H}_v^{c,c_o}(\mathbf{P}_c\mathbf{X}_v^t))\|} \right\|\right)$$

where $C(\mathbf{X})$ is a set of cameras where $\mathbf{X}$ is visible in the previous frame and $\mathbf{P}_c$ is the camera matrix of camera c. Note that I in E_{pho} is mean-subtracted. $\psi(\cdot)$ is a robust kernel to handle outliers [Zollhöfer et al. 2014], formulated as

$$\psi(e) = \min_{\omega}(2\omega^2 e^2/\gamma^2 + (1-\omega^2)^2),$$

where γ for E_{pho} is set to 0.1. We use a patch size of 15×15 pixels for the photo-consistency term.

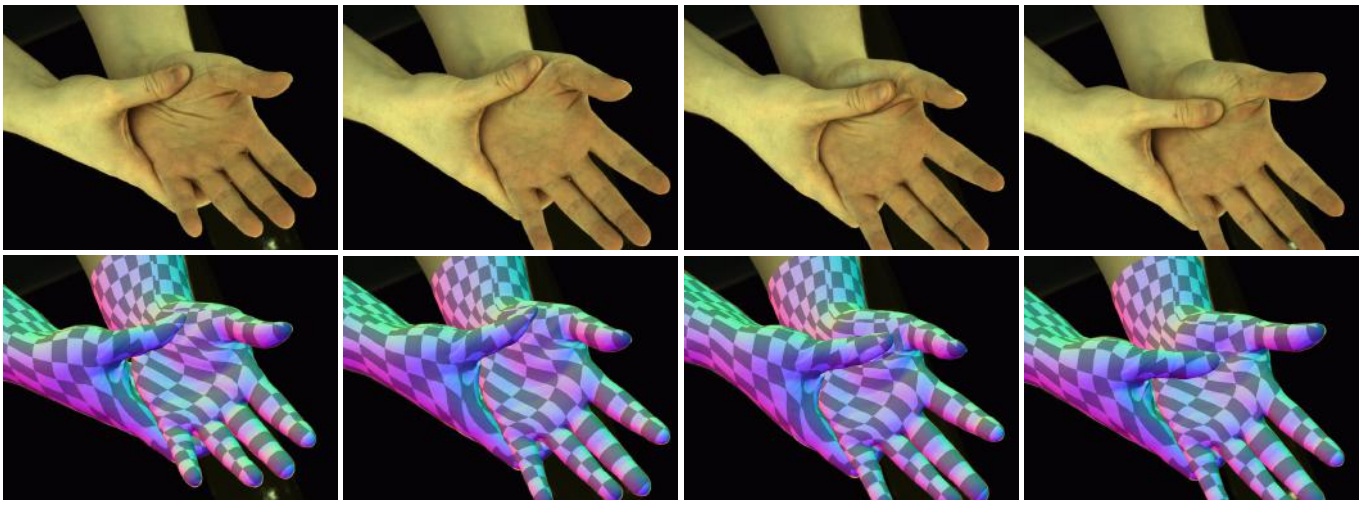

Fig. 4. Top row: A subject massages her left palm in a circular motion using her right thumb. As a result, we observe on-surface deformations of the subject's flesh. Bottom row: Our tracking results overlaid on the captured images. We are able to successfully track this motion through significant amounts of sustained self-contact. Notice that we begin to capture surface-level deformations, a result that would be difficult to achieve with a model-based tracking algorithm. Additional modeling for both skin and friction could capture finer scale creases and folds than we demonstrate here, while techniques from the face tracking literature could reduce some of the surface drift we observe around the thumb. See Section 6 for a more detailed discussion.

Geometric Consistency Term E_{geo}. Similar to the geometric consistency terms used for hand surface tracking [Tzionas et al. 2016; Mueller et al. 2019], we consider point-to-point and point-to-plane distances for each vertex $\mathbf{X}$ on the surface mesh. Given a position map D rendered from a 3D scan at frame $t+1$ for each view c, the point-to-point distance E_{pt} is formulated as

$$E_{\text{pt}} = \sum_{v} \sum_{c \in C(\mathbf{X}_v^{t+1})} \psi\left(\| \mathbf{X}_v^{t+1} - D_c^{t+1}(\mathbf{P}_c \mathbf{X}_v^{t+1}) \| \right),$$

while the point-to-plane distance E_{pl} is formulated as

$$E_{\text{pl}} = \sum_{v} \sum_{c \in C(\mathbf{X}_v^{t+1})} \psi\left(\left\| \mathbf{n}_{c,v}^{t+1} \cdot \left(\mathbf{X}_v^{t+1} - D_c^{t+1}(\mathbf{P}_c \mathbf{X}_v^{t+1}) \right) \right\| \right).$$

Here $\mathbf{n}_{c,v}^{t+1}$ is the normal of vertex v obtained from the normal map of D_c^{t+1}, and γ in the robust kernel for E_{pt} and E_{pl} is set to 5 and 1, respectively. Finally, E_{geo} is defined as

$$E_{\text{geo}} = w_{\text{pt}} E_{\text{pt}} + w_{\text{pl}} E_{\text{pl}}.$$

Surface Regularization Term E_{reg}. To avoid the generation of implausible surface shapes in unobserved or textureless regions, we regularize our surface estimation with a conventional as-rigid-as-possible (ARAP) [Sorkine and Alexa 2007] term by comparing to the surface in the previous frame:

$$E_{\text{reg}} = w_{\text{A}} \sum_{v} \sum_{i \in \mathcal{N}(v)} \| (\mathbf{X}_v^t - \mathbf{X}_i^t) - \mathbf{R}_i^{t+1} (\mathbf{X}_v^{t+1} - \mathbf{X}_i^{t+1}) \|^2,$$

where w_{A} is a weight for the ARAP regularization term (set to 0.5), and $\mathcal{N}(v)$ is a set of neighbors of vertex v. Note that, while this term allows the vision-only optimization stage to handle unseen surfaces, this surface-only regularization is not sufficient to obtain physically plausible deformations in highly occluded regions, in highly deformed regions, or in regions subject to many self-collisions. See Figure 10 for detailed comparisons.

4.1.2 Elastic Physics Term. Owing to the failure of the surface-only term to handle highly occluded and colliding regions, we add an additional physically based energy to our system. To serve as a useful physically based model for a hand, a deformation model should have a few important properties: the model should preserve volume well [Irving et al. 2007], the model should be minimal in the sense that it does not over-constrain the solution space, the model should remain robust in the face of heavy self-collisions and under large changes in shape, and the model should be suitable for inclusion in a numerical optimization setting. Smith *et al.* [2018] recently demonstrated that Neo-Hookean elastic models satisfy these properties in the context of body and hand simulation, so we have selected a variant of Neo-Hookean elasticity as the core of our physics based energy term. Note that we omit $t+1$ in this section for simplicity, but all physics terms are computed based on the tetrahedral mesh at frame $t+1$ and the neutral state.

A Neo-Hookean energy density can be written as

$$\Psi_{\text{nh}} = \frac{w_\mu}{2} \left(\text{Tr}\, \mathbf{F}^\top \mathbf{F} - 3 \right) - w_\mu \log J + \frac{w_\lambda}{2} \log^2 J,$$

where w_μ is the shear modulus, w_λ is Lamé's first parameter, $\mathbf{F}$ is the deformation gradient, and $J = \det \mathbf{F}$. The deformation gradient $\mathbf{F}$ transforms a frame in an object's rest configuration to its deformed configuration, and we can thus see that J measures relative changes to an object's volume and that the Neo-Hookean model strongly penalizes changes in a material's volume, as desired. We compute the deformation gradient $\mathbf{F}$ per tetrahedron as a linear function of each tetrahedron's vertices [Sifakis and Barbič 2012]. The total internal elastic energy is now given by evaluating the energy density at each tetrahedron, multiplying by the tetrahedron's rest volume

V_i, and summing over all tetrahedra according to

$$E_{\text{nh}} = \sum_i V_i \Psi_{\text{nh}} \left(\mathbf{F}(\mathbf{x}) \right).$$

To handle collisions, we employ a fast, reference-map based penalty approach [Hirota et al. 2001; Irving et al. 2004; McAdams et al. 2011; Smith et al. 2018]. We begin by detecting which vertices of the tetrahedral mesh's surface $\mathbf{x}^{\text{interior}}$ are interior by casting a ray along each vertex's angle-weighted normal $\mathbf{n}^{\text{interior}}$ [Jin et al. 2005] and tallying the number of ray-face intersections with the surface, where an odd number of intersections indicates a vertex is interior. For each interior surface vertex, we compute the non-incident tetrahedron with which the vertex collides, and using the barycentric coordinates in this tetrahedron, we compute the interior position within the rest configuration. We next project this rest interior position to the closest surface face, and using the barycentric coordinates of the surface face map the position back to the deformed pose, giving us an estimated target position for the interior vertex. As the deformation map is not guaranteed to preserve closest features, we iteratively check neighboring features of the target position, updating the target position if any features are closer to the interior point. This process typically terminates in one or two iterations. This final position $\mathbf{x}^{\text{target}}$ is the desired target position of the interior vertex. In the presence of very large inter-penetrations (*e.g.* a finger penetrating more than halfway through another finger), we have found that this algorithm can produce target positions that will lead to deeper penetrations. We thus perform an additional filtering step on top of existing works. Letting $\delta = \mathbf{x}^{\text{target}} - \mathbf{x}^{\text{interior}}$ and denoting the surface normal of the target feature as $\mathbf{n}^{\text{target}}$, we discard collisions where either $\delta^\top \mathbf{n}^{\text{target}} > 0$ or $\delta^\top \mathbf{n}^{\text{interior}} < 0$. This simple filter produces significantly more robust behavior during fast motion sequences where portions of the hand undergo large changes in position. We accelerate all ray-surface intersection tests using the Embree library [Wald et al. 2014], we accelerate point vs. tetrahedron intersection queries using a fast spatial hash, and we accelerate point vs. mesh closest point projections using a k-d tree. Summing over all collisions, we compute the total collision penalty energy as

$$E_{\text{collision}} = \sum_j w_{\text{col}} \left(\mathbf{x}_j^{\text{target}} - \mathbf{x}_j^{\text{interior}} \right)^2.$$

Our complete physics-based energy is now the combination of the internal Neo-Hookean energy and the collision penalty:

$$E_{\text{physics}} = E_{\text{nh}} + E_{\text{collision}}.$$

While alternative penalty formulations are possible, including barrier formulations [Harmon et al. 2009], we found this simple form to work well in practice. Alternative collision detection strategies bring unique advantages, and while our strategy has proven effective for hand tracking where significant portions of the fingers can intersect during intermediate stages of the optimization, it is interesting to consider scenarios in which alternatives could improve results. If the mesh were to experience extreme amounts of deep self-penetration, a contour-based strategy that first computes explicit contours of intersection and then computes closest points between opposing surface patches could prove more robust [Baraff et al. 2003]. Alternatively, a method that propagates constraint information to interior tetrahedral mesh vertices using the mesh's connectivity could prove more robust, but at the cost of an extra propagation step through the colliding volume [Heidelberger et al. 2004]. In performance sensitive applications, image-based strategies could further accelerate collision detection [Faure et al. 2008], and are especially appealing for their potential use on the GPU.

4.1.3 Coupling Term E_{link} and Optimization. As our vision term and physics term act on different representations, we need to link the representations to reap the benefits of each energy. To achieve this link, as noted previously, we constrain our tetrahedral mesh generation step to create a volume mesh $\mathbf{V}$ where each surface vertex of $\mathbf{V}$ has an exact correspondence to a vertex in the vision-based surface representation $\mathbf{X}$. We then define a coupling term that penalizes deviations in the $l2$ norm between vertices of the mesh position $\mathbf{X}$ and corresponding surface vertices $\mathbf{V}^{\text{surf}}$ of $\mathbf{V}$:

$$E_{\text{link}} = w_{\text{link}} \sum_v \| \mathbf{X}_v - \mathbf{V}_v^{\text{surf}} \|^2.$$

We could solve Equation (1) by simultaneously optimizing the surface vertex and tetrahedral vertex positions, but this is a large-scale, non-convex objective that presents many challenges. Fortunately, if we consider the physics-based term to be fixed, the remaining vision terms are readily solved with existing numerical machinery. Similarly, if we fix the vision-based term, the remaining physics-based term optimization is equivalent to a quasi-static optimization, the solution of which has received significant attention in the graphics community. We thus employ an alternating optimization method [Bezdek and Hathaway 2003] to solve the total optimization by first freezing the tetrahedral physics degrees of freedom and optimizing the remaining $E_{\text{vision}} + E_{\text{link}}$ terms, and next freezing the vision-based surface degrees of freedom and optimizing the remaining $E_{\text{link}} + E_{\text{physics}}$ terms. Alternating in this fashion, if each stage of the optimization decreases the energy, E_{total} will decrease with each iteration, guaranteeing progress with tracking.

To this end, we first solve the vision system to optimize the surface mesh $\mathbf{X}$ by fixing the tetrahedral mesh $\mathbf{V}$ and applying a Gauss-Newton method, which can be efficiently solved for the large number of parameters with a GPU-based implementation that computes the Jacobian matrix of each term in parallel and solves parameter updates with preconditioned conjugate gradient [Zollhöfer et al. 2014]. We then solve the physics system by optimizing $\mathbf{V}$ with $\mathbf{X}$ fixed as a data constraint. To optimize the physics-based energy, we use a projected Newton solver [Teran et al. 2005] with fast analytical Eigenvalues [Smith et al. 2019] and an inversion-avoiding line search [Smith and Schaefer 2015] that preserves local injectivity. As the Hessian is projected to positive-definiteness at each iteration of the Newton solve, we use preconditioned conjugate gradient to efficiently solve this system on the GPU. We run this alternating optimization in a coarse-to-fine manner on an image resolution hierarchy with 5 layers to capture features across scales and to improve convergence [Bergen et al. 1992; Bouguet 2001].

4.2 Tracking Initialization with Deformation Model

As Equation (1) is highly nonlinear, we need to provide a good initial guess to converge to a good local minimum. For this, we first build a region-based linear deformation model [Tena et al. 2011; Wu et al. 2016] and seek to minimize E_{vision} by model-based tracking, followed by a solve of the physics system $E_{\text{nh}} + E_{\text{link}}$ with $E_{\text{collision}}$ disabled.

Given a set of tracked surface meshes, we first uniformly segment the meshes in UV space, and select up to 10 meshes for each region as a linear deformation basis via shape similarity analysis. We parameterize the deformation of each region with 16 parameters: 6 for rigid transformations and 10 for deformation coefficients. To run this model-based tracking stably and efficiently, we first optimize the 6 rigid transformation parameters for each region via LBS-based tracking, and then optimize all the parameters of all the regions simultaneously to match to the visual data.

Similar to Taylor *et al.*'s method [2016], the LBS-based tracking in the first step seeks to minimize the geometric distance between the 3D scan and the LBS-deformed mesh and the hand keypoint distance, together with several priors such as smoothness and joint limits. For the hand keypoints, we use the convolutional pose machine algorithm [Wei et al. 2016] with the multi-view bootstrapping training method [Simon et al. 2017]. Note that these hand keypoints are used only in this initialization step.

Once the LBS-based tracking is complete, we compute a rigid transformation for each region through Procrustes alignment, and use these transforms as initial guesses for tracking with the region-based model. Instead of optimizing the full set of surface vertices $\mathbf{X}$ and their orientations $\mathbf{R}$, this model-based initialization instead optimizes E_{vision} with respect to this reduced set of degrees of freedom, a significantly faster procedure.

Finally, we seek to optimize tetrahedral mesh vertices $\mathbf{V}$ by minimizing $E_{\text{nh}} + E_{\text{link}}$ via a projected Newton solve to update the tetrahedral mesh state. Note that during initialization we disable $E_{\text{collision}}$, which in our tests stabilizes the optimization if any surface mesh vertices initially collide.

If the initial pose is far from the ground truth the photometric consistency term grows less effective, and the geometric consistency term is forced to behave in a similar fashion to a non-rigid iterative closest point formulation to recover the hand's pose. In this scenario, more alternating iterations are required to obtain similar quality results to those we present. We have found that sub-optimal initialization with insufficient iterations can lead to jittering geometry artifacts and surface level sliding artifacts as the large initial error is reduced over multiple subsequent frames. Our tests have revealed more leeway in the linear deformation analysis: while including the linear deformation bases in the initialization accelerates the first few iterations of our optimization, capturing the overall pose is much more important to achieving an artifact-free tracking result.

5 RESULTS

All captures and tracking runs were performed at 30 frames per second using 124 cameras unless noted otherwise, and all results are presented at the same frame rate. We ran all tests using six threads on a 2.2GHz Intel E5-2698 Xeon processor and a single NVIDIA Tesla V100 GPU. Please see the supplemental video for footage of results.

We tuned all of our parameters to achieve high quality tracking results on one subject, and subsequently found these parameters to perform well on all other tested subjects and motion sequences. Additional subject-specific parameter tuning did not materially alter the quality of the tracking results, and we found our algorithm to be fairly parameter insensitive. In all tests we set $w_{\text{link}} = 1$ in E_{link}, $w_{\text{pho}} = 10$, $w_{\text{pt}} = 10$, $w_{\text{pl}} = 1$, and $w_{\text{A}} = 0.5$ in E_{vision}, and $w_{\text{col}} = 1250$, $w_\mu = 100$, and $w_\lambda = 1000$ in E_{physics}. Our choices of w_μ and w_λ correspond to a Poisson's ratio of ≈ 0.455.

We note that setting w_μ to extreme values can lead to suboptimal tracking results. Setting w_μ to excessively large values prevents non-rigid deformations, in which case the tracking results reduce to a six degree of freedom rigid alignment of the reference configuration of the hand to the current configuration. Setting w_μ to an excessively small value results in the hand not faithfully preserving its shape under large deformations. After bisecting an effective value for w_μ relative to the parameters in E_{vision}, we tuned w_λ to a large enough value that preserved volume well without introducing numerical difficulties.

Robustness over variations in appearance and shape. We tested our algorithm across eight subjects from young to old, whose hands vary greatly in size, skin tone, shape, fat content, uniformity of appearance, and amount of wrinkles. See Figure 5 for results from five subjects and Figures 1, 3 and 4 for the remaining three subjects. All subjects moved their hands through a wide range of motions, across multiple poses, and through varying amounts of self-contact and self-occlusion, confirming our algorithm's robustness.

Two hand stress test: intertwined fingers. To stress test our algorithm, we asked a subject to intertwine her fingers in the most complex fashion she could envision (Figure 1). The subject brought her hands together, folded her middle fingers against one another and wedged them between opposing fingers, and proceeded to pivot her hands 180 degrees about this region of self-contact. With her hands now oppositely oriented, the subject proceeded to wiggle her middle fingers while maintaining hand contact before pulling her hands apart. Despite the significant and sustained self-contact and the occlusions from one hand to another, our algorithm successfully tracks this performance. The surface mesh for this test contains 115,618 vertices and 230,912 faces, while the tetrahedral mesh contains 174,615 vertices and 615,703 tetrahedra. Each frame takes, on average, 788 seconds to track.

Two hand stress test: one hand massaging the other. To further stress test our algorithm under self-contact, we asked a subject to vigorously massage one of her hands with the other (Figure 4). The subject quickly rotated her left hand 90 degrees and grabbed her left palm with her right hand. With her left hand firmly squeezed by her right hand, the subject proceeded to massage the entirety of her lower hand before moving on to squeeze and massage her fingers. Our algorithm is robust to these sustained periods of strong self-contacts between two hands. Further observe that in this sequence our algorithm tracks deformations on the skin surface itself due to self-collisions. The surface mesh for this test contains 126,327 vertices and 152,372 faces, while the tetrahedral mesh contains 169,239

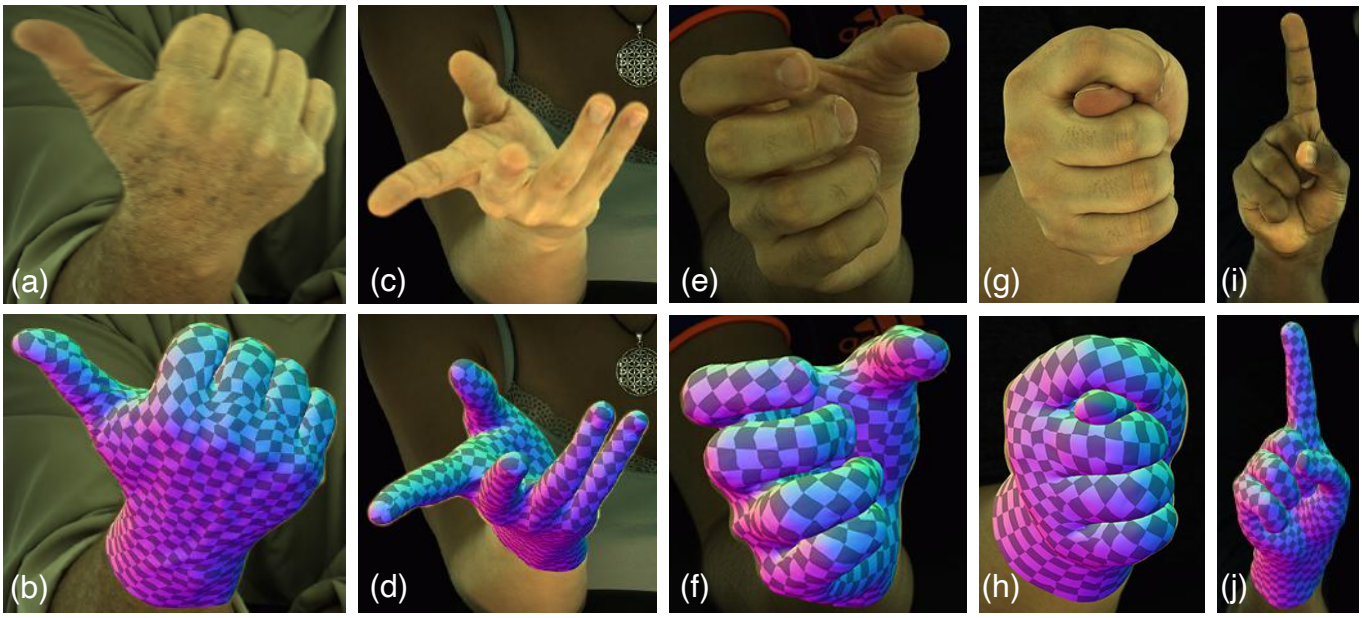

Fig. 5. Tracking results obtained by our method for five subjects. Top row: Reference images for all five subjects. Bottom row: Our tracking results overlaid on the reference images. Observe that our method robustly handles a wide range of hand shapes, ages and appearances. Further observe the range of hand poses we are able to track, including poses with significant self-occlusions and large numbers of self-collisions. All surface meshes for these subjects contain 57,809 vertices and 115,456 faces. From left to right, each tetrahedralization contains 85,935, 87,569, 85,388, 90,428, and 87,452 vertices, while each tetrahedralization contains 305,102, 308,512, 300,503, 323,642, and 308,527 tetrahedra. Finally, from left to right the average time to compute a frame for each subject is 457s, 295s, 391s, 542s, and 494s.

vertices and 581,344 tetrahedra. Each frame takes, on average, 843 seconds to track.

Two hand stress test: American Sign Language. To test our algorithm's robustness to large motions of the hands, in a real-world example we asked a subject to convey a sentence non-verbally using American Sign Language (Figure 3). During this capture, the subject deformed each of her hands significantly while also quickly translating and rotating the base of her wrist. Throughout this sequence, the subject brings her hands in and out of contact. The surface mesh for this test contains 115,618 vertices and 230,912 faces, while the tetrahedral mesh contains 181,564 vertices and 645,817 tetrahedra. Each frame takes, on average, 1,284 seconds to track.

Single hand stress test: extreme self-occlusion and self-contact. In this test, we asked a subject to exercise his hand in a manner that produced as much self-occlusion and self-contact as possible. The subject completely tucked his thumb below his fingers so it was entirely occluded from all views (Figure 6). Even with an entirely non-visible thumb in contact with all other fingers, our inclusion of a physically based deformation and collision energy enables a robust result. As the sequence continued, the subject tightly squeezed his hand into a fist, brusquely rubbed his fingers against one another, and squeezed his thumb between his other fingers (Figure 5g/h, Figure 6), all while his wrist was rotating and translating. Even with this massive amount of tight self-contact, we are able to track through the sequence. The surface mesh for this test contains 57,809 vertices and 115,456 faces, while the tetrahedral mesh contains 90,428 vertices and 323,642 tetrahedra. Each frame takes, on average, 542 seconds to track.

Ablation Study. To evaluate the impact of each of our energy terms on the final tracking results, we track a sequence with multiple variants of the total energy. We track a sequence with no physics energy term, with a volumetric ARAP ($\Psi_{\text{ARAP}} = \|\mathbf{F}-\mathbf{R}\|$) deformation term and no collision term, with a Neo-Hookean deformation term and no collision term, with no geometric consistency term, and with no photo-consistency term. See Figure 10. Without a physics term, after the fingers collide with the base of the hand, extreme artifacts emerge as the pose relaxes. While the addition of an ARAP term removes the more glaring artifacts, we observe unnatural deformation in the fingers and joints, where each segment of the finger takes on a bubble-like appearance. With the addition of a Neo-Hookean term, the local deformations of the fingers are greatly improved, but we still observe self-penetration. The addition of our collision avoidance energy resolves the remaining self-penetration, and we observe good agreement with the reference images. Removing the geometric consistency term, the fingers assume entirely incorrect poses. Removing the photo-consistency term, we observe subtle errors in the surface shape, including unnatural bumps near fingernails. We note that the photo-consistency term is important for anchoring tangential motion of the surface and preventing unnatural sliding. Please see the video for details.

Camera Count Study. While we employ a multi-view capture system with 124 calibrated cameras, smaller systems can produce similar quality results. We explore the effect of camera count on

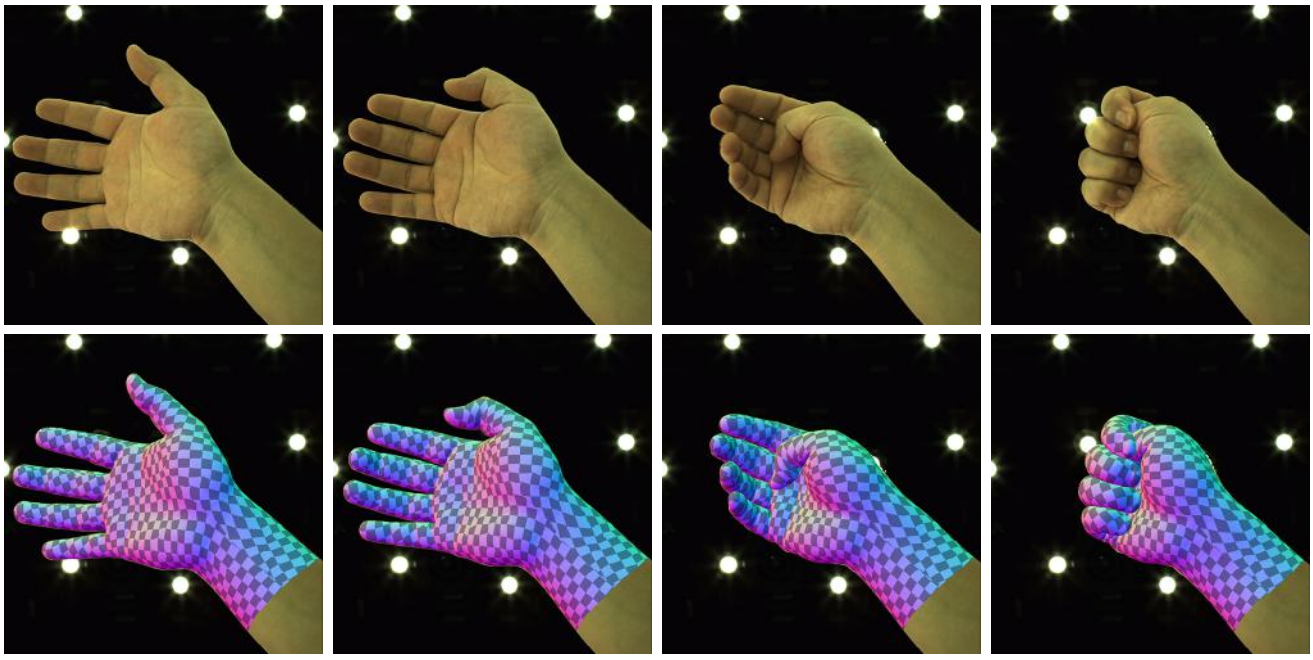

Fig. 6. A subject closes his hand, tucking his thumb beneath his fingers. Top row: Input images from a fixed camera. Bottom row: Our tracking results overlaid on the input images. Our algorithm robustly tracks this sequence through extreme occlusions around the thumb and the upper palm, and our algorithm is robust in the face of sustained collisions between the fingers, the thumb and the palm.

result quality by tracking a sequence with randomly sampled subsets of cameras from 3 to 124 cameras and comparing the final mesh to that from tracking with 124 cameras (Figure 7). We observe that improvement in distance to the 124 camera result flattens at 43 cameras, the camera count at which the hand becomes fully visible. We further observe that, visually, the 43 camera and 124 camera results are very similar (Figure 8). We hypothesize that hand-selected camera subsets designed to maximize coverage would lead to a flattening of progress even sooner and could reduce the variance in the maximum distance.

Synthetic Data Tests. The presence of highly occluded and highly self-colliding regions of the hand complicates ground truth data collection, as these regions are by definition not visible. While alternative sensor types, including those in personalized glove form factors [Glauser et al. 2019a,b], can provide data in difficult to image regions, we require higher resolution data than these sensors currently provide, and worse, the presence of these sensors changes the behavior of the underlying physical system we are trying to measure. To that end, we instead use a synthetic, animated hand mesh sequence as a source of ground truth to study our proposed formulation E_{total}.

We animated a pre-purchased, off-the-shelf hand mesh and rig to move from an open hand pose to a closed fist pose over 45 frames and rendered this sequence in V-Ray[3] at 1334×2048 resolution using 45 virtual cameras corresponding to a subset of those in our capture system. See Figure 9 for example images from this synthetic capture system. Finally, given these images of a synthetic capture session,

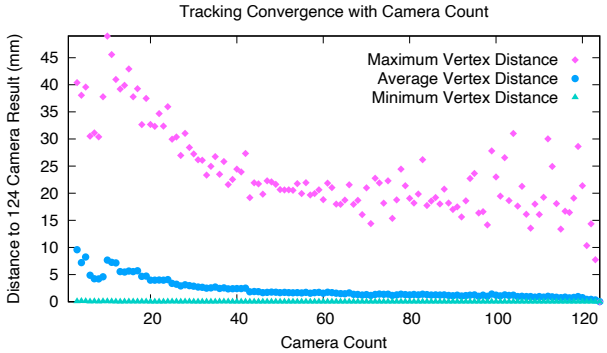

Fig. 7. As we increase the number of cameras used to track a sequence, we measure the maximum (pink diamond), average (blue circle), and minimum (green triangle) vertex distance to the result obtained with 124 cameras. After 43 cameras, progress flattens noticeably and we obtain similar meshes to the 124 camera result.

we ran our complete hand tracking pipeline for each variant of the formulation from our ablation study, producing a mesh sequence for each variant of the formulation.

We summarize the results of this synthetic capture session in Table 1. For each algorithm variant, we report the average and standard deviation of the per-vertex distances between the final tracked mesh and the ground-truth synthetic mesh, as well as the total residual penetration depth in the final frame. Notably, our proposed variant of E_{total} produces the closest mesh and least residual

[3]https://www.chaosgroup.com/

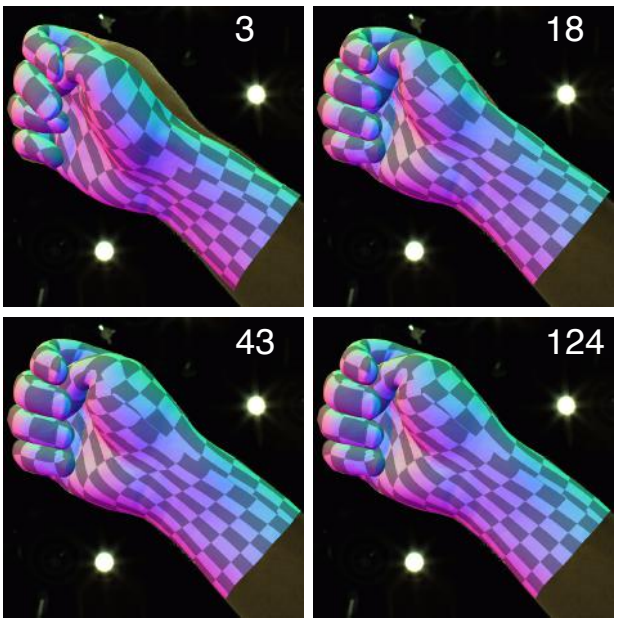

Fig. 8. Visual results for 3, 18, 43, and 124 cameras. While the 3 camera result does not closely track the input data, the 18 camera result begins to look plausible, while the 43 camera result is visually quite close to the 124 camera result.

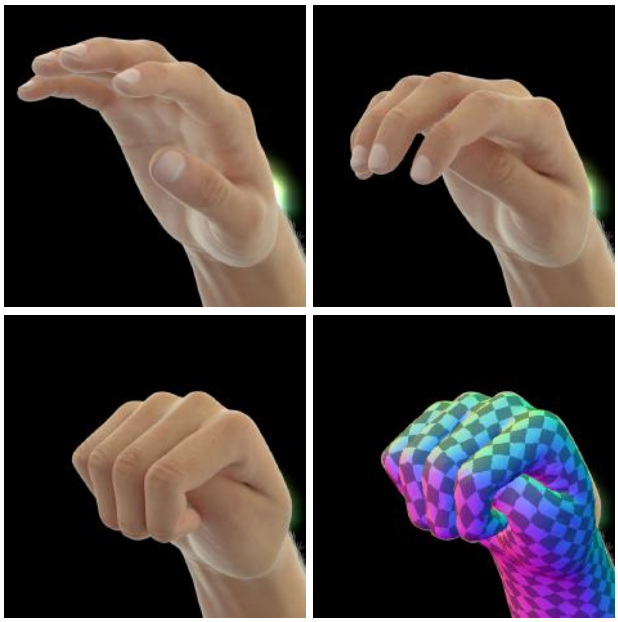

Fig. 9. Three frames from a synthetic hand ground truth test sequence and our tracking result for the final frame of this sequence.

Method	Dist. Avg.	Dist. Std. Dev.	Pen. Depth
No Physics Term	6.8310	7.4095	N/A
Volume ARAP	3.7591	4.0804	14,498.9
N.H., No Col. Term	3.3475	2.9240	44,557.5
No Photo Term	4.9100	3.4491	1.5828
No Geo. Term	3.5923	3.2300	0.9811
Our Full Energy	**3.3357**	**2.8649**	**0.8726**

Table 1. For the synthetic hand test and for each method variant from our ablation study, we report the average per-vertex distance to the ground truth mesh, the standard deviation of the per-vertex distance to the ground truth, and the total penetration depth summed over all vertices. All reported statistics are in millimeters.

Example	Total	Vision Solve	Physics Solve	Other
Fig. 1	787.89	274.39	395.16	118.35
Fig. 3	1284.58	201.90	967.24	115.43
Fig. 4	843.29	214.99	524.01	104.30
Fig. 5a/b	457.28	158.88	188.63	109.77
Fig. 5c/d	295.18	167.98	27.82	99.38
Fig. 5e/f	391.05	166.00	135.68	89.38
Figs. 6, 5g/h	541.74	158.88	284.53	98.34
Fig. 5i/j	493.97	174.06	219.39	100.52

Table 2. Wall-clock timings averaged over all frames. For each sequence, we report the average time to track a single frame, the average time to minimize the vision and link terms, the average time to minimize the physics and link terms, and the average time spent on auxiliary tasks. We report all times in seconds.

total penetration depth. Further note the relative importance of the photo-consistency term. While it is often difficult to visually spot differences between the results with and without the photo-consistency term, disabling this term leads to a 47% increase in the average vertex distance to the ground truth.

Algorithm Timings. We report wall-clock timings for all examples in Table 2. We break these profiles down into the total time, the time spent minimizing the vision and link terms $E_{\text{vision}} + E_{\text{link}}$ alone, the time spent minimizing the physics and link terms $E_{\text{physics}} + E_{\text{link}}$ alone, and the time spent on all other tasks (this includes network and storage input/output, image processing, and more). As expected, for easier highly visible sequences the cost of vision-based solves dominates, while in harder examples with many self-collisions and self-occlusions the physics-based term dominates.

We plot a more detailed wall-clock profile for Figure 1 in Figure 11. From this plot, we see that when the subject's hands are separate and visible, the minimization of the vision-based term dominates and the physics-based term does not impose major overhead. When the hands are in tight contact, we see that the cost of the vision-based optimization dips while the physics-based optimization dominates, as expected. The cost of the physics-based optimizations are in turn dominated by Hessian construction and linear system solution, suggesting that recent advances in numerical optimization for this type of system could further accelerate our algorithm.

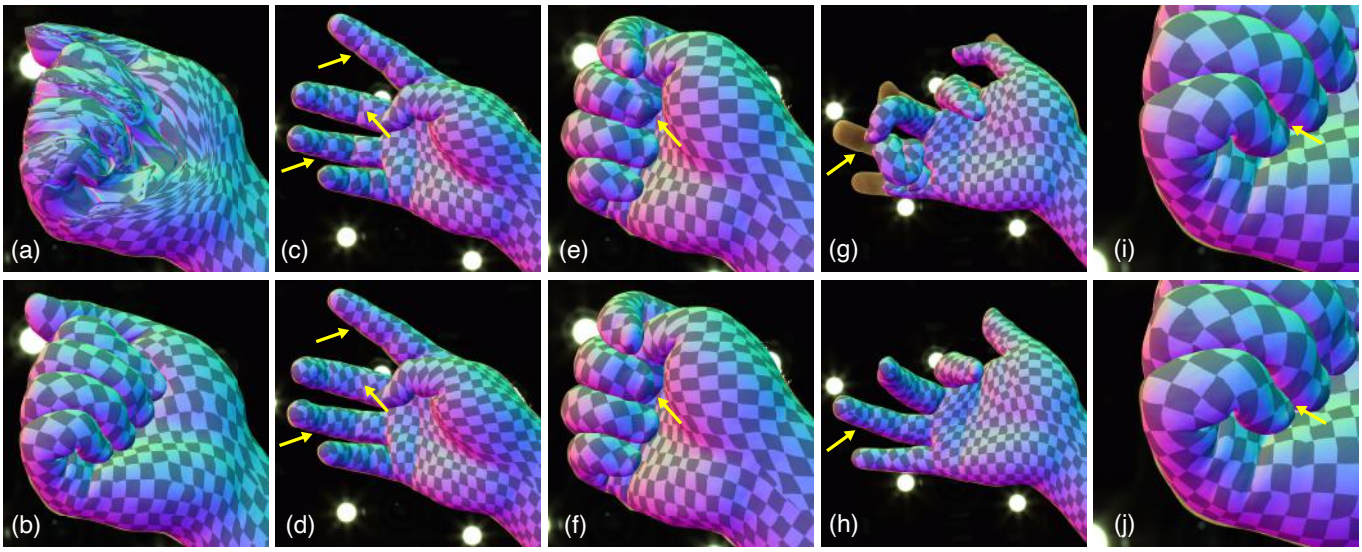

Fig. 10. Ablation study results. We label regions of interest with arrows. First column: (a) Result with no physics term. (b) Result with our full energy. Without a physics term, the tracking result has completely deteriorated after experiencing self-contact and self-occlusion. Second column: (c) Result with a volume ARAP physics term alone. (d) Result with our full energy. Observe that volume ARAP fails to maintain the fingers' shapes near joints, where they take on a balloon-like shape. The result in (d) for this preserves the finger shapes more effectively. Third column: (e) Result with no collision term. (f) Result with our full energy. Without a collision term, observe that the thumb completely penetrates the middle finger. Fourth column: (g) Result with no geometric consistency term. (h) Result with our full energy. Notice that without a geometric consistency term, fingers are in entirely incorrect positions. Fifth column: (i) Result with no photo-consistency term. (j) Result with our full energy. Notice that the photo-consistency term results in better preservation of the fingernail shape, where the result without this term has an extra bump.

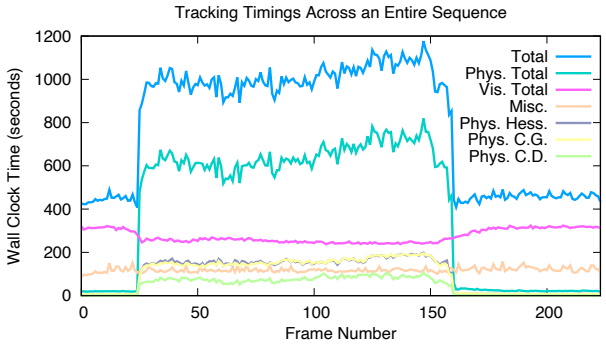

Fig. 11. Timing breakdown for each frame of Figure 1. We report the wall clock time per frame (Total), the time to minimize the physics and link terms (Phys. Total), the time to minimize the vision and link terms (Vis. Total), and the remaining time per frame to run auxiliary tasks (Misc.). We further report a subset of the total physics timings, including the time to build the Hessian (Phys. Hess), the time to solve linear systems with conjugate gradient (Phys. C.G.), and the time to detect collisions (Phys. C.D.).

6 DISCUSSION

We present an approach for precise hand tracking in situations where significant regions are occluded or in self-contact by constraining a vision-based tracking algorithm with a physically-based deformable hand model. We demonstrate the effectiveness of this approach by testing on a variety of complicated and rapid hand motions, with one and two hands, and with a number of different subjects with variations in the appearance of their hands. We further performed ablation studies on real and synthetic data to demonstrate the value of each term of our formulation, and we studied the impact of camera count on the quality of our tracking results.

Despite the improvement in tracking performance over a pure vision-based technique, our method has limitations. It is computationally expensive, as evident in Table 2. One advantage of our approach is the ability to call bespoke numerical methods for the vision-based and physically based optimizations. Incorporating recent advances in asymptotic numerical methods [Chen et al. 2014] and in quasi-Newton methods [Zhu et al. 2018] for optimizing our physically based energy could lead to faster tracking times or allow us to increase the resolution of our tracked meshes.

A second limitation of our proposed method is drift. In Figure 4, observe that over the course of the sequence, the thumb appears to twist around itself while the overall distance to the 3D scan remains small. As our method tracks hands sequentially (*i.e.*, we track frame-to-frame from the beginning to the end of a sequence), drift can accumulate over time. Existing methods on facial performance tracking mitigate the drift problem by using anchor frames [Beeler et al. 2011], by using a similarity graph between FACS scans and frames in a sequence [Fyffe et al. 2015], and by initializing each frame with a deep learning-based facial model [Wu et al. 2018]. Such approaches combined with our view selection strategy might be able to reduce drift artifacts.

A third limitation is our ability to capture high frequency folds and wrinkles. As we observe in Figure 4, detailed wrinkles from the captured images are sometimes missing in the tracked meshes. We believe that an advanced frictional simulation model [Macklin et al. 2019], improved skin modeling with thin shells [Rémillard and Kry 2013], or finer grained control of the output resolution of the tetrahedral mesh [Molino et al. 2003; Alliez et al. 2005] could enhance our ability to capture high frequency details.

While a uniform elastic Neo-Hookean model performs well in our empirical tests, recent works suggest interesting directions towards building more predictive physically based models. Wang *et al.* [2019] build subject-optimized 'bone rigs' using magnetic resonance imaging and incorporate observable anatomical features into a simulatable hand model. Previous works have explored the data-driven construction of personalized anatomical models [Cong et al. 2015; Kadleček et al. 2016; Kadleček and Kavan 2019], the use of which could improve subject-specific tracking results. Other works have championed the use of Fung hardening [Pan et al. 2015; Wang and Yang 2016] and generalized Rivlin [Pai et al. 2018] models for soft tissue, the benefits of which would be interesting to explore for tracking. Finally, while we have found uniform material settings to work well for tracking, Wang *et al.* [2019] found benefits in using spatially varying parameters for simulating folds in the hand, and it would be interesting to see whether optimizing spatially varying material parameters [Bickel et al. 2009; Wang et al. 2015; Pai et al. 2018; Sengupta et al. 2020; Weiss et al. 2020] could help us capture higher frequency folds and wrinkles.

ACKNOWLEDGMENTS

We thank Autumn Trimble, Laura Millerschoen and Taylor Koska for their assistance in planning and executing hand capture sessions. We thank Ryan Goldade for his assistance in profiling and optimizing our codebase.

REFERENCES

Pierre Alliez, David Cohen-Steiner, Mariette Yvinec, and Mathieu Desbrun. 2005. Variational Tetrahedral Meshing. *ACM Trans. Graph.* 24, 3 (2005), 617–625.

S. Baek, K. I. Kim, and T. Kim. 2019. Pushing the Envelope for RGB-Based Dense 3D Hand Pose Estimation via Neural Rendering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 1067–1076.

David Baraff, Andrew Witkin, and Michael Kass. 2003. Untangling Cloth. *ACM Trans. Graph.* 22, 3 (July 2003), 862–870.

Vincent Barrielle, Nicolas Stoiber, and Cédric Cagniart. 2016. BlendForces: A Dynamic Framework for Facial Animation. *Comput. Graph. Forum* 35, 2 (2016), 341–352.

Thabo Beeler, Derek Bradley, Bernd Bickel, and Marku Gross. 2014. Medusa Performance Capture. https://studios.disneyresearch.com/medusa/.

Thabo Beeler, Fabian Hahn, Derek Bradley, Bernd Bickel, Paul Beardsley, Craig Gotsman, Robert W. Sumner, and Markus Gross. 2011. High-Quality Passive Facial Performance Capture Using Anchor Frames. *ACM Trans. Graph.* 30, 4 (2011), 75:1–75:10.

James R. Bergen, P. Anandan, Keith J. Hanna, and Rajesh Hingorani. 1992. Hierarchical Model-Based Motion Estimation. In *Computer Vision — ECCV'92*. Springer Berlin Heidelberg, Berlin, Heidelberg, 237–252.

James C. Bezdek and Richard J. Hathaway. 2003. Convergence of Alternating Optimization. *Neural, Parallel Sci. Comput.* 11, 4 (2003), 351–368.

Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Wojciech Matusik, Hanspeter Pfister, and Markus Gross. 2009. Capture and Modeling of Non-Linear Heterogeneous Soft Tissue. *ACM Trans. Graph.* 28, 3, Article 89 (July 2009), 9 pages.

Jean-Yves Bouguet. 2001. Pyramidal Implementation of the Affine Lucas Kanade Feature Tracker Description of the Algorithm. *Intel corporation* 5, 1-10 (2001), 4.

Xiang Chen, Changxi Zheng, Weiwei Xu, and Kun Zhou. 2014. An Asymptotic Numerical Method for Inverse Elastic Shape Design. *ACM Trans. Graph.* 33, 4, Article 95 (July 2014), 11 pages.

Matthew Cong, Michael Bao, Jane L. E, Kiran S. Bhat, and Ronald Fedkiw. 2015. Fully Automatic Generation of Anatomical Face Simulation Models. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation*. Association for Computing Machinery, New York, NY, USA, 175–183.

Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. 2008. Performance Capture from Sparse Multi-View Video. *ACM Trans. Graph.* 27, 3 (2008), 1–10.

G. D. Evangelidis and E. Z. Psarakis. 2008. Parametric Image Alignment Using Enhanced Correlation Coefficient Maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 10 (2008), 1858–1865.

François Faure, Sébastien Barbier, Jérémie Allard, and Florent Falipou. 2008. Image-Based Collision Detection and Response between Arbitrary Volume Objects. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, Goslar, DEU, 155–162.

Graham Fyffe, Andrew Jones, Oleg Alexander, Ryosuke Ichikari, and Paul Debevec. 2015. Driving High-Resolution Facial Scans with Video Performance Capture. *ACM Trans. Graph.* 34, 1 (2015), 8:1–8:14.

G. Fyffe, K. Nagano, L. Huynh, S. Saito, J. Busch, A. Jones, H. Li, and P. Debevec. 2017. Multi-View Stereo on Consistent Face Topology. *Comput. Graph. Forum* 36, 2 (2017), 295–309.

S. Galliani, K. Lasinger, and K. Schindler. 2015. Massively Parallel Multiview Stereopsis by Surface Normal Diffusion. In *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 873–881.

L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan. 2019. 3D Hand Shape and Pose Estimation From a Single RGB Image. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 10825–10834.

Oliver Glauser, Daniele Panozzo, Otmar Hilliges, and Olga Sorkine-Hornung. 2019a. Deformation Capture via Self-Sensing Capacitive Arrays. *ACM Trans. Graph.* 38, 2 (2019), 16:1–16:16.

Oliver Glauser, Shihao Wu, Daniele Panozzo, Otmar Hilliges, and Olga Sorkine-Hornung. 2019b. Interactive Hand Pose Estimation using a Stretch-Sensing Soft Glove. *ACM Trans. Graph.* 38, 4 (2019), 41:1–41:15.

David Harmon, Etienne Vouga, Breannan Smith, Rasmus Tamstorf, and Eitan Grinspun. 2009. Asynchronous Contact Mechanics. *ACM Trans. Graph.* 28, 3, Article 87 (July 2009), 12 pages.

Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid. 2019. Learning Joint Reconstruction of Hands and Manipulated Objects. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 11799–11808.

Bruno Heidelberger, Matthias Teschner, Richard Keiser, Matthias Müller, and Markus H Gross. 2004. Consistent Penetration Depth Estimation for Deformable Collision Response. In *VMV*, Vol. 4. 339–346.

G. Hirota, S. Fisher, A. State, C. Lee, and H. Fuchs. 2001. An Implicit Finite Element Method for Elastic Solids in Contact. In *Proceedings of Computer Animation*. 136–254.

Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2018. Tetrahedral Meshing in the Wild. *ACM Trans. Graph.* 37, 4, Article 60 (July 2018), 14 pages.

Nikolaos Kyriazis Iason Oikonomidis and Antonis Argyros. 2011. Efficient Model-Based 3D Tracking of Hand Articulations using Kinect. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 101.1–101.11.

Geoffrey Irving, Craig Schroeder, and Ronald Fedkiw. 2007. Volume Conserving Finite Element Simulations of Deformable Models. *ACM Trans. Graph.* 26, 3 (July 2007), 13:1–13:6.

G. Irving, J. Teran, and R. Fedkiw. 2004. Invertible Finite Elements for Robust Simulation of Large Deformation. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, Goslar, DEU, 131–140.

Shuangshuang Jin, Robert R. Lewis, and David West. 2005. A Comparison of Algorithms for Vertex Normal Computation. *The Visual Computer* 21, 1 (01 Feb 2005), 71–82.

Petr Kadleček, Alexandru-Eugen Ichim, Tiantian Liu, Jaroslav Křivánek, and Ladislav Kavan. 2016. Reconstructing Personalized Anatomical Models for Physics-Based Body Animation. *ACM Trans. Graph.* 35, 6, Article 213 (Nov. 2016), 13 pages.

Petr Kadleček and Ladislav Kavan. 2019. Building Accurate Physics-Based Face Models from Data. *Proc. ACM Comput. Graph. Interact. Tech.* 2, 2, Article 15 (July 2019), 16 pages.

Michael Kass, Andrew Witkin, and Demetri Terzopoulos. 1988. Snakes: Active Contour Models. *International Journal of Computer Vision* 1, 4 (1988), 321–331.

J. P. Lewis, Ken Anjyo, Taehyun Rhee, Mengjie Zhang, Fred Pighin, and Zhigang Deng. 2014. Practice and Theory of Blendshape Facial Models. In *Eurographics 2014 - State of the Art Reports*. The Eurographics Association.

Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Viktor Makoviychuk. 2019. Non-Smooth Newton Methods for Deformable Multi-Body Dynamics. *ACM Trans. Graph.* 38, 5, Article 140 (Oct. 2019), 20 pages.

Aleka McAdams, Yongning Zhu, Andrew Selle, Mark Empey, Rasmus Tamstorf, Joseph Teran, and Eftychios Sifakis. 2011. Efficient Elasticity for Character Skinning with Contact and Collisions. *ACM Trans. Graph.* 30, 4, Article 37 (July 2011), 12 pages.

Dimitri Metaxas and Demetri Terzopoulos. 1993. Shape and Nonrigid Motion Estimation through Physics-Based Synthesis. *IEEE Transactions on Pattern Analysis and Machine*

Intelligence 15, 6 (1993), 580–591.

Neil Molino, Robert Bridson, Joseph Teran, and Ronald Fedkiw. 2003. A Crystalline, Red Green Strategy for Meshing Highly Deformable Objects with Tetrahedra. In *IMR*. 103–114.

Franziska Mueller, Micah Davis, Florian Bernard, Oleksandr Sotnychenko, Mickeal Verschoor, Miguel A. Otaduy, Dan Casas, and Christian Theobalt. 2019. Real-time Pose and Shape Reconstruction of Two Interacting Hands With a Single Depth Camera. *ACM Trans. Graph.* 38, 4 (2019), 49:1–49:13.

Matthias Müller, Julie Dorsey, Leonard McMillan, Robert Jagnow, and Barbara Cutler. 2002. Stable Real-Time Deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '02)*. Association for Computing Machinery, New York, NY, USA, 49–54.

I. Oikonomidis, N. Kyriazis, and A. A. Argyros. 2011. Full DOF Tracking of a Hand Interacting with an Object by Modeling Occlusions and Physical Constraints. In *2011 International Conference on Computer Vision*. IEEE, 2088–2095.

Dinesh K. Pai, Austin Rothwell, Pearson Wyder-Hodge, Alistair Wick, Ye Fan, Egor Larionov, Darcy Harrison, Debanga Raj Neog, and Cole Shing. 2018. The Human Touch: Measuring Contact with Real Human Soft Tissues. *ACM Trans. Graph.* 37, 4, Article 58 (July 2018), 12 pages.

Zherong Pan, Hujun Bao, and Jin Huang. 2015. Subspace Dynamic Simulation Using Rotation-Strain Coordinates. *ACM Trans. Graph.* 34, 6, Article 242 (Oct. 2015), 12 pages.

Olivier Rémillard and Paul G. Kry. 2013. Embedded Thin Shells for Wrinkle Simulation. *ACM Trans. Graph.* 32, 4, Article 50 (July 2013), 8 pages.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Trans. Graph.* 36, 6 (2017), 245:1–245:17.

J. Schulman, A. Lee, J. Ho, and P. Abbeel. 2013. Tracking Deformable Objects with Point Clouds. In *2013 IEEE International Conference on Robotics and Automation*. IEEE, 1130–1137.

Agniva Sengupta, Romain Lagneau, Alexandre Krupa, Eric Marchand, and Maud Marchal. 2020. Simultaneous Tracking and Elasticity Parameter Estimation of Deformable Objects. In *IEEE Int. Conf. on Robotics and Automation, ICRA'20*. IEEE.

Hang Si. 2015. TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator. *ACM Trans. on Mathematical Software* 41, 2 (2015), 11:1–11:36.

Eftychios Sifakis and Jernej Barbič. 2012. FEM Simulation of 3D Deformable Solids: A Practitioner's Guide to Theory, Discretization and Model Reduction. In *ACM SIGGRAPH 2012 Courses*. Association for Computing Machinery, New York, NY, USA, 50.

T. Simon, H. Joo, I. Matthews, and Y. Sheikh. 2017. Hand Keypoint Detection in Single Images Using Multiview Bootstrapping. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 4645–4653.

Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable Neo-Hookean Flesh Simulation. *ACM Trans. Graph.* 37, 2, Article 12 (2018), 12 pages.

Breannan Smith, Fernando De Goes, and Theodore Kim. 2019. Analytic Eigensystems for Isotropic Distortion Energies. *ACM Trans. Graph.* 38, 1, Article 3 (Feb. 2019), 15 pages.

Jason Smith and Scott Schaefer. 2015. Bijective Parameterization with Free Boundaries. *ACM Trans. Graph.* 34, 4, Article 70 (July 2015), 9 pages.

Olga Sorkine and Marc Alexa. 2007. As-Rigid-As-Possible Surface Modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*. Eurographics Association, Goslar, DEU, 109–116.

S. Sridhar, A. Oulasvirta, and C. Theobalt. 2013. Interactive Markerless Articulated Hand Motion Tracking Using RGB and Depth Data. In *2013 IEEE International Conference on Computer Vision*. IEEE, 2456–2463.

Richard Szeliski and Demetri Terzopoulos. 1991. Physically-Based and Probabilistic Models for Computer Vision. In *Geometric Methods in Computer Vision*, Vol. 1570. International Society for Optics and Photonics, Springer, 140–152.

D. J. Tan, T. Cashman, J. Taylor, A. Fitzgibbon, D. Tarlow, S. Khamis, S. Izadi, and J. Shotton. 2016. Fits Like a Glove: Rapid and Reliable Hand Shape Personalization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 5610–5619.

Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, Arran Topalian, Erroll Wood, Sameh Khamis, Pushmeet Kohli, Shahram Izadi, Richard Banks, Andrew Fitzgibbon, and Jamie Shotton. 2016. Efficient and Precise Interactive Hand Tracking through Joint, Continuous Optimization of Pose and Correspondences. *ACM Trans. Graph.* 35, 4 (2016), 143:1–143:12.

Jonathan Taylor, Vladimir Tankovich, Danhang Tang, Cem Keskin, David Kim, Philip Davidson, Adarsh Kowdle, and Shahram Izadi. 2017. Articulated Distance Fields for Ultra-Fast Tracking of Hands Interacting. *ACM Trans. Graph.* 36, 6 (2017), 244:1–244:12.

J. Rafael Tena, Fernando De la Torre, and Iain Matthews. 2011. Interactive Region-Based Linear 3D Face Models. *ACM Trans. Graph.* 30, 4 (2011), 76:1–76:10.

Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. 2005. Robust Quasistatic Finite Elements and Flesh Simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Association for Computing Machinery, New York, NY, USA, 181–190.

Demetri Terzopoulos, Andrew Witkin, and Michael Kass. 1987. Symmetry-Seeking Models and 3D Object Reconstruction. *International Journal of Computer Vision* 1, 3 (1987), 211–221.

Demetri Terzopoulos, Andrew Witkin, and Michael Kass. 1988. Constraints on Deformable Models: Recovering 3D Shape and Nonrigid Motion. *Artificial Intelligence* 36, 1 (1988), 91–123.

Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. 2016. Sphere-Meshes for Real-Time Hand Modeling and Tracking. *ACM Trans. Graph.* 35, 6 (2016), 222:1–222:11.

Anastasia Tkach, Andrea Tagliasacchi, Edoardo Remelli, Mark Pauly, and Andrew Fitzgibbon. 2017. Online Generative Model Personalization for Hand Tracking. *ACM Trans. Graph.* 36, 6 (2017), 243:1–243:11.

Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. 2016. Capturing Hands in Action using Discriminative Salient Points and Physics Simulation. *International Journal of Computer Vision* 118, 2 (2016), 172–193.

Ingo Wald, Sven Woop, Carsten Benthin, Gregory S. Johnson, and Manfred Ernst. 2014. Embree: A Kernel Framework for Efficient CPU Ray Tracing. *ACM Trans. Graph.* 33, 4, Article 143 (July 2014), 8 pages.

Bohan Wang, George Matcuk, and Jernej Barbič. 2019. Hand Modeling and Simulation Using Stabilized Magnetic Resonance Imaging. *ACM Trans. Graph.* 38, 4, Article 115 (July 2019), 14 pages.

Bin Wang, Longhua Wu, KangKang Yin, Uri Ascher, Libin Liu, and Hui Huang. 2015. Deformation Capture and Modeling of Soft Objects. *ACM Trans. Graph.* 34, 4, Article 94 (July 2015), 12 pages.

Huamin Wang and Yin Yang. 2016. Descent Methods for Elastic Body Simulation on the GPU. *ACM Trans. Graph.* 35, 6, Article 212 (Nov. 2016), 10 pages.

Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional Pose Machines. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 4724–4732.

S. Weiss, R. Maier, D. Cremers, R. Westermann, and N. Thuerey. 2020. Correspondence-Free Material Reconstruction using Sparse Surface Constraints. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 4685–4694.

Chenglei Wu, Derek Bradley, Markus Gross, and Thabo Beeler. 2016. An Anatomically-Constrained Local Deformation Model for Monocular Face Capture. *ACM Trans. Graph.* 35, 4 (2016), 115:1–115:12.

Chenglei Wu, Takaaki Shiratori, and Yaser Sheikh. 2018. Deep Incremental Learning for Efficient High-Fidelity Face Tracking. *ACM Trans. Graph.* 37, 6 (2018), 234:1–234:12.

Stefanie Wuhrer, Jochen Lang, Motahareh Tekieh, and Chang Shu. 2015. Finite Element Based Tracking of Deforming Surfaces. *Graphical Models* 77 (2015), 1–17.

Shanxin Yuan, Guillermo Garcia-Hernando, Björn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Liuhao Ge, Junsong Yuan, Xinghao Chen, Guijin Wang, Fan Yang, Kai Akiyama, Yang Wu, Qingfu Wan, Meysam Madadi, Sergio Escalera, Shile Li, Dongheui Lee, Iason Oikonomidis, Antonis Argyros, and Tae-Kyun Kim. 2018. Depth-Based 3D Hand Pose Estimation: From Current Achievements to Future Goals. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2636–2645.

Yufeng Zhu, Robert Bridson, and Danny M. Kaufman. 2018. Blended Cured Quasi-Newton for Distortion Optimization. *ACM Trans. Graph.* 37, 4, Article 40 (July 2018), 14 pages.

Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, and Marc Stamminger. 2014. Real-Time Non-Rigid Reconstruction Using an RGB-D Camera. *ACM Trans. Graph.* 33, 4, Article 156 (July 2014), 12 pages.