RGB-assisted Depth-based Registration for Accurate Hand Tracking

Submission Id 154

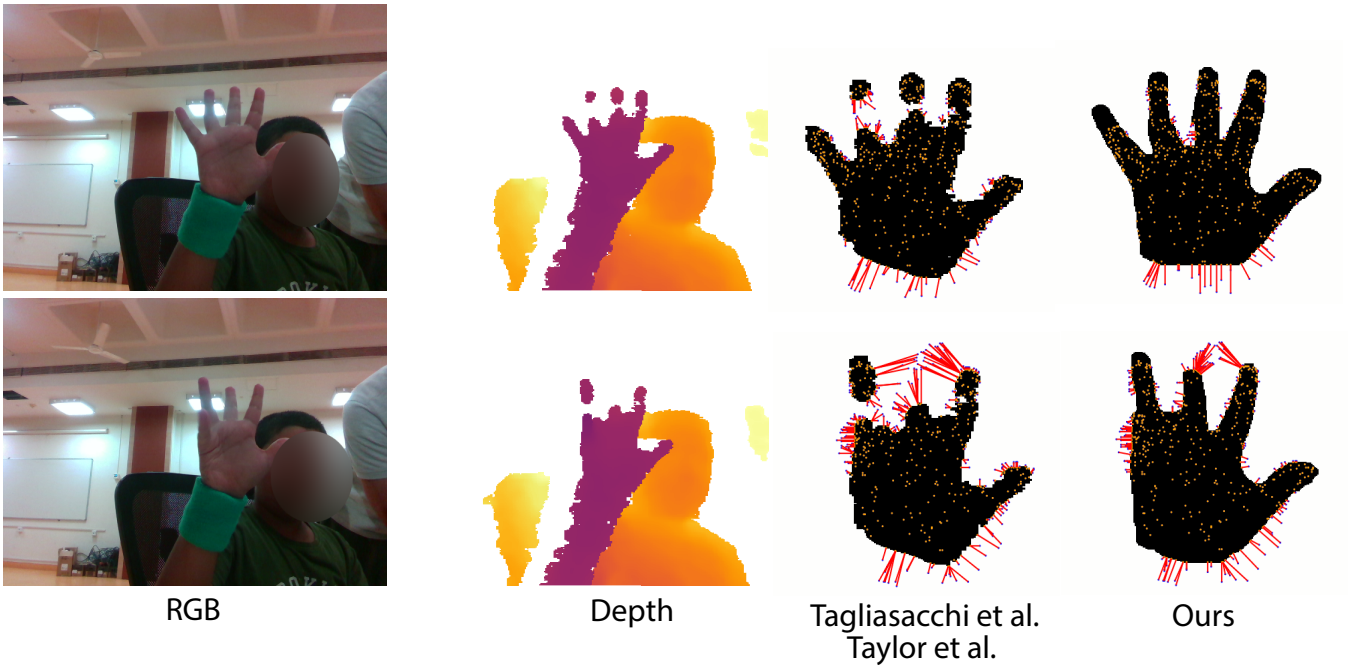

Figure 1: We present a method to use an RGB image to assist depth-based registration of a hand mesh. We highlight the limitation of existing methods that use a depth silhouette, where the correspondences (red lines) between the model (orange points) and the observation (black silhouette) are incorrect, leading to incorrect registration. Our RGBD silhouette is more reliable and thus the resulting correspondences are accurate.

ABSTRACT

Hand tracking is an essential component in modern virtual reality systems. Despite the presence of RGBD cameras in such systems, most existing methods have demonstrated hand tracking exclusively from either depth or RGB cameras. We present the first method to simultaneously leverage depth and RGB images for hand tracking. We formulate an energy minimization framework that registers a rigged hand mesh to RGB and depth data. To this end, we propose a novel method to extract a more accurate silhouette by utilizing information from both RGB and depth images. Our method outperforms state-of-the-art hand tracking methods using only RGB or depth information on publicly available datasets. We also demonstrate that our energy minimization framework is flexible enough to adapt to work with only RGB or only depth, or both. We believe this work lays the foundation for simultaneously using RGB and depth cameras for more accurate shape registration.

ICVGIP'22, December 2022, Gandhinagar, India

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

CCS CONCEPTS

• **Computing methodologies** → **Reconstruction**; *Image segmentation.*

KEYWORDS

RGBD Hand Tracking, Shape and pose estimation

ACM Reference Format:
Submission Id 154. 2022. RGB-assisted Depth-based Registration for Accurate Hand Tracking. In *Proceedings of 13th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'22).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Hands are the most natural way of interaction for humans. Understanding human hand motion from videos has been quite popular over the past few decades [18, 20, 23, 36, 40]. With the advent of virtual and augmented reality, hand tracking has gained a renewed interest in academia and industry [22, 24, 30, 42].

Although modern VR headsets are equipped with RGBD cameras capable of acquiring RGB and depth images of the user's hand, most existing methods exclusively use RGB [2, 3, 6, 15, 39, 43] or depth [8, 11, 20, 22, 26, 27, 29, 30, 30–33, 35, 36, 38, 41] images as input. Depth-based methods are more accurate than RGB-based methods because depth information disambiguates the 3D pose that might appear similar in the 2D projected space. However, the quality of depth images often suffers from limited accuracy and stability due to holes at object boundaries or smooth and shiny surfaces and inconsistent depth values across frames [16]. As a result, the depth images cannot provide reliable data required for accurate registration.

We present a method that utilizes information from both RGB and depth images to achieve a more robust and accurate registration. (see Fig. 1) Specifically, we register the parametric hand mesh model, aMANO [9], to the point cloud obtained from the depth image and silhouette extracted from both RGB and depth images. Our novel silhouette extractor provides reliable data for accurate registration even from noisy depth data. Further, our registration method is adaptable to the available data source, i.e., only RGB, or only depth, or both.

Unlike existing methods [12, 15] that optimize the joint location alignment, our registration energy also minimizes surface-to-surface distance (whenever depth is available) and thus achieves a more accurate registration. When only the RGB image is present, our registration procedure achieves a more accurate and kinematically valid pose than state-of-the-art iterative 2D joint location alignment [15].

We summarize our contributions below.

- We present the first method to utilize information from RGB and depth images to achieve a more accurate hand tracking.
- We also propose a novel silhouette extractor that provides reliable silhouette information for robust and accurate registration.

2 RELATED WORK

The problem of markerless hand tracking is closely associated with hand pose estimation and full 3D hand shape and pose reconstruction. We refer the reader to Armagan et al. [1] for an in-depth overview of hand pose estimation. In this section, we discuss methods that reconstruct the geometry of the hand from RGB and depth images.

2.1 Depth Hand Tracking

Oikonomidis et al. [20] introduced one of the early methods for reconstructing a primitive-based geometric hand model from depth images using particle swarm optimization (PSO). Qian et al. [22] demonstrate the use of ICP with PSO for tracking a sphere-based hand model. Tagliasacchi et al. [30] introduce the idea of using a gradient-based optimization, Levenberg-Marquardt (LM), to register a cylinder-based hand model to depth data. Tkach et al. [34] further extended the idea to a sphere-mesh model. We adapt the energy terms used in their method and use them in our registration framework. However, unlike these methods, we use a mesh-based model that can be easily integrable into an existing graphics pipeline.

Taylor et al. [33] introduced the first method to model a user-specific mesh from depth images, and Sharp et al. [26] used it in their robust hand tracker. Taylor et al. [32] use the hand model of Khamis et al. [11] to track a subdivided mesh. Further, Shen et al. [27] demonstrated that one could get away without subdividing the mesh using ideas from Phong shading. Our barycentric sampling of the mesh follows this approach. Unfortunately, unlike ours, these implementations are not publicly available, restricting their usage for scientific research.

Until recently, most methods used a fixed hand template model. Inspired by the recent advances in the human body shape model (e.g., SMPL [14]), Romero et al. [24] proposed a learned blend-shape hand mesh model, MANO. However, it cannot adapt to unseen hand shapes with substantially large deviations from the training set. Kalshetti and Chaudhuri [9] introduced adaptive MANO (aMANO), which extends MANO by incorporating local scale adaptation parameters, to be used in a tracking framework Kalshetti and Chaudhuri [10]. However, these methods require depth images as input.

2.2 RGB Hand Tracking

de La Gorce et al. [5] introduced the idea of analysis-by-synthesis to track hands from monocular RGB video. However, their generative method is sensitive to initialization and not robust to any unseen background. Boukhayma et al. [3] introduce the first end-to-end deep learning method that regress MANO shape and pose parameters from RGB images. Zhou et al. [46] propose a real-time method to regress joint rotations using an inverse kinematics module. Yang et al. [43] recover the mesh using a multi-stage hourglass network. Chen et al. [4] reconstruct the hand mesh using self-supervision from the texture and lighting in the input image along with detected 2D keypoints. Kulon et al. [13] incorporate mesh convolutions for recovering the hand mesh. However, the accuracy of these discriminative methods suffers on unseen data.

Baek et al. [2] introduce the idea of iterative refinement using neural rendering. However, it assumes the availability of a 2D segmentation mask. Zhang et al. [45] also use a silhouette-based self-supervision loss. Panteleris et al. [21] use a non-linear least squares minimization to fit a 3D hand model to 2D keypoints detected using OpenPose [28]. Mueller et al. [19] use 3D joint locations for fitting a hand model. Although these hybrid methods improve over discriminative-based methods, they are not as accurate as depth-based tracking methods. Our method uses a registration-based optimization method that leverages RGB and depth data.

3 METHOD

Given a sequence of RGB $\left\{I_c^{(f)}\right\}_{f=1}^{n_f}$ and corresponding depth $\left\{I_d^{(f)}\right\}_{f=1}^{n_f}$ images acquired from a single RGBD camera, our goal is to register a hand model, M, to each of these frames. We formulate the registration problem as an energy minimization such that at each frame f, the hand model, $M^{(f)}$, explains the observed data in the RGB image, $I_c^{(f)}$, and the depth image, $I_d^{(f)}$. Specifically, we register the hand model to the point cloud obtained from the depth image

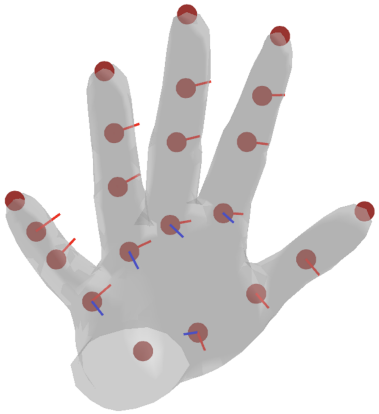

Figure 2: Degrees of freedom (DoF) at each joint: two at the metacarpophalangeal (MCP) joint, one at each proximal interphalangeal (PIP) and distal interphalangeal (DIP) joints of a finger. We plot the axis of rotation for each DoF.

and the silhouette obtained from both the RGB and the depth image. Our novel RGBD registration procedure outperforms existing state-of-the-art methods, as evident in Sec. 4.3.

We now describe the hand model used in our registration framework.

3.1 Hand Model

We use the parametric hand mesh model, aMANO [9], $M(\phi, \beta, \theta) \in \mathbb{R}^{n_v \times 3}$ with fixed triangulation among the n_v vertices. Here $\phi \in \mathbb{R}^{n_b}$ represents the user-specific local scale parameters for each of the n_b bones, calculated as

$$\phi_j = \frac{l_j^{(data)}}{l_j^{(template)}} \tag{1}$$

where $l_j^{(data)} \in \mathbb{R}$ is the length of the j^{th} bone in the observed data, obtained from keypoints (each bone connects two keypoints), and $l_j^{(template)} \in \mathbb{R}$ is the length of the j^{th} bone in the template mesh model. $\beta \in \mathbb{R}^{10}$ captures the user-specific shape parameters corresponding to the PCA shape blends of MANO [24]. The pose parameters are encoded by $\theta \in \mathbb{R}^{20}$ which represent the angles at each of the fixed axes of rotation for each degree of freedom (DoF) as shown in Fig. 2.

We now describe the procedure to deform the aMANO mesh. First, a template mesh vertex $\bar{v}_i$ is offset as

$$v_i = \bar{v}_i + S_i\beta + P_i\left(r(\theta) - r(\bar{\theta})\right) \tag{2}$$

where $S_i \in \mathbb{R}^{3\times10}$ and $P_i \in \mathbb{R}^{3\times135}$ are the shape and pose blend shapes corresponding to the vertex v_i, $\beta \in \mathbb{R}^{10}$ is the shape parameter, $\theta \in \mathbb{R}^{15\times3}$ is the pose parameter capturing the axis angle rotation at each of the 15 joints, and $r(\theta) \in \mathbb{R}^{135}$ is the vectorized version of the stacked rotation matrices at each joint with pose θ; $\bar{\theta}$ is the rest pose.

We then use the modified linear blend skinning [9] to pose the mesh. Let $a_j \in \mathbb{R}^3$ and $b_j \in \mathbb{R}^3$ be the start and end positions of j^{th} bone in rest pose mesh (after applying MANO shape-blends) respectively, and $R_j \in \mathbb{R}^{3\times3}$ be the rotation matrix that takes bone j's rest vector $(b_j - a_j)$ to its pose vector $(b'_j - a'_j)$. Now, the deformed vertex v'_i is given by

$$v'_i = \sum_{j=1}^{n_b} W_{b_{ij}} \left\{a'_j + R_j\left(W_{e_{ij}} s_j + (-a_j + v_i)\right)\right\} \tag{3}$$

where $s_j = (\phi_j - 1)(b_j - a_j)$, and $W_b \in \mathbb{R}^{n_v\times n_b}$ and $W_e \in \mathbb{R}^{n_v\times n_b}$ are the bone weight and endpoint weight matrices.

We also use a sparse regression matrix $K \in \mathbb{R}^{21\times n_v}$ to calculate the 3D keypoints, $k \in \mathbb{R}^{21\times3}$, from vertices, $v \in \mathbb{R}^{n_v\times3}$ as

$$k = Kv \tag{4}$$

3.2 RGBD Registration

We now detail our registration procedure that utilizes information from both RGB and depth images. Our RGBD registration procedure involves minimizing a registration energy to optimize the model parameters $\left\{\beta^{(f)}, \theta^{(f)}\right\}$ for each frame f. In the subsequent discussion, we drop the superscript f denoting the frame index for brevity. We use a sum of weighted terms in our registration energy as

$$E(\beta, \theta) = \sum_{\tau\in\mathcal{T}} \omega_\tau E_\tau(\beta, \theta) \tag{5}$$

where the terms E_τ are:

$E_{data_{3D}}$	the model explains the depth point cloud
$E_{data_{2D}}$	the model lies inside the observed sensor silhouette
E_{bound}	angle at each joint should respect the kinematic bounds
E_{pca}	hand pose lies in a low-dimensional manifold
E_{int}	fingers cannot inter-penetrate
E_{reinit}	model's fingertips are close to detected fingertips
E_{shape}	avoid drifting from human hand shape
E_{temp}	avoid jittery tracking

We refer the reader to Kalshetti and Chaudhuri [10] for details about these terms where the data terms only use the depth image. We adapt the $E_{data_{2D}}$ and E_{reinit} terms to leverage information from the RGB image too. In the absence of a depth image, we use an RGB-based hand pose estimation method [28, 37] to detect fingertips from the RGB image in the E_{reinit} term for achieving robust tracking.

We now focus our attention to the $E_{data_{2D}}$ term

$$E_{data_{2D}}(\theta) = \sum_{i=1}^{n_{2D}} \|q_i - p_i(\theta)\|^2 \tag{6}$$

where $p_i \in \mathbb{R}^2$ is the 2D image space position of the evaluated predefined barycenter, and $q_i \in \mathbb{R}^2$ is the 2D image space position of the closest point on the silhouette, computed using the distance transform [17] of the cropped depth image. This term is similar to the E_{m2d} in Tkach et al. [34] and E_{bg} in Taylor et al. [32]. This term penalizes the projection of the model outside the observed silhouette. However, the silhouette obtained from the depth image is inaccurate because of missing regions, as shown in Fig. 3. Thus, the resulting distance transform is incorrect, leading to wrong correspondences, q_i.

Instead, we use the information from the RGB image to extract a more reliable silhouette described below.

3.3 Silhouette Extraction

We present a new method to extract a reliable silhouette from RGB and the corresponding depth image. We assume the availability of a hand pose estimation network from either RGB [2, 28, 37] or depth [8, 38, 41] images, that provides detected keypoints, $k^{(data)}$.

We use GrabCut [25] to segment the RGB image. Instead of manual user interaction, we automatically initialize the foreground and background Gaussian Mixture Models (GMMs) with the help of the depth image and the detected keypoints to obtain a more accurate segmentation of the hand region. Specifically, we preprocess the depth image to crop a bounding box around the hand using the detected keypoints. We then define the region outside the box as background, $\alpha = 0$, and the pixels around the detected keypoints as foreground, $\alpha = 1$.

We show the effect of our initialization on the resulting segmentation in Fig. 3. The correspondences obtained using the distance transform of our silhouette are more accurate than those obtained from the silhouette of the noisy depth image, as shown in Fig. 1.

3.4 Optimization

We use Levenberg-Marquardt to optimize the registration energy in Eq. 5 by linearizing each term written as a sum of squared residuals. To initialize the pose for the current frame, we use the previous frame's optimized pose; for the first frame, we register the model to the keypoints obtained either from the dataset or marked manually.

4 EVALUATION

4.1 Datasets

We evaluate our RGBD registration method on datasets that provide RGB and corresponding depth frames. There are two publicly available datasets: Rendered Hand pose Dataset (RHD) [47] and Stereo hand pose Tracking Benchmark (STB) [44]. The RHD dataset is a synthetic dataset with 41258 images for training and 2728 images for evaluation with a 320×320 pixels resolution. Each sample contains 21 3D keypoint annotations and 33 segmentation masks corresponding to palm, finger segments, human, and background. We combine the segmentation masks on the right hand and use them in our registration.

The STB dataset contains 18000 stereo pairs and depth images captured from a Point Grey Bumblebee2 stereo camera and Intel Real Sense F200 active depth camera, respectively. The dataset contains 21 3D keypoint annotations corresponding to each image.

We also capture a dataset comprising adult and child users using the Intel RealSense D435 stereo depth camera. This dataset contains 1000 RGB and corresponding depth frames. The user wears a wristband to crop out the wrist region from the hand, similar to Tagliasacchi et al. [30].

4.2 Metrics

We quantitatively evaluate the registration accuracy using a dense metric and a sparse metric.

Method	E_{3D} (in mm)
HandTailor [15]	30.22
BiHand [43]	31.92
Our	**21.12**

Table 1: Comparison of our registration framework with state-of-the-art methods on the RHD dataset.

Method	E_{3D} (in mm)
BiHand [43]	20
Our	**14**

Table 2: Comparison of our registration framework with the best performing method on the STB dataset.

Method	AUC_{5-20}
HandTailor [15]	0.632
Ours	**0.775**

Table 3: AUC from 5mm to 20mm on the RHD dataset.

The dense metric is the data-to-model error, E_{3D}. Given an observed depth image $I_d^{(data)}$ and a rendered depth image of the model $I_d^{(model)}$, E_{3D} is defined as

$$E_{3D} = \frac{1}{|I_d^{(data)}|} \sum_{p \in I_d^{(data)}} \|p - \Pi_{I_d^{(model)}}(p)\| \tag{7}$$

where $\Pi_{I_d^{(model)}}$ denotes the closest point correspondence of the observed data point p to the rendered model point cloud.

The sparse metric is the keypoint error, E_k. Given a set of observed keypoints $k^{(data)}$ and model keypoints $k^{(model)}$, the keypoint error is given by

$$E_k = \frac{1}{|k|} \sum_{i \in \{1 \ldots |k|\}} \|k_i^{(data)} - k_i^{(model)}\| \tag{8}$$

where $k_i \in \mathbb{R}^3$ is the i^{th} keypoint. We report keypoint error wherever $k^{(data)}$ is available in the annotated dataset.

4.3 Comparison with state-of-the-art

We compare our registration framework with state-of-the-art RGB 3D hand shape and pose recovery methods using the dense data-to-model metric in Table 1 and Table 2. Our method clearly outperforms state-of-the-art methods on each dataset.

In hand pose estimation a challenging metric is the sparse keypoint error metric. It is plotted as the fraction of test samples that have all predicted keypoints below a given maximum Euclidean distance from the ground truth. We compare the pose estimation accuracy of our method by using the area under curve (AUC) for a threshold range of 5mm to 20mm for the keypoints in Table 3. A higher area under the curve denotes more accurate results.

The curves showing the percentage of correct keypoints (PCK) can be seen in Fig. 4.

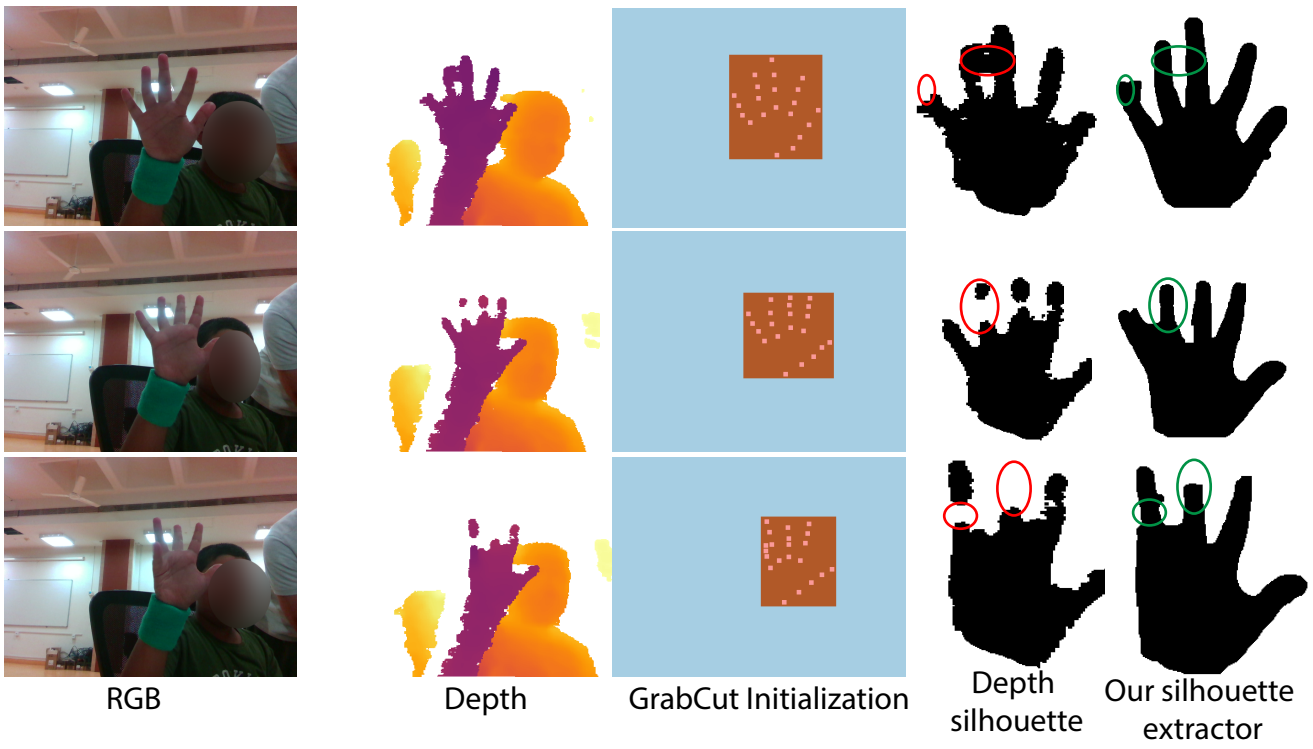

Figure 3: We use RGB and depth image along with the detected keypoints to initialize the GrabCut [25] segmentation algorithm. We define the region outside the bounding box around the keypoints as background (light blue) and the region inside the bounding box as probable foreground (brown). Further, we use the dilated keypoint locations to initialize the foreground (orange). Our silhouette extractor preserves more detail compared to the depth silhouette used in existing methods.

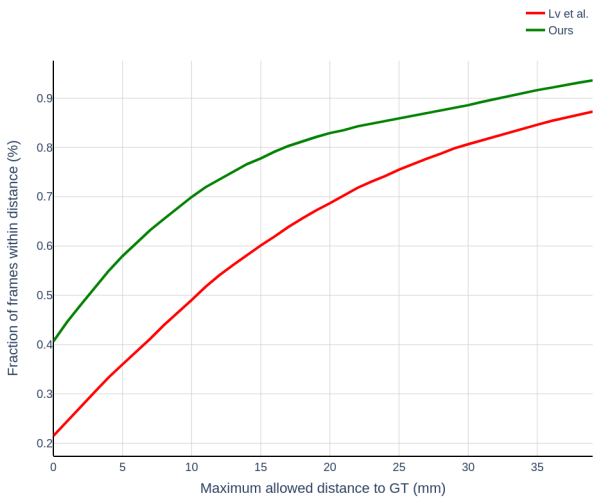

Figure 4: We calculate the percentage of correct frames within a threshold of the keypoint error on the RHD dataset. Our method outperforms state-of-the-art HandTailor method of Lv et al. [15] by a substantial margin.

We qualitatively compare the registered meshes on RHD and STB datasets in Fig. 5 and Fig. 6, respectively.

5 CONCLUSION

We present a new registration method to utilize information from RGB and depth images to achieve more accurate hand tracking than existing methods. To this end, we also propose a novel silhouette extractor that provides a reliable silhouette for the background data term in the optimization. Based on the available input data, RGB or depth or RGBD, our method is flexible enough to adapt to it and achieve state-of-the-art tracking accuracy. We lay the foundation for using the available RGBD sensors on virtual reality headsets for achieving accurate and reliable hand tracking.

Future work. Our method provides accurately registered meshes on RGB images and can serve as a dataset for training dense pose estimation networks for hands similar to dense human body pose estimation networks [7]. Further, our method can be extended to use texture and shading information for registration.

REFERENCES

[1] Anil Armagan, Guillermo Garcia-Hernando, Seungryul Baek, Shreyas Hampali, Mahdi Rad, Zhaohui Zhang, Shipeng Xie, MingXiu Chen, Boshen Zhang, Fu Xiong, et al. 2020. Measuring Generalisation to Unseen Viewpoints, Articulations, Shapes and Objects for 3D Hand Pose Estimation under Hand-Object Interaction. In *ECCV*.

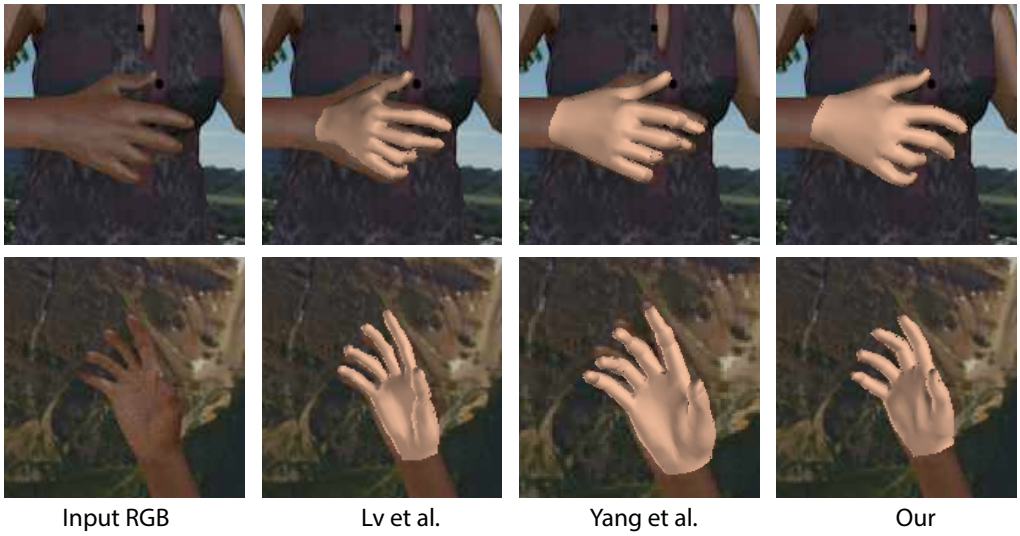

Figure 5: We qualitatively compare our method with Lv et al. [15] and Yang et al. [43] on the RHD dataset. Our method estimates shape and pose more accurately.

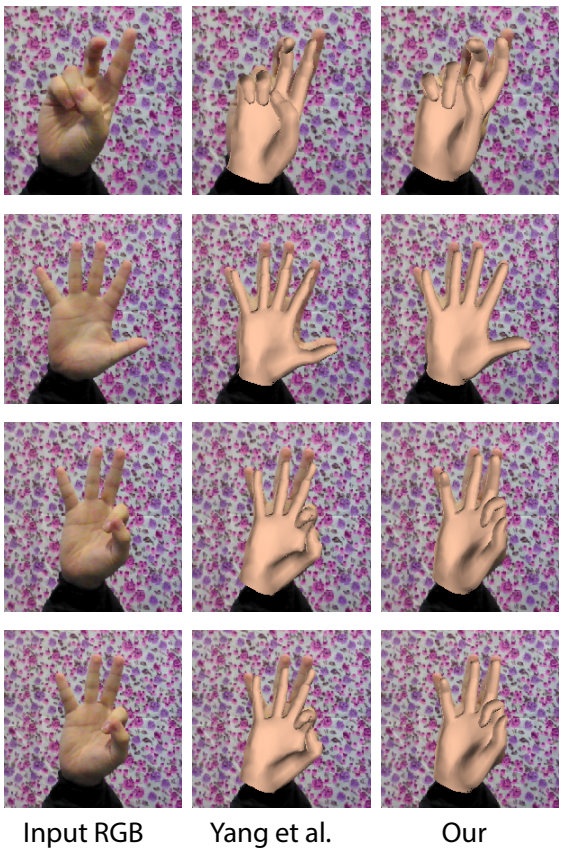

Figure 6: We qualitatively compare our method with Yang et al. [43] on the STB dataset. Our method aligns the finger articulations more accurately.

[2] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. 2019. Pushing the Envelope for RGB-Based Dense 3D Hand Pose Estimation via Neural Rendering. In *CVPR*.

[3] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 2019. 3d hand shape and pose from images in the wild. In *CVPR*. 10843–10852.

[4] Yujin Chen, Zhigang Tu, Di Kang, Linchao Bao, Ying Zhang, Xuefei Zhe, Ruizhi Chen, and Junsong Yuan. 2021. Model-based 3D Hand Reconstruction via Self-Supervised Learning. In *CVPR*.

[5] Martin de La Gorce, David J. Fleet, and Nikos Paragios. 2011. Model-Based 3D Hand Pose Estimation from Monocular Video. *IEEE TPAMI* 33, 9 (2011).

[6] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 2019. 3D Hand Shape and Pose Estimation from a Single RGB Image. In *CVPR*.

[7] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. 2018. Densepose: Dense human pose estimation in the wild. In *CVPR*.

[8] Weiting Huang, Pengfei Ren, Jingyu Wang, Qi Qi, and Haifeng Sun. 2020. AWR: Adaptive Weighting Regression for 3D Hand Pose Estimation. In *AAAI*.

[9] Pratik Kalshetti and Parag Chaudhuri. 2022. Local Scale Adaptation for Augmenting Hand Shape Models. In *ACM SIGGRAPH 2022 Posters*.

[10] Pratik Kalshetti and Parag Chaudhuri. 2022. Local Scale Adaptation to Hand Shape Model for Accurate and Robust Hand Tracking. *Comput. Graph. Forum* 41, 8 (2022). To appear.

[11] Sameh Khamis, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. 2015. Learning an Efficient Model of Hand Shape Variation from Depth Images. In *CVPR*.

[12] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. 2019. Learning to Reconstruct 3D Human Pose and Shape via Model-fitting in the Loop. In *ICCV*.

[13] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M. Bronstein, and Stefanos Zafeiriou. 2020. Weakly-Supervised Mesh-Convolutional Hand Reconstruction in the Wild. In *CVPR*.

[14] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM TOG* 34, 6 (2015), 248:1–248:16.

[15] Jun Lv, Wenqiang Xu, Lixin Yang, Sucheng Qian, Chongzhao Mao, and Cewu Lu. 2021. HandTailor: Towards High-Precision Monocular 3D Hand Recovery. In *BMVC*.

[16] Tanwi Mallick, Partha Pratim Das, and Arun Kumar Majumdar. 2014. Characterizations of Noise in Kinect Depth Images: A Review. *IEEE Sensors Journal* 14, 6 (2014).

[17] Calvin R Maurer, Rensheng Qi, and Vijay Raghavan. 2003. A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. *IEEE TPAMI* 25, 2 (2003), 265–270.

[18] Stan Melax, Leonid Keselman, and Sterling Orsten. 2013. Dynamics Based 3D Skeletal Hand Tracking. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*.

[19] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2018. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *CVPR*.

[20] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. 2011. Efficient model-based 3D tracking of hand articulations using Kinect. In *BMVC*, Vol. 1:2. 3.

[21] Paschalis Panteleris, Iason Oikonomidis, and Antonis Argyros. 2018. Using a single rgb frame for real time 3d hand pose estimation in the wild.

[22] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. 2014. Realtime and Robust Hand Tracking from Depth. In *CVPR*. 1106–1113.

[23] James M. Rehg and Takeo Kanade. 1994. Visual tracking of high DOF articulated structures: An application to human hand tracking. In *ECCV*.

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

[25] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. 2004. " GrabCut" interactive foreground extraction using iterated graph cuts. *ACM TOG* 23, 3 (2004), 309–314.

[26] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel Freedman, Eyal Krupka, Andrew Fitzgibbon, Shahram Izadi, and Pushmeet Kohli. 2015. Accurate, Robust, and Flexible Real-time Hand Tracking. In *CHI*. 3633–3642.

[27] Jingjing Shen, Thomas J. Cashman, Qi Ye, Tim Hutton, Toby Sharp, Federica Bogo, Andrew Fitzgibbon, and Jamie Shotton. 2020. The Phong Surface: Efficient 3D Model Fitting Using Lifted Optimization. In *ECCV*. 687–703.

[28] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. 2017. Hand Keypoint Detection in Single Images using Multiview Bootstrapping. In *CVPR*.

[29] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. 2015. Cascaded hand pose regression. In *CVPR*.

[30] Andrea Tagliasacchi, Matthias Schröder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly. 2015. Robust articulated-icp for real-time hand tracking. In *Comput. Graph. Forum*, Vol. 34:5. 101–114.

[31] David Joseph Tan, Tom Cashman, Jonathan Taylor, Andrew Fitzgibbon, Daniel Tarlow, Sameh Khamis, Shahram Izadi, and Jamie Shotton. 2016. Fits Like a Glove: Rapid and Reliable Hand Shape Personalization. In *CVPR*.

[32] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, Arran Topalian, Erroll Wood, Sameh Khamis, Pushmeet Kohli, Toby Sharp, 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 TOG* 35 (2016).

[33] Jonathan Taylor, Richard Stebbing, Varun Ramakrishna, Cem Keskin, Jamie Shotton, Shahram Izadi, , and Andrew Fitzgibbon. 2014. User-Specific Hand Modeling from Monocular Depth Sequences. In *CVPR*.

[34] Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. 2016. Sphere-meshes for real-time hand modeling and tracking. *ACM TOG* 35, 6 (2016), 1–11.

[35] Anastasia Tkach, Andrea Tagliasacchi, Edoardo Remelli, Mark Pauly, and Andrew Fitzgibbon. 2017. Online generative model personalization for hand tracking. *ACM TOG* 36, 6 (2017), 1–11.

[36] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. 2014. Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks. *ACM TOG* 33 (2014).

[37] Andrey Vakunov, Chuo-Ling Chang, Fan Zhang, George Sung, Matthias Grundmann, and Valentin Bazarevsky. 2020. MediaPipe Hands: On-device Real-time Hand Tracking. In *Workshop on Computer Vision for AR/VR*.

[38] C. Wan, T. Probst, L. Gool, and A. Yao. 2018. Dense 3D Regression for Hand Pose Estimation. In *CVPR*.

[39] Jiayi Wang, Franziska Mueller, Florian Bernard, Suzanne Sorli, Oleksandr Sotnychenko, Neng Qian, Miguel A. Otaduy, Dan Casas, and Christian Theobalt. 2020. RGB2Hands: Real-Time Tracking of 3D Hand Interactions from Monocular RGB Video. *ACM TOG* 39, 6 (2020).

[40] Robert Y. Wang and Jovan Popović. 2009. Real-Time Hand-Tracking with a Color Glove. *ACM TOG* 28, 3 (2009).

[41] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Zhou Tianyi, and Junsong Yuan. 2019. A2J: Anchor-to-Joint Regression Network for 3D Articulated Pose Estimation from a Single Depth Image. In *ICCV*.

[42] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. 2020. GHUM & GHUML: Generative 3D Human Shape and Articulated Pose Models. In *CVPR*. 6184–6193.

[43] Lixin Yang, Jiasen Li, Wenqiang Xu, Yiqun Diao, and Cewu Lu. 2020. BiHand: Recovering Hand Mesh with Multi-stage Bisected Hourglass Networks. In *BMVC*.

[44] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. 2017. A hand pose tracking benchmark from stereo matching. In *ICIP*.

[45] Xiong Zhang, Qiang Li, Hong Mo, Wenbo Zhang, and Wen Zheng. 2019. End-to-end Hand Mesh Recovery from a Monocular RGB Image. In *ICCV*.

[46] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. 2020. Monocular Real-Time Hand Shape and Motion Capture Using Multi-Modal Data. In *CVPR*.

[47] Christian Zimmermann and Thomas Brox. 2017. Learning to estimate 3d hand pose from single rgb images. In *ICCV*.