2024 18th International Conference on Automatic Face and Gesture Recognition (FG)

Two Hands Are Better Than One: Resolving Hand to Hand Intersections via Occupancy Networks

Maksym Ivashechkin, Oscar Mendez, Richard Bowden
CVSSP, University of Surrey, Guildford, United Kingdom
{m.ivashechkin, o.mendez, r.bowden}@surrey.ac.uk

***Abstract*— 3D hand pose estimation from images has seen considerable interest from the literature, with new methods improving overall 3D accuracy. One current challenge is to address hand-to-hand interaction where self-occlusions and finger articulation pose a significant problem to estimation. Little work has applied physical constraints that minimize the hand intersections that occur as a result of noisy estimation. This work addresses the intersection of hands by exploiting an occupancy network that represents the hand's volume as a continuous manifold. This allows us to model the probability distribution of points being inside a hand. We designed an intersection loss function to minimize the likelihood of hand-to-point intersections. Moreover, we propose a new hand mesh parameterization that is superior to the commonly used MANO model in many respects including lower mesh complexity, underlying 3D skeleton extraction, watertightness, etc. On the benchmark INTERHAND2.6M dataset, the models trained using our intersection loss achieve better results than the *state-of-the-art* by significantly decreasing the number of hand intersections while lowering the mean per-joint positional error. Additionally, we demonstrate superior performance for 3D hand uplift on RE:INTERHAND and SMILE datasets and show reduced hand-to-hand intersections for complex domains such as sign-language pose estimation.**

I. INTRODUCTION

Hand pose estimation in 3D has seen lots of interest across a broad range of applications. For instance, 3D hand estimation can be used for sign language recognition, human-computer interaction, animation, virtual/augmented reality, etc. 3D hand estimation is an extremely challenging problem due to factors including motion blur, self-occlusion of the fingers, and interaction with body and face. The literature contains many works on single 3D hand pose estimation (*e.g.,* [6, 29, 30, 34, 37, 38]) that tackle this problem by either direct hand prediction from an image or by decomposing estimation into an image-to-2D, then 2D-to-3D uplift.

The interaction of two hands in 3D space presents even greater complexity due to mutual occlusions and intersections. A naive approach is to apply a single-hand model twice to an image but this leads to poor estimation that lacks realism, especially in subtle cases such as interlocking fingers. Furthermore, interacting hands may provide additional information on mutual hand position, and this can prove beneficial in decreasing the search space of solutions.

A. Related Work

Early works on 3D pose estimation for interacting hands tried to solve the problem using classic optimization approaches. Ballan *et al*. [3] proposed tackling the problem by exploiting salient points and formulating a differentiable objective function that incorporates edges, optical flow, and collisions extracted from an image. Oikonomidis *et al*. [23] tracked interacting hands by using a stochastic optimization method with the objective of finding the two-hand configuration that best explains observations from an RGB-D sensor.

With the rise of deep learning, hand interaction was approached by Taylor *et al*. [31] who parameterized hands and exploited an articulated signed distance function to fit their model to multiview depth data. Mueller *et al*. [22] proposed a method that uses a single depth camera along with an angular hand parameterization. A multiview setup for interacting hand estimation from RGB images was employed by Smith *et al*. [28] where a physically based deformable model constrains a vision-based tracking algorithm to tackle self-occlusions and self-intersections.

More recent methods apply hand estimation to a single RGB image exploiting various techniques including image segmentation, mesh rendering, relative depth regression, etc. A common approach is to parameterize the hand using the MANO [25] model, which represents the hand skeleton via joint orientations, and the hand's volume is parameterized by a shape vector. Moreover, the MANO library provides an efficient model that converts angular and shape hand parameterization into a 3D mesh surface. MANO delivers many benefits that help to enforce a realistic hand, work with volumetric hand shapes, and render a mesh onto an image. But as will be seen, it also has shortcomings.

RGB2Hands [32] by Wang *et al*. tackles 3D pose estimation and tracking of interacting hands from a monocular input by leveraging a segmentation mask, 2D detection, and dense matching to regress MANO hand parameters. Zhang *et al*. [4] address the hand interaction problem by applying a hand pose-aware attention module to retrieve features corresponding to each hand.

The decomposition of interacting hands was tackled by Meng *et al*. [18] who utilize de-occlusion and removal modules to recover the appearance content of the occluded part of one hand and remove the distracting hand. On the contrary, Fan *et al*. [8] process an image using a per-pixel semantic part segmentation mask and a visual feature volume to leverage the per-pixel probabilities directly during pose estimation without decoupling the segmentation stage or individual hands in the pipeline. Rong *et al*. [26] introduce a two-stage approach where at the first stage the CNN module makes a coarse prediction of interacting hands, and the

979-8-3503-9494-8/24/$31.00

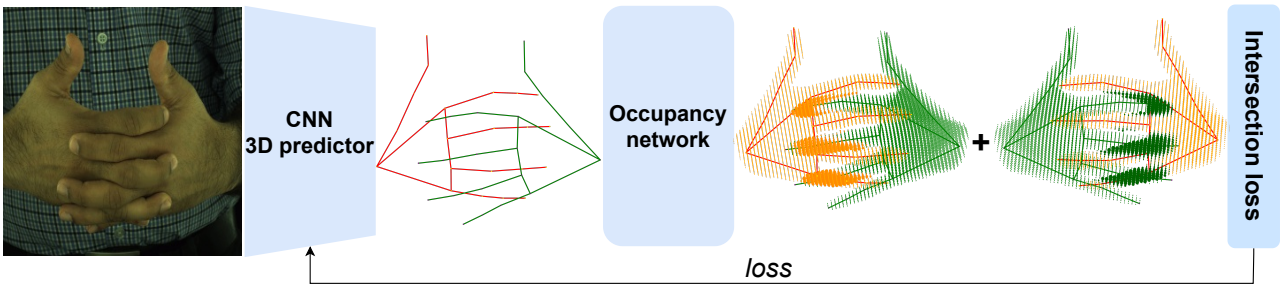


Fig. 1: The figure demonstrates the pipeline for accurate 3D interacting hands estimation. The input image is processed via a CNN model (*e.g.,* ResNet [10], MediaPipe, etc.) that enables the extraction of image features or 2D keypoints necessary to uplift hands into 3D. Note our approach is invariant to the backbone used. Afterward, a pre-trained occupancy network with frozen weights is conditioned via the right hand, and the intersections are tested with the left hand, and optionally *vice-versa* with the hands flipped as illustrated. The red and green edges highlight the right and left hands, respectively. Light green and light orange points visualize the density of hands determined by the occupancy network and their size emphasizes the likelihood of intersection. Since both hands are fully differentiable with respect to the occupancy and CNN networks, it provides efficient backpropagation of the intersection loss. The source image is taken from the INTERHAND2.6M dataset.

second step progressively ameliorates the hands' collisions through a series of factorized refinements.

INTERHAND2.6M [21] is a benchmark dataset for single and interacting hand pose estimation released by Moon *et al.*. It was released together with a baseline approach that exploits a ResNet image model to regress 2.5D keypoints (image points with relative depth) that, via back-projection, are uplifted to 3D space. This dataset has become widely used for hand pose evaluation and comparison. The most recent *state-of-the-art* methods achieve excellent performance, setting a high bar for interacting hand estimation on the INTERHAND2.6M dataset. One of those methods is IntagHand [15] by Li *et al.* which uses a graph convolutional neural network to solve occlusions and interactions. This is achieved by adding attention-based modules to implicitly obtain vertex-to-image alignment that encode the coherence of interacting hands. Jiang *et al.* present the A2J-Transformer [12] that extends the A2J framework [33], which uses anchor points to capture global-local spatial context. The A2J-Transformer includes several advantages such as leveraging self-attention across local anchor points for spatial context awareness and utilizing anchor points as learnable queries with adaptive feature learning in 3D. Yu *et al.* [35] introduce the ACR framework that explicitly mitigates interdependencies between hands by leveraging center and part-based attention for feature extraction and learns the corresponding cross-hand prior.

A similar work that tackles hand-to-object interactions is HALO [13] by Karunratanakul *et al.* who present an occupancy network for continuous hand representation. HALO employs a 3D sparse point cloud as input to the network to predict hand surface and utilizes hand-to-object occupancies to minimize the number of intersections. However, it did not explore resolving hand-to-hand intersections.

B. Motivation & Contributions

Our primary focus is to improve 3D hand pose estimation from a single image by leveraging the physical constraints of hand-to-hand interaction. Importantly, our framework can be applied to any *state-of-the-art* 3D hand estimation approach to improve performance. We demonstrate a reduction in hand intersections for 5 *state-of-the-art* approaches. Through extensive evaluation on INTERHAND2.6M [21], RE:INTERHAND [20], and SMILE [7] sign-language datasets, we show significant reduction in the number of intersections and 3D per-joint error.

To do this we employ an occupancy network to exclude mutual intersections. The occupancy model provides a continuous volumetric representation of the hand conditioned on a sparse skeletal model. We propose a new hand mesh parameterization that exploits a kinematic hand model which is more robust than the MANO framework. But this is only used to train the occupancy network.

By leveraging the hand occupancy, we can resolve volumetric intersections without the use of a mesh. We structure our approach around an end-to-end differentiable pipeline and employ an intersection loss that, in combination with a CNN, allows the reduction of hand intersections which improves 3D estimation. In contrast to HALO, which minimizes hand-to-object intersections, we model intersections for two dynamic and interacting hands. This is a more significant challenge than hand-to-object, as it involves the simultaneous prediction of both articulated hands, while in hand-to-object, the object usually remains constant.

II. METHODOLOGY

We propose an intersection loss for 3D interacting hand estimation that enforces physical constraints and visual realism. The pipeline of our approach is outlined in Fig. 1. The method relies on a CNN-based 3D pose estimator to lift initial 2D hand skeletons to 3D. By exploiting a pre-trained occupancy network [19] conditioned with a 3D skeleton, we apply an unsupervised intersection loss on both predicted hands. The occupancy network is pre-trained on hand meshes extracted with a custom hand mesh parameterization.

A. Hand Mesh Representation

The MANO library is extensively used in the literature as a hand mesh representation. It provides essential attributes such as differentiability, angular and shape mesh parameterization, facilitates retrieval of a 3D skeleton via linear blend skinning, and a heightened level of visual realism. Unfortunately, despite all the advantages, it has certain limitations. Firstly, it has a reliance on a pre-trained statistical hand model computed with principal component analysis. Secondly, the 3D skeleton can only be obtained after mesh regression. Finally, the generated mesh is not watertight.

The primary challenge lies in the complexity of fitting MANO meshes to pre-existing 3D skeletons, *i.e.,* when 3D hand skeletons are known, but the goal is to obtain the volumetric shape. This optimization could be done over the shape and angular MANO parameters. However, the pre-computed MANO weights may not match the target skeleton distribution, which can result in strange mesh deformations.

Therefore, we present a new hand-mesh parameterization that is more practical for use with hand-to-hand interaction. The core of the proposed mesh model is a 3D hand skeleton generated with forward kinematics combining joint angles and bone lengths. We add vertices along the fingers that span the envelope of the hand, providing a volumetric shape. Through an awareness of the arrangement of nodes, we are able to triangulate the hand's surface and construct a watertight mesh (*i.e.,* no holes) using triangular faces. The full process of adding new nodes is fully automatic, and more vertices can be added to give a more complex hand shape. The proposed custom mesh has multiple advantages over the MANO mesh:

1) It is watertight.
2) It has less than half the vertices (307 vs. 778).
3) The volumetric shape is added on top of the skeleton (as opposed to MANO that generates the mesh and then skeleton), which enables lazy evaluation if only a skeleton is needed.
4) No dependency on the pre-trained weights.

The mesh watertightness plays a crucial role in training the occupancy network since it allows us to determine whether a 3D point is inside the mesh (using ray casting [27]). Both MANO and the proposed meshes are differentiable, and the proposed model has more parameters to differentiate, which gives more user control over subtle details.

To provide a good trade-off between visual quality and complexity, we designed a refined version of the proposed mesh model that has more points (699) and triangles, for the purpose of visualization. Fig. 2 demonstrates a visual comparison of two meshes. MANO versus the proposed custom mesh comparison is illustrated in Fig. 4.

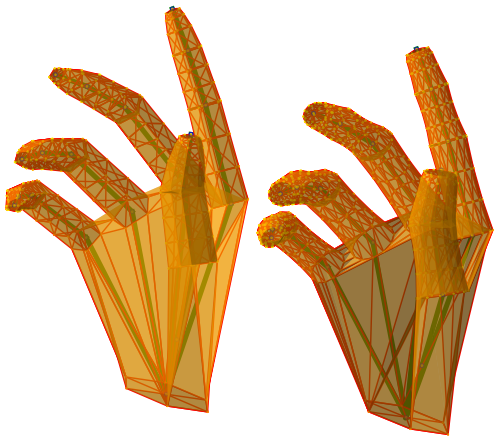


Fig. 2: Comparison of plain (left) and complex (right) watertight hand meshes generated with our parameterized mesh model. The green pose (underneath the orange envelope) is found using forward kinematics (FK) that combine angles and bone length. The yellow points are also obtained via FK with pre-determined offsets from the underlying skeleton, the red triangles span the entire hand surface.

B. Occupancy Network

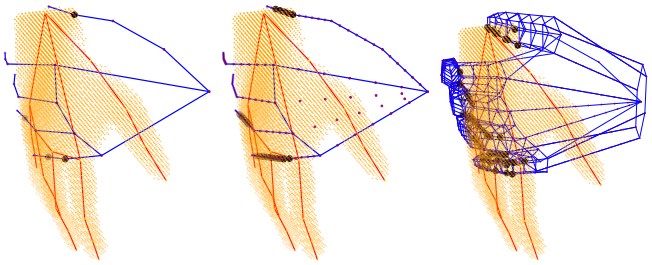
The occupancy network serves as the primary component in the proposed pipeline. Its purpose is to model the volumetric shape of the hands and resolve intersections at the physical level, since it determines whether a point occupies the same space as a hand. The benefit of utilizing an occupancy network is to obtain a continuous manifold of the hand that can be used in a lazy evaluation. To obtain such a representation, the occupancy network is conditioned with a feature vector that corresponds to the desired shape.

Mathematically, let $\mathcal{O} : \mathbb{R}^3 \times \mathbb{F} \rightarrow [0, 1]$ be an occupancy network that given a point in 3D space $\mathbf{x} \in \mathbb{R}^3$, and feature vector $\mathbf{f} \in \mathbb{F}$, returns a probability $p \in [0, 1]$ of the point $\mathbf{x}$ being occupied in the target space conditioned by vector $\mathbf{f}$.

Implementation-wise, the occupancy network consists of an encoder and a decoder. The encoder processes 3D observations to produce a feature vector and acquires knowledge about the mean and standard deviation of the Gaussian distribution within the latent space. At inference, the model samples the learned latent space, and the decoder transforms the latent space into occupancy logits, which can be converted to probabilities using the Bernoulli two-class model.

C. Intersection Loss

The pipeline in Fig. 1 shows hand intersections as dense point clouds estimated using the occupancy network. However, this representation would be inefficient. In practice, a different technique is used. For interacting hands, let $\mathbf{X} \in \mathbb{R}^{3\times N}$ be a matrix of 3D hand joints stacked in a column, where the most common hand representation defines $N = 21$ points. We distinguish joints of right and left hands by the corresponding underscore letters, *i.e.,* $\mathbf{X}_R$ and $\mathbf{X}_L$. Without loss of generality, the point set of the right hand is utilized to condition the occupancy network, and the left hand's set is employed to check point-wise intersections. For neural networks, it is essential to work in a high-dimensional space as it enables them to capture subtle data patterns, learn features faster, and thus have better convergence. Therefore, we exploit an encoder $\mathcal{F} : \mathbb{R}^{3\times N} \rightarrow \mathbb{F}$ that converts the point set $\mathbf{X}_R$ to a feature representation.



(a) sparse (b) dense (c) surface

Fig. 3: Comparison of applying different point sets to check intersections: a) sparse skeleton, b) skeleton with additional points along the edges, c) skeleton with mesh surface points. The yellow points show the density of the right hand, and the black points highlight intersections.

The goal of the intersection loss is to minimize the probabilities of the left hand's points when the occupancy network is conditioned on the right hand. The corresponding objective is the following:

$$\sum_{i=1}^{N}\Big[\mathcal{O}\big(\mathbf{X}_{Li},\mathcal{F}(\mathbf{X}_R)\big)^2+\alpha\mathcal{O}\big(\bar{\mathbf{X}}_{Ri},\mathcal{F}(\bar{\mathbf{X}}_L)\big)^2\Big], \quad (1)$$

where $\mathbf{X}_{\{L,R\}_i}$ denotes indexing the i-th point of the matrix. The parameter $\alpha \in \{0,1\}$ is a constant that determines whether flipped hands are tested for intersection. Correspondingly, $\bar{\mathbf{X}}_{Ri}$ and $\bar{\mathbf{X}}_{Li}$ denote a point set flipped along the x-axis (*i.e.,* multiplied by -1). The square loss function is particularly effective at emphasizing points with a higher likelihood of intersection, as it progressively diminishes the impact of lower values within the range of zero to one. Along with the squared loss, a truncated loss can also be employed, *i.e.,* probabilities >50%. The motivation being to focus on points more likely to intersect. However, we empirically found that a non-truncated loss resolves more intersections for the same hyperparameter settings (*e.g.,* learning rate).

In practice, the batched calculations are performed within the Pytorch [24] library, which makes the intersection loss extremely efficient, as only 21 points need to be tested. Nevertheless, 21 joints do not span the whole hand nor its volume, and gaps between points result in a slightly less accurate intersection test. This can be mitigated by adding virtual points along the hand edges, which introduces an additional 100 points, *e.g.,* the hand skeleton contains 20 edges where 5 new points are added to each edge. Alternatively, the whole hand mesh surface can be used for intersection verification. However, this slows model training as the total number of points to test rises significantly. In total, there are six options for checking hand intersections, *i.e.,* testing sparse, dense, and mesh surface vertices for a single (left) hand, or additionally flipping both hands to repeat the test. In the experiments, we provide results for each of these options. The comparison of skeletons used for testing is demonstrated in Fig. 3, where larger point sets reveal areas where intersections could happen.

III. EXPERIMENTS

To evaluate the effect of the intersection loss and occupancy network on the pose estimation of interacting 3D hands, we perform several sets of experiments. Firstly, on the INTERHAND2.6M dataset, we compare against *state-of-the-art* methods. Secondly, on RE:INTERHAND dataset of interacting hands. Finally, we train the hand pose estimation model on the SMILE dataset [7], and evaluate the accuracy on "in the wild" videos where the ground truth is not known.

The standard metric for 3D accuracy is the mean per-joint position error (MJPJE). However, to measure the number of intersections from the occupancy network, we additionally exploit a ray-casting algorithm to precisely check whether points are inside a mesh to confirm the results.

As mentioned previously, the proposed mesh is watertight, which enables us to test if an arbitrary point is inside the mesh. We create a mesh grid of uniformly distributed 3D points that span the size of meshes. In total, 125 thousand points are tested using the ray-casting algorithm to find mesh occupancies and mask points inside meshes for each hand.

A. Training the Occupancy Network

The occupancy network is conditioned with a single-hand skeleton that has 21 joints. Surprisingly, conditioning the occupancy network with a mesh surface instead of a sparse skeleton does not significantly increase the accuracy of the model, however, it does make training times much longer.

Random samples of points (8192 per hand) and corresponding occupancy masks are used to train the occupancy network. During validation, all 125 thousand points are employed to compute the intersection over union rate of the ground truth and predicted occupancies, which is the primary metric for the occupancy network evaluation. During training, additional augmentations on the input and sampled points are employed. We randomly rotate input skeletons and corresponding meshes by ± 180 degrees, and random Gaussian noise is added to the sampled mesh points.

B. Results on INTERHAND2.6M dataset

The INTERHAND2.6M dataset [21] contains 2.6 million images of single and interacting hands, with ground truth poses found from the triangulation of 80-140 views. The 2D hand detections were obtained by either manual human labeling or an automatic annotation tool. The accuracy of INTERHAND2.6M is notably high, as the reconstruction process leveraged a large number of different camera views.

The dataset also contains MANO meshes that were fitted to the triangulated skeletons with about a 5 mm error[1]. By exploiting the custom mesh parameterization and an inverse kinematics solver [11] we fit our new mesh with less than 0.5 mm error. Having significantly more accurate meshes is needed for the occupancy network and is vital for the hand intersection test, where a small finger's shift is crucial.

Where pre-trained models were available, we applied the intersection loss to the top performing methods with

[1] https://mks0601.github.io/InterHand2.6M/

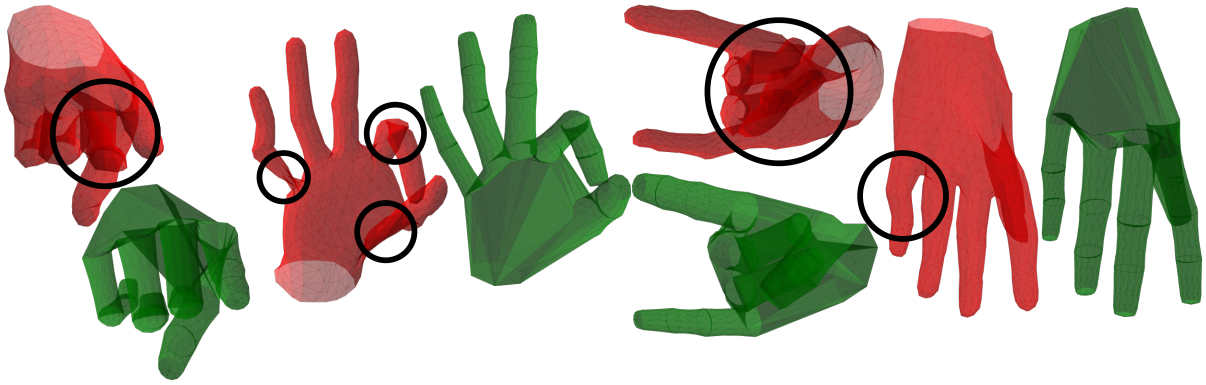


Fig. 4: This figure shows a comparison of the failed MANO (red) and our (green) meshes fitted to the INTERHAND2.6 3D hand joints. The black circles on the MANO meshes highlight specific problems of the MANO hand's appearance, such as twisted fingers, unrealistic shape, incorrect finger orientation, etc.

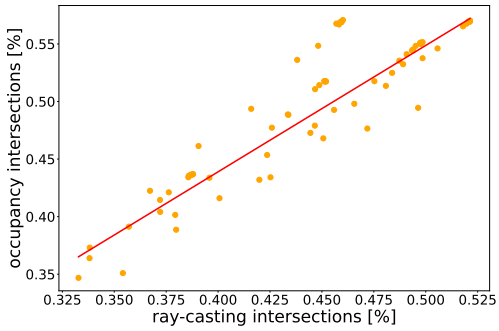


Fig. 5: Correlation trend of per-point intersection probability found via occupancy network and ray-casting algorithm.

respect to the lowest MPJPE error on the INTERHAND2.6M dataset. Since not all of them provide the training code, we trained a simple linear network (MLP) that takes as input 3D interacting hand skeletons from a *state-of-the-art* model and generates the same 3D skeletons. The accuracy of this network matches the accuracy of the *state-of-the-art* methods. We then took the subset of the data containing two interacting hands, and fine-tuned the 3D-to-3D model with the intersection loss.

The main parameter that influences both MPJPE and intersection accuracy is the weight of the intersection loss function. If the weight is too high, the model tries to push hands away from each other to prevent any intersections, which increases the 3D error. Conversely, a small weight does not have any impact on the model. We found that the optimal weight that balances lower MPJPE while minimizing the intersections is within the range 10^{-5} to 10^{-8} depending on the size of the points tested (*e.g.,* sparse or mesh).

In our experiments, the fine-tuned models trained with the intersection loss consistently have a lower number of intersections and improved MPJPE over *state-of-the-art* methods. When comparing to the *state-of-the-art*, we specifically selected models that achieve a similar mean per-joint position error while minimizing the overall number of intersections.

The number of intersections is found in two ways. First, using the occupancy network with skeleton conditioning.

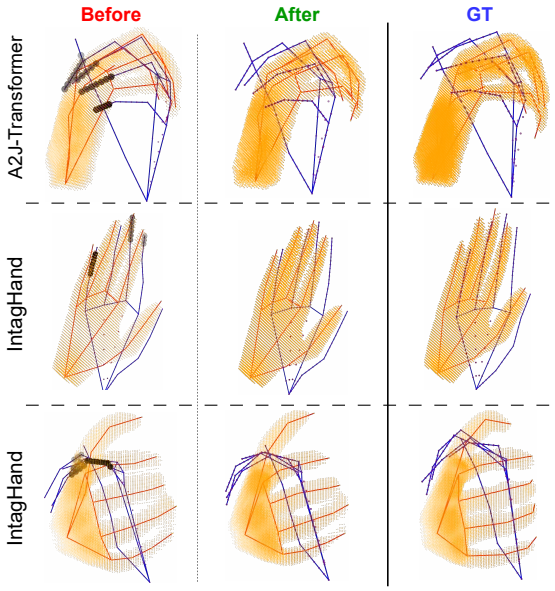


Fig. 6: Comparison of 3D hand poses estimated by *state-of-the-art* methods (IntagHand [15] and A2J-Transformer [12]) before (left) and after (middle) training with the hand intersection loss on INTERHAND2.6M. The ground truth hands are shown in the right column. Red skeletons show right hands, blue skeletons left hands. Orange points around the right hand highlight volumetric hand density found via the occupancy network. Large black points show point-to-hand intersections. Note: there are no intersections in either the model trained with intersection loss or the ground truth.

However, to achieve a fair comparison, with less reliance on the occupancy network, we find intersections by fitting custom meshes to skeletons returned by the *state-of-the-art*. The skeletons of new models are trained with the intersection loss; the same way we converted the INTERHAND2.6M 3D hands to meshes. The mean fitting error is lower than 0.4 mm, which provides highly precise meshes. We have conservatively chosen a thinner volumetric shape for meshes, particularly the finger thickness, to ensure a high level of confidence in intersection detection.

TABLE I: Comparison of *state-of-the-art* models and their updated versions ("+Ours") trained with intersection loss in terms of a number of intersections. The second column indicates the amount of intersections found via the ray-casting algorithm ("Ray-C."), and the last column shows the quantity of intersections determined by the occupancy network ("Occ."). The "%" row reports a percentage decrease in the amount of intersections for the proposed models. The total number of tested hand pairs is around 220000. CNN stands for our baseline model.

Method	Ray-C.	Occ.	Method	Ray-C.	Occ.
Xiong [33]	376111	20771	Moon [21]	418079	22233
Xiong+Ours	**306478**	**17835**	Moon+Ours	**295261**	**16269**
%	18.51	14.13	%	29.38	26.82
Li [33]	583062	32253	Zhang [36]	592685	32240
Li+Ours	**372534**	**20286**	Zhang+Ours	**336591**	**18843**
%	36.11	37.10	%	43.21	41.55
Hampali [9]	452408	25765	CNN	490212	25904
Hampali+Ours	**325803**	**19827**	CNN+Ours	**394345**	**22184**
%	27.98	23.05	%	19.56	14.36

Table I reports the number of intersections, found via the ray-casting algorithm and the occupancy network, for both *state-of-the-art* and versions with the proposed intersection loss. We found that the best intersection loss used for fine-tuning all models employs a sparse set of points of a single hand with a non-truncated kernel. In the experiments, these settings resulted in the best reduction of intersections and MPJPE. The proposed CNN model provides excellent MPJPE accuracy having an 11.4 mm error compared to the A2J [33] method (11.2 mm), or the third place Keypoint Transformer [9] (14.7 mm). In Table I, the percentage decrease in the number of intersections found via the ray-casting algorithm correlates with intersections detected with the occupancy network, see Fig. 5. Therefore, it additionally confirms the accuracy of the occupancy model.

Models including the intersection loss achieve significantly fewer hand-to-hand intersections than *state-of-the-art*. For example, Fig. 3 shows a comparison of interacting hands of IntagHand and A2J-Transformer models against their versions trained with the intersection loss.

Removing all intersections is challenging for several reasons: Firstly, a higher weight for the intersection loss may lead to significant deterioration in 3D accuracy of hand estimator model. Secondly, there is not enough data for hands in close interaction where possible intersections could occur. Such situations correspond to approximately 17% percent of the INTERHAND2.6M data. Thirdly, error propagation from the occupancy network, which in experiments on the validation set has around 80% of intersection over the union (IoU) accuracy. Finally, imprecision of the ground truth data that also contains intersections, as the 3D ground skeletons are often extracted with triangulation algorithms.

For the same reasons, lowering the mean per-joint position error is difficult. Since we use an unsupervised intersection loss, there is no guidance to the model on how it should fix the intersections, *i.e.,* the intersection loss function only tells the model where it cannot position the hand joints. During training, the best model was selected using the minimum error on the validation set in terms of mean per-joint accuracy (not the minimum amount of intersections).

C. Results on the RE:INTERHAND dataset

The RE:INTERHAND dataset [20] of Moon *et al*. provides a large collection of synthetically generated realistic images of interacting hands. In our experiments, we used egocentric viewpoints where 8 captures are reserved for training (402200 frames) and the remaining 2 captures for testing (90380 frames). Since the dataset is recent, we have not found any released competitor models. Therefore, we used a CNN model to predict the two 3D hands directly from the image. Then, we fine-tuned the corresponding model with intersection loss (testing dense points of a single hand). A 10^{-4} weight on intersection loss decreased the error by 0.15% and intersections (found via ray-casting) by 27% compared to the model trained without intersection loss. Similarly, a 10^{-5} weight decreased the error by 3.1% and the intersections by 6.9%, and a 10^{-6} weight by 3.8% and 3.6% (respectively). Therefore, experiments on this dataset confirm that the weight of the intersection loss provides a good trade-off that minimizes error and hand intersections.

D. In The Wild Evaluation

It is important to evaluate the hand pose estimator and its performance "in the wild". This means randomly selected videos where the quality of interacting hands is crucial for understanding, *e.g.,* sign language. However, such videos do not have the corresponding 3D ground truth reconstruction. Therefore, the evaluation metrics consist of qualitative results as well as the number of intersections determined via the ray-casting algorithm, as shown on INTERHAND2.6M.

We designed a multi-layer perception (MLP) to uplift right and left 3D hands with hand-to-hand offset from 2D keypoint detections (obtained with MediaPipe [17]). The MLP has three submodules that separately predict joint angles of the hands, the bone lengths for both hands, and the relative offset between the two hands, as in [11]. The total size of the network is 9.5 million parameters. Without loss of generality, the MLP predicts two right hands and flips the left hand in the x-axis. The prediction of angles enables it to directly find the corresponding meshes with the proposed mesh parameterization.

For training the MLP, we used a version of the SMILE [7] dataset that has the 3D hands triangulated from three calibrated cameras. This makes the ground truth reconstruction significantly less reliable (in terms of interacting hands accuracy) compared to the INTERHAND2.6M dataset. However, it provides a greater challenge to remove intersections and lower the MPJPE error. We opted for the SMILE dataset for several key reasons. Firstly, it enables us to show the versatility of the intersection loss when applied to an alternative dataset. Secondly, we can evaluate the effectiveness of the intersection loss in scenarios where ground truth accuracy for interacting hands is less than ideal. Lastly, as a sign language dataset, it provides a significantly larger and diverse set of real-world interacting hand data, enhancing the depth and practicality of the evaluation.

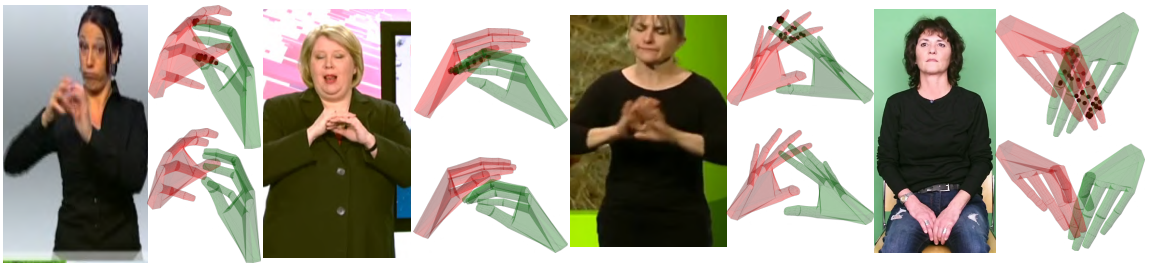


Fig. 7: Comparison of 3D hand estimation models after applying the intersections loss (bottom) and without (top) on images taken from various sign language datasets not used in training (left to right): RWTH-Phoenix Weather 2014 [14], German sign language DSGS [16], BBC-Oxford British Sign Language [1, 2], and SMILE Swiss sign language dataset [7]. The black points highlight point-wise intersections. The model trained with intersection loss produced 3D hands completely free of intersections. Consequently, the bottom meshes are the refined version of the custom mesh parameterization.

We first train a baseline network that is supervised with the ground truth 3D skeletons. Following the training of the baseline model, it was fine-tuned using interacting hands, and the cloned models were fine-tuned with different types of the intersection loss, *i.e.,* testing single/both hands or a sparse/dense/mesh surface point sets. All models were given the same number of epochs to converge.

TABLE II: Percentage decrease in intersections (found with an occupancy network) for the baseline model vs models trained with intersection loss. The rows show whether single or both hands were employed. The columns indicate the density of points. The results correspond to the test set (in total 734120 pairs) of the SMILE dataset.

Number of intersections decrease in %			
	sparse	dense	mesh
single	8.189	16.346	9.444
both	5.361	**18.727**	7.167

TABLE III: Percentage decrease in the number of intersections found via the ray-casting algorithm comparing the baseline network and models trained with different types of intersection loss on four sign-language datasets. The rows indicate whether single or both hands were tested, and the columns show the density of the point sets.

	BOBSL [1, 2]				DSGS [16]		
	S	D	M		S	D	M
single	3.94	16.30	8.88	single	4.04	11.47	7.46
both	5.18	**16.59**	9.67	both	3.97	**14.97**	5.41
	Phoenix [14]				SMILE [7]		
	S	D	M		S	D	M
single	2.53	3.59	2.76	single	8.78	20.16	12.74
both	0.63	**7.87**	2.06	both	10.84	**31.20**	15.52
	Average over datasets						
	S		D		M		
single	4.82		12.88		7.96		
both	5.15		**17.66**		8.16		

Table II compares the number of hand intersections (found via occupancy network) for the baseline and models trained with intersection loss. Similarly to the experiments on the InterHand2.6M dataset, we selected models that have around the same error as the baseline, but the least number of intersections. Table III shows the number of intersections found with the ray-casting algorithm on four "in the wild" sign-language datasets. In each dataset, we selected 250000 random interacting hand pairs, except for the Phoenix dataset [14], which has around 120000 pairs; and is the only dataset where models with the intersection loss do not fully outperform the baseline model. The smaller quantity of test data could be the reason for this. From Table II and III the results suggest that models with the intersection loss resolve a substantial amount of intersections. Moreover, the decrease in the number of intersections found via the occupancy network and ray-casting algorithm (averaged over datasets) correlates, similar to the results on the InterHand2.6M dataset, which again confirms the accuracy of the occupancy network. In the majority of cases, the model trained using the dense points intersection loss yields the fewest intersections. Testing mesh surface vertices results in a greater reduction of intersections compared to a sparse points set. Nevertheless, with a significantly larger distribution of points in space, the model's performance unexpectedly declines. Testing both hands shows (on average) a further reduction in intersections compared to a single hand. The qualitative evaluation of the baseline and models with intersection loss is shown in Fig. 7, where hands returned by the network with the intersection loss are more physically plausible.

TABLE IV: Comparison of training times across models with varying types of intersection loss. Rows specify the test type and rows the number of hands tested. The model without intersection loss is in the bottom row.

Iterations per second			
	sparse	dense	mesh
single	13.23	11.72	6.62
both	9.88	8.22	4.19
no intersection loss		20.85	

E. Time complexity

To investigate the impact of testing different point sets in the intersection loss against the model training time, *e.g.,* sparse/dense/surface or the number of hands, we measured the amount of iterations executed per second (*i.e.,* model forward pass, loss computation, backpropagation, and parameter update) for a batch size of 256 interacting hands. Table IV shows the time comparison for each setting, where testing

both hands in the intersection loss accounts for around 50% of computational time. The computational complexity of the intersection loss increases with the number of points tested. The fastest option (sparse, single hand) is more than three times faster than the slowest (surface points with two hands), and around 36% slower than training a model without an intersection loss. Note, the values in table IV are obtained by running occupancy on all interacting hands. However, the additional speed-up is gained by filtering the hand pairs with a 3D bounding box intersection test. The measurements were done on the 11th Gen Intel Core i9 Ubuntu machine with an NVIDIA GeForce RTX 3090 GPU.

Considering that the 3D ground truth of the SMILE dataset is not very accurate, it is noteworthy that the models with intersection loss have managed to decrease intersections while lowering or maintaining the MPJPE accuracy. Testing dense points of a single hand provides a good trade-off maintaining training speed while ensuring minimal intersections.

F. Noise Influence on Intersection Loss

The intersection can provide additional cues to the 3D uplift model by providing information where hand joints cannot be located to avoid hand intersection. To confirm this hypothesis, we designed an experiment, where during training, the ground truth 3D points of highly accurate hands from the InterHand2.6M dataset were artificially noised (both hands slightly rotated) with a range of probabilities from 0 to 1. The objective is to demonstrate that the model with intersection loss performs better than its version without, as the model with intersection loss can avoid intersections and thus should be closer to the ground truth. Fig. 8 shows that with a higher probability of noise, the model with intersection loss linearly outperforms the baseline model (the number of intersections and is slightly better in MPJPE for noise probability higher than 0.8). This empirically proves the initial assumption, since the figure suggests that the baseline model has no perception of hand intersections, and higher noise leads to a less accurate estimate. The model with intersection loss results in better prediction.

G. Implementation details

The intersection loss requires a pre-trained occupancy network which consists of a PointNet encoder [5] with residual blocks that encode input 3D skeletons into a feature space and a decoder with conditional batch normalization that transforms the latent space into logits. In total, the occupancy network has 7 million parameters.

A CNN network for predicting the hands consists of a ResNet-50 [10] model that returns image features, and an MLP with a couple of fully connected linear layers (in total around 2m parameters) that regresses a 3D hand pose. The predicted hands from the MLP are used in the intersection loss with the occupancy network.

H. Limitations

As was mentioned in the section III-B describing the results on the InterHand2.6M dataset, the main limitation of our approach is that not all intersections are resolved.

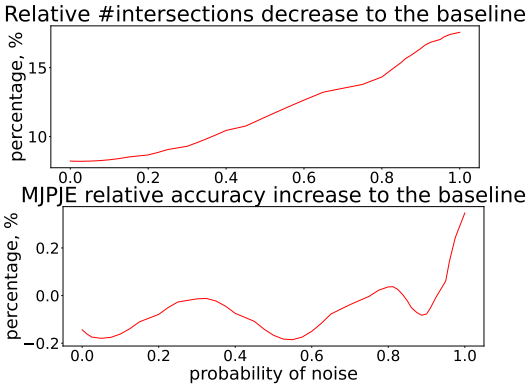


Fig. 8: The plots show the impact of artificially noising 3D training data on mean per-joint position error (MJPJE) and the number of intersections found with the occupancy network. For each noise probability, we compared two models – with and without intersection loss. The top graph demonstrates a decrease in the intersections. The bottom shows an increase in 3D accuracy for the model with intersection loss.

Primarily, this is caused by the weight of intersection loss in the model training, where a high weight leads to solving more intersections but deteriorates the 3D accuracy. A smaller weight, on the other hand, fixes fewer intersections but lowers the 3D error.

IV. CONCLUSIONS

This paper presents an intersection loss function for interacting 3D hand estimation to introduce physical constraints on hand estimation. By exploiting an occupancy network conditioned on a 3D skeleton, the hand volume is represented as a continuous manifold, where for an arbitrary 3D point, the occupancy model represents the likelihood of a hand intersection. With an extensive ablation study, we investigated the impact of testing different point sets for hand intersection on the model's accuracy and training speed. Additionally, we propose a custom hand mesh parameterization that overcomes some of the limitations of the MANO model, such as lower complexity, access to underlying 3D skeleton, watertightness, etc. The custom mesh serves a versatile purpose, enabling fast mesh generation essential for our occupancy network or mesh rendering, while providing a refined version tailored for mesh visualization.

The experiments on the benchmark InterHand2.6M dataset improve *state-of-the-art* models in both mean per-joint position error and significantly decrease the number of intersections. The cross-validation on the Re:InterHand and "in the wild" videos of sign language confirms a reduction of hand intersections both quantitatively and qualitatively, while maintaining the same 3D accuracy even though the hand estimator was trained on a very noisy dataset.

Acknowledgement. This work was supported by the SNSF project 'SMILE II' (CRSII5 193686), European Union's Horizon2020 programme ('EASIER' grant agreement 101016982) and the Innosuisse IICT Flagship (PFFS-21-47). This work reflects only the authors view and the Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] S. Albanie, G. Varol, L. Momeni, T. Afouras, J. S. Chung, N. Fox, and A. Zisserman. BSL-1K: Scaling up co-articulated sign language recognition using mouthing cues. In *European Conference on Computer Vision*, 2020. 7

[2] S. Albanie, G. Varol, L. Momeni, H. Bull, T. Afouras, H. Chowdhury, N. Fox, B. Woll, R. Cooper, A. McParland, and A. Zisserman. BOBSL: BBC-Oxford British Sign Language Dataset. 2021. 7

[3] L. Ballan, A. Taneja, J. Gall, L. Van Gool, and M. Pollefeys. Motion capture of hands in action using discriminative salient points. In *Computer Vision – ECCV 2012*, pages 640–653, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 1

[4] Z. Baowen, W. Yangang, D. Xiaoming, Z. Yinda, T. Ping, M. Cuixia, and W. Hongan. Interacting two-hand 3d pose and shape reconstruction from single color image. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11334–11343, 2021. 1

[5] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017. 8

[6] L. Chen, S.-Y. Lin, Y. Xie, H. Tang, Y. Xue, X. Xie, Y.-Y. Lin, and W. Fan. Generating realistic training images based on tonality-alignment generative adversarial networks for hand pose estimation. *ArXiv*, abs/1811.09916, 2018. 1

[7] S. Ebling, N. C. Camgöz, P. Boyes Braem, K. Tissi, S. Sidler-Miserez, S. Stoll, S. Hadfield, T. Haug, R. Bowden, S. Tornay, M. Razavi, and M. Magimai-Doss. SMILE Swiss German sign language dataset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). 2, 4, 6, 7

[8] Z. Fan, A. Spurr, M. Kocabas, S. Tang, M. Black, and O. Hilliges. Learning to disambiguate strongly interacting hands via probabilistic per-pixel part segmentation. In *International Conference on 3D Vision (3DV)*, 2021. 1

[9] S. Hampali, S. D. Sarkar, M. Rad, and V. Lepetit. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *IEEE Computer Vision and Pattern Recognition Conference*, 2022. 6

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2, 8

[11] M. Ivashechkin, O. Mendez, and R. Bowden. Improving 3d pose estimation for sign language. In *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pages 1–5, 2023. 4, 6

[12] C. Jiang, Y. Xiao, C. Wu, M. Zhang, J. Zheng, Z. Cao, and J. T. Zhou. A2j-transformer: Anchor-to-joint transformer network for 3d interacting hand pose estimation from a single rgb image, 2023. 2, 5

[13] K. Karunratanakul, A. Spurr, Z. Fan, O. Hilliges, and S. Tang. A skeleton-driven neural occupancy representation for articulated hands. In *International Conference on 3D Vision (3DV)*, 2021. 2

[14] O. Koller, J. Forster, and H. Ney. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding*, 141:108–125, Dec. 2015. 7

[15] M. Li, L. An, H. Zhang, L. Wu, F. Chen, T. Yu, and Y. Liu. Interacting attention graph for single image two-hand reconstruction. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 2, 5

[16] X. Li, P. Michel, A. Anastasopoulos, Y. Belinkov, N. Durrani, O. Firat, P. Koehn, G. Neubig, J. Pino, and H. Sajjad. Findings of the first shared task on machine translation robustness. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 91–102, Florence, Italy, Aug. 2019. Association for Computational Linguistics. 7

[17] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C. Chang, M. G. Yong, J. Lee, W. Chang, W. Hua, M. Georg, and M. Grundmann. Mediapipe: A framework for building perception pipelines. *CoRR*, abs/1906.08172, 2019. 6

[18] H. Meng, S. Jin, W. Liu, C. Qian, M. Lin, W. Ouyang, and P. Luo. 3d interacting hand pose estimation by hand de-occlusion and removal. October 2022. 1

[19] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[20] G. Moon, S. Saito, W. Xu, R. Joshi, J. Buffalini, H. Bellan, N. Rosen, J. Richardson, M. Mallorie, P. Bree, T. Simon, B. Peng, S. Garg, K. McPhail, and T. Shiratori. A dataset of relighted 3D interacting hands. In *NeurIPS Track on Datasets and Benchmarks*, 2023. 2, 6

[21] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee. Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *European Conference on Computer Vision (ECCV)*, 2020. 2, 4, 6

[22] F. Mueller, M. Davis, F. Bernard, O. Sotnychenko, M. Verschoor, M. A. Otaduy, D. Casas, and C. Theobalt. Real-time pose and shape reconstruction of two interacting hands with a single depth camera. *ACM Trans. Graph.*, 38(4), jul 2019. 1

[23] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Tracking the articulated motion of two strongly interacting hands. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1862–1869, 2012. 1

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 4

[25] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017. 1

[26] Y. Rong, J. Wang, Z. Liu, and C. C. Loy. Monocular 3d reconstruction of interacting handsvia collision-aware factorized refinements. In *International Conference on 3D Vision*, 2021. 1

[27] S. D. Roth. Ray casting for modeling solids. *Computer Graphics and Image Processing*, 18(2):109–144, 1982. 3

[28] B. Smith, C. Wu, H. Wen, P. Peluse, Y. Sheikh, J. K. Hodgins, and T. Shiratori. Constraining dense hand surface tracking with elasticity. *ACM Trans. Graph.*, 39(6), nov 2020. 1

[29] A. Spurr, A. Dahiya, X. Wang, X. Zhang, and O. Hilliges. Self-supervised 3d hand pose estimation from monocular rgb via contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11230–11239, 2021. 1

[30] A. Spurr, J. Song, S. Park, and O. Hilliges. Cross-modal deep variational hand pose estimation. In *CVPR*, 2018. 1

[31] J. Taylor, V. Tankovich, D. Tang, C. Keskin, D. Kim, P. Davidson, A. Kowdle, and S. Izadi. Articulated distance fields for ultra-fast tracking of hands interacting. *ACM Trans. Graph.*, 36(6), nov 2017. 1

[32] J. Wang, F. Mueller, F. Bernard, S. Sorli, O. Sotnychenko, N. Qian, M. A. Otaduy, D. Casas, and C. Theobalt. Rgb2hands: Real-time tracking of 3d hand interactions from monocular rgb video. *ACM Trans. Graph.*, 39(6), nov 2020. 1

[33] F. Xiong, B. Zhang, Y. Xiao, Z. Cao, T. Yu, J. Zhou, and J. Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 793–802, Los Alamitos, CA, USA, nov 2019. IEEE Computer Society. 2, 6

[34] L. Yang and A. Yao. Disentangling latent hands for image synthesis and pose estimation. pages 9869–9878, 06 2019. 1

[35] Z. Yu, S. Huang, F. Chen, T. P. Breckon, and J. Wang. Acr: Attention collaboration-based regressor for arbitrary two-hand reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023. 2

[36] B. Zhang, Y. Wang, X. Deng, Y. Zhang, P. Tan, C. Ma, and H. Wang. Interacting two-hand 3d pose and shape reconstruction from single color image. In *International Conference on Computer Vision (ICCV)*, 2021. 6

[37] R. Zhao, Y. Wang, and A. Martinez. A simple, fast and highly-accurate algorithm to recover 3d shape from 2d landmarks on a single image, 2016. 1

[38] C. Zimmermann and T. Brox. Learning to estimate 3d hand pose from single rgb images. Technical report, arXiv:1705.01389, 2017. https://arxiv.org/abs/1705.01389. 1