GANerated Hands for Real-Time 3D Hand Tracking from Monocular RGB

Franziska Mueller[1,2] Florian Bernard[1,2] Oleksandr Sotnychenko[1,2] Dushyant Mehta[1,2]
Srinath Sridhar[3] Dan Casas[4] Christian Theobalt[1,2]

[1] MPI Informatics [2] Saarland Informatics Campus [3] Stanford University [4] Univ. Rey Juan Carlos

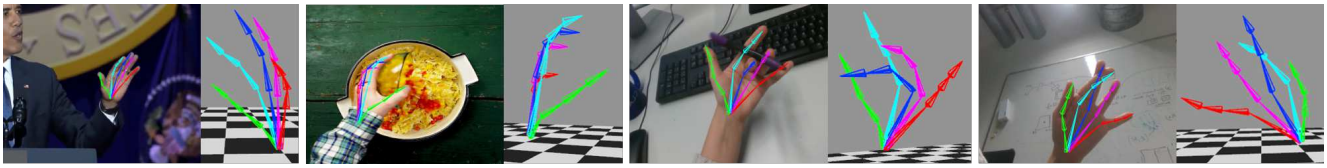


Figure 1: We present an approach for real-time 3D hand tracking from monocular RGB-only input. Our method is compatible with unconstrained video input such as community videos from YouTube (left), and robust to occlusions (center-left). We show real-time 3D hand tracking results using an off-the-shelf RGB webcam in unconstrained setups (center-right, right).

Abstract

We address the highly challenging problem of real-time 3D hand tracking based on a monocular RGB-only sequence. Our tracking method combines a convolutional neural network with a kinematic 3D hand model, such that it generalizes well to unseen data, is robust to occlusions and varying camera viewpoints, and leads to anatomically plausible as well as temporally smooth hand motions. For training our CNN we propose a novel approach for the synthetic generation of training data that is based on a geometrically consistent image-to-image translation network. To be more specific, we use a neural network that translates synthetic images to "real" images, such that the so-generated images follow the same statistical distribution as real-world hand images. For training this translation network we combine an adversarial loss and a cycle-consistency loss with a geometric consistency loss in order to preserve geometric properties (such as hand pose) during translation. We demonstrate that our hand tracking system outperforms the current state-of-the-art on challenging RGB-only footage.

1. Introduction

Estimating the 3D pose of the hand is a long-standing goal in computer vision with many applications such as in virtual/augmented reality (VR/AR) [17, 26] and human–computer interaction [38, 18]. While there is a large body of existing works that consider marker-free image-based hand tracking or pose estimation, many of them require depth cameras [34, 47, 39, 44, 27, 7, 54] or multi-view setups [41, 1, 56]. However, in many applications these requirements are unfavorable since such hardware is less ubiquitous, more expensive, and does not work in all scenes.

In contrast, we address these issues and propose a new algorithm for *real-time skeletal 3D hand tracking* with a *single color camera* that is robust under object *occlusion and clutter*. Recent developments that consider RGB-only markerless hand tracking [36, 63, 8] come with clear limitations. For example, the approach by Simon *et al*. [36] achieves the estimation of 3D joint locations within a multi-view setup; however in the monocular setting only 2D joint locations are estimated. Similarly, the method by Gomez-Donoso *et al*. [8] is also limited to 2D. Recently, Zimmermann and Brox [63] presented a 3D hand pose estimation method from monocular RGB which, however, only obtains relative 3D positions and struggles with occlusions.

Inspired by recent work on hand and body tracking [49, 20, 19], we combine CNN-based 2D and 3D hand joint predictions with a kinematic fitting step to track hands in global 3D from monocular RGB. The major issue of such (supervised) learning-based approaches is the requirement of suitable *annotated* training data. While it has been shown to be feasible to manually annotate 2D joint locations in single-view RGB images [14], it is impossible to accurately annotate in 3D due to the inherent depth ambiguities. One way to overcome this issue is to leverage existing multi-camera methods for tracking hand motion in 3D [41, 1, 56, 8]. However, the resulting annotations would lack precision due to inevitable tracking errors. Some works render synthetic hands for which the perfect ground truth is known [20, 63]. However, CNNs trained on synthetic

data may not always generalize well to real-world images. Hence, we propose a method to *generate suitable training data* by performing image-to-image translation between synthetic and real images. We impose two strong requirements on this method. First, we want to be able to train on *unpaired images* so that we can easily collect a large-scale real hands dataset. Second, we need the algorithm to preserve the pose of the hand such that the annotations of the synthetic images are still valid for the translated images. To this end, we leverage the seminal work on CycleGANs [62], which successfully learns various image-to-image translation tasks with unpaired examples. We extend it with a *geometric consistency loss* which improves the results in scenarios where we only want to learn spatially localized (*e.g*. only the hand part) image-to-image conversions, producing pose-preserving results with less texture bleeding and sharper contours. Once this network is trained, we can use it to translate any synthetically generated image into a "real" image while preserving the perfect (and inexpensive) ground truth annotation. Throughout the rest of the paper we denote images as "real" (in quotes), or *GANerated*, when we refer to synthetic images after they have been processed by our translation network such that they follow the same statistical distribution as real-world images.

Finally, using annotated RGB images produced by our GAN, we train a CNN that jointly regresses image-space 2D and root-relative 3D hand joint positions. While the skeletal hand model in combination with the 2D predictions are sufficient to estimate the global translation of the hand, the relative 3D positions resolve the inherent ambiguities in global rotation and articulation which occur in the 2D positions. In summary, our main contributions are:

- The first real-time hand tracking system that tracks *global 3D hand pose* from unconstrained monocular RGB-only images.
- A novel geometrically consistent GAN that performs image-to-image translation while preserving poses during translation.
- Based on this network, we are able to *enhance synthetic hand image datasets* such that the statistical distribution resembles real-world hand images.
- A *new RGB dataset* with annotated 3D hand joint positions. We overcome existing datasets in terms of size (>260k frames), image fidelity, and annotation precision.

2. Related Work

Our goal is to track hand pose from *unconstrained monocular RGB video streams* at real-time framerates. This is a challenging problem due the large pose space, occlusions due to objects, depth ambiguity, appearance variation due to lighting and skin tone, and camera viewpoint variation. While glove-based solutions would address some of these challenges [57], they are cumbersome to wear. Thus, in the following we restrict our discussion to markerless camera-based methods that try to tackle these challenges.

Multi-view methods: The use of multiple RGB cameras considerably alleviates occlusions during hand motion and interaction. Wang *et al*. [56] demonstrated hand tracking with two cameras using a discriminative approach to quickly find the closest pose in a database. Oikonomidis *et al*. [23] showed tracking of both the hand and a manipulated object using 8 calibrated cameras in a studio setup. Ballan *et al*. [1] also used 8 synchronized RGB cameras to estimate pose with added input from discriminatively detected points on the fingers. Sridhar *et al*. [41, 42] used 5 RGB cameras and an additional depth sensor to demonstrate real-time hand pose estimation. Panteleris and Argyros [24] propose using a short-baseline stereo camera for hand pose estimation without the need for a disparity map. All of the above approaches utilize multiple calibrated cameras, making it hard to setup and operate on general hand motions in unconstrained scenes (*e.g*. community videos). More recently, Simon *et al*. [36] proposed a method to generate large amounts of 2D and 3D hand pose data by using a panoptic camera setup which restricts natural motion and appearance variation. They also leverage their data for 2D hand pose estimation but cannot estimate 3D pose in monocular RGB videos. Our contributions address both data variation for general scenes and the difficult 3D pose estimation problem.

Monocular methods: Monocular methods for 3D hand pose estimation are preferable because they can be used for many applications without a setup overhead. The availability of inexpensive consumer depth sensors has lead to extensive research in using them for hand pose estimation. Hamer *et al*. [10] proposed one of the first generative methods to use monocular RGB-D data for hand tracking, even under partial occlusions. As such methods often suffer from issues due to local optima, a learning-based discriminative method was proposed by Keskin *et al*. [15]. Numerous follow-up works have been proposed to improve the generative component [44, 46, 51, 52], and the learning-based discriminator [58, 16, 45, 49, 55, 7, 37, 22, 61, 5, 4, 29]. Hybrid methods that combine the best of both generative and discriminative methods show the best performance on benchmark datasets [47, 39, 40, 20, 59].

Despite all the above-mentioned progress in monocular RGB-D or depth-based hand pose estimation, it is important to notice that these devices do not work in all scenes, *e.g*. outdoors due to interference with sunlight, and have higher power consumption. Furthermore, 3D hand pose estimation in unconstrained RGB videos would enable us to handle community videos, as shown in Figure 1. Some of the first methods for this problem [12, 43, 30] did not produce metrically accurate 3D pose as they only fetched the

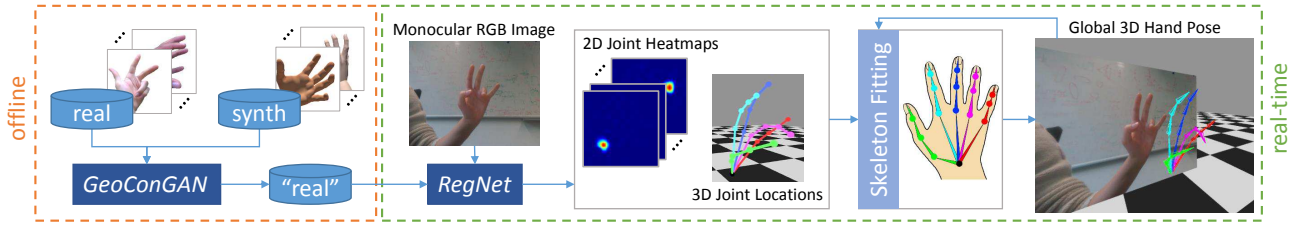


Figure 2: Pipeline of our real-time system for monocular RGB hand tracking in 3D.

nearest 3D neighbor for a given input or assume that the z-coordinate is fixed. Zimmermann and Brox [63] proposed a learning-based method to address this problem. However, their 3D joint predictions are *relative* to a canonical frame, *i.e*. the absolute coordinates are unknown, and it is not robust to occlusions by objects. Furthermore, their method is not able to distinguish 3D poses with the same 2D joint position projection since their 3D predictions are merely based on the abstract 2D heatmaps and do not directly take the image into account. In contrast, our work addresses these limitations by jointly learning 2D and 3D joint positions from image evidence, so that we are able to correctly estimate poses with ambiguous 2D joint positions. In addition, our skeleton fitting framework combines a prior hand model with these predictions to obtain global 3D coordinates.

Training of learning-based methods: One of the challenges in using learning-based models for hand pose estimation is the difficulty of obtaining annotated data with sufficient real-world variations. For depth-based hand pose estimation, multiple training datasets have been proposed that leverage generative model fitting to obtain ground truth annotations [49, 39] or to sample pose space better [21]. A multi-view bootstrapping approach was proposed by Simon *et al*. [36]. However, such outside-in capture setups could still suffer from occlusions due to objects being manipulated by the hand. Synthetic data is promising for obtaining perfect ground truth, but there exists a domain gap when models trained on this data are applied to real input [20].

Techniques like domain adaptation [6, 50, 25] aim to bridge the gap between real and synthetic data by learning features that are invariant to the underlying differences. Other techniques use real–synthetic image pairs [13, 32, 3] to train networks that can generate images that contain many features of real images. Because it is hard to obtain real–synthetic image pairs, Shrivastava *et al*. [35] recently proposed a synthetic-to-real refinement network requiring only *unpaired* examples. However, the extent of refinement is limited due to pixel-wise similarity constraints to the input. In contrast, the unpaired image-to-image translation work of Zhu *et al*. [62] relaxes these constraints to finding a bijection between the two domains. We build upon [62] to enable richer refinement and introduce a geometric consistency constraint to ensure valid annotation transfer. Without the need for corresponding real–synthetic image pairs, we can generate images of hands that contain many of the features found in real datasets.

3. Hand Tracking System

The main goal of this paper is to present a real-time system for monocular RGB-only hand tracking in 3D. The overall system is outlined in Fig. 2. Given a live monocular RGB-only video stream we use a CNN hand joint regressor, the *RegNet*, to predict 2D joint heatmaps and 3D joint positions (Sec. 3.2). The *RegNet* is trained with images that are generated by a novel image-to-image translation network, the *GeoConGAN*, (Sec. 3.1) that enriches synthetic hand images. The output images of the *GeoConGAN*—the *GANerated* images—are better suited to train a CNN that will work on real imagery. After joint regression, we fit a kinematic skeleton to both the 2D and 3D predictions by minimizing our fitting energy (Sec. 3.3), which has several key advantages for achieving a robust 3D hand pose tracking: it enforces biomechanical plausibility; we can retrieve the *absolute* 3D positions; and furthermore we are able to impose temporal stability across multiple frames.

3.1. Generation of Training Data

Since the annotation of 3D joint positions in hundreds of real hand images is infeasible, synthetically generated images are commonly used. While the main advantage of synthetic images is that the ground truth 3D joint positions are known, an important shortcoming is that they usually lack realism. Such discrepancy between real and synthetic images limits the generalization ability of a CNN trained only on the latter. In order to account for this disparity, we propose to use an image-to-image translation network, the *GeoConGAN*, with the objective to translate synthetic to real images. Most importantly, to train this network we use *unpaired* real and synthetic images, as will be described in the following. Note that for both the real and the synthetic data we use only foreground-segmented images that contain a hand on white background, which facilitates training and focuses the network capacity on the hand region.

Real hand image acquisition: To acquire our dataset of real images we used a green-screen setup to capture hand images with varying poses and camera extrinsics from 7 dif-

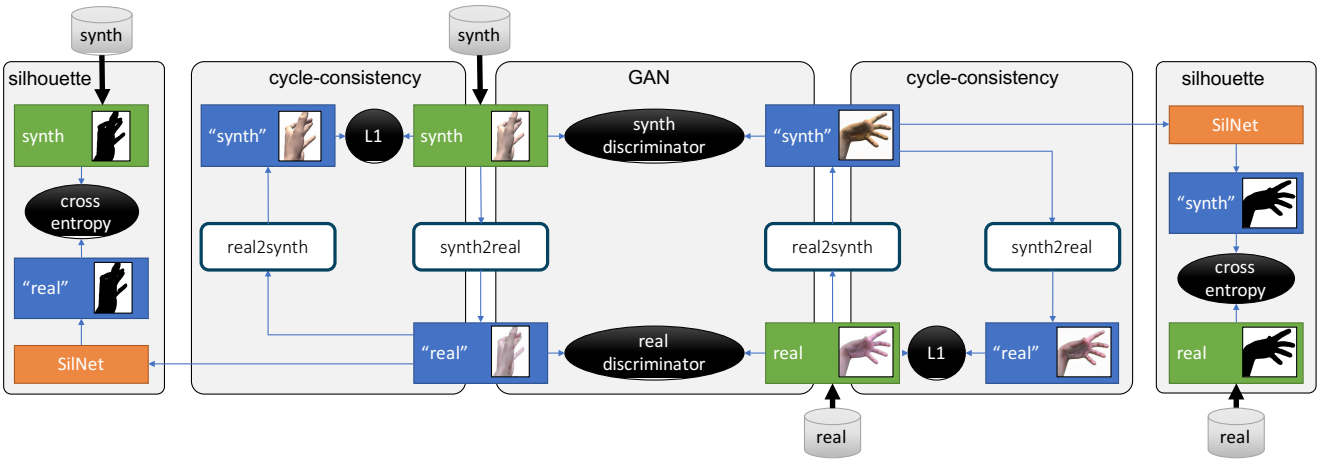


Figure 3: Network architecture of our *GeoConGAN*. The trainable part comprises the *real2synth* and the *synth2real* components, where we show both components twice for visualization purposes. The loss functions are shown in black, images from our database in green boxes, images generated by the networks in blue boxes, and the existing *SilNet* in orange boxes.

ferent subjects with different skin tones and hand shapes. In total, we captured 28,903 real hand images using a desktop webcam with image resolution 640×480.

Synthetic hand image generation: Our synthetic hand image dataset is a combination of the SynthHands dataset [20] that contains hand images from an egocentric viewpoint, with our own renderings of hand images from various third-person viewpoints. In order to generate the latter, we used the standard strategy in state-of-the-art datasets [20, 63], where the hand motion, obtained either via a hand tracker or a hand animation platform, is re-targeted to a kinematic 3D hand model.

Geometrically consistent CycleGAN (*GeoConGAN*): While the above procedure allows to generate a large amount of synthetic training images with diverse hand pose configurations, training a hand joint regression network based on synthetic images alone has the strong disadvantage that the so-trained network has limited generalization to real images, as we will demonstrate in Sec. 4.1

To tackle this problem, we train a network that translates synthetic images to "real" (or *GANerated*) images. Our translation network is based on CycleGAN [62], which uses adversarial discriminators [9] to simultaneously learn cycle-consistent forward and backward mappings. Cycle-consistency means that the composition of both mappings (in either direction) is the identity mapping. In our case we learn mappings from synthetic to real images (*synth2real*), and from real to synthetic images (*real2synth*). In contrast to many existing image-to-image or style transfer networks [13, 32], CycleGAN has the advantage that it does not require paired images, *i.e.* there must not exist a real image counterpart for a given synthetic image, which is crucial for our purpose due to the unavailability of such pairs.

The architecture of this *GeoConGAN* is illustrated in Fig. 3. The input to this network are (cropped) synthetic and real images of the hand on a white background in conjunction with their respective silhouettes, *i.e.* foreground segmentation masks. In its core, The *GeoConGAN* resembles CycleGAN [62] with its discriminator and cycle-consistency loss, as well as the two trainable translators *synth2real* and *real2synth*. However, unlike CycleGAN, we incorporate an additional geometric consistency loss (based on cross-entropy) that ensures that the *real2synth* and *synth2real* components produce images that *maintain the hand pose* during image translation. Enforcing consistent hand poses is of utmost importance in order to ensure that the ground truth joint locations of the synthetic images are also valid for the "real" images produced by *synth2real*. Fig. 4 shows the benefits of adding this new loss term.

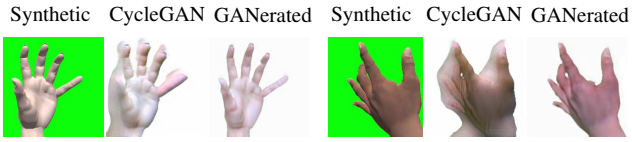


Figure 4: Our *GeoConGAN* translates from synthetic to real images by using an additional geometric consistency loss.

In order to extract the silhouettes of the images that are produced by both *real2synth* and *synth2real* (blue boxes in Fig. 3), we train a binary classification network, the *SilNet*, based on a simple UNet [31] that has three 2-strided convolutions and three deconvolutions. Note that this is a relatively easy task as the images have white background. We chose a *differentiable* network over naïve thresholding to make the training of *GeoConGAN* more well-behaved. Our *SilNet* is trained beforehand on a small disjoint subset of the data and is fixed while training *synth2real* and *real2synth*. Details can be found in the supplementary document.

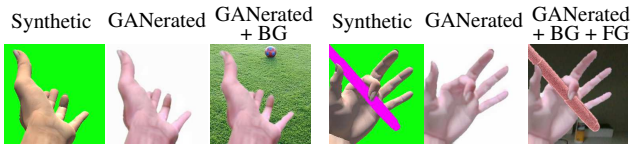


Figure 5: Two examples of synthetic images with background/object masks in green/pink.

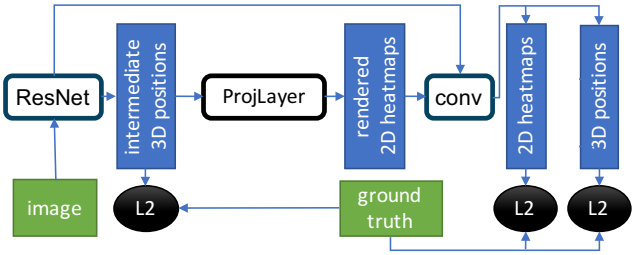


Figure 6: Architecture of *RegNet*. While only *ResNet* and *conv* are trainable, errors are still back-propagated through our *ProjLayer*. Input data is shown in green, data generated by the network in blue, and the loss is shown in black.

Data augmentation: Once the *GeoConGAN* is trained, we feed all synthetically generated images into the *synth2real* component and obtain the set of "real" images that have associated ground truth 3D joint locations. By using the background masks from the original synthetic images, we perform background augmentation by compositing GANerated images (foreground) with random images (background) [53, 19, 28]. Similarly, we also perform augmentation with a randomly textured object by leveraging the object masks produced when rendering the synthetic sequences [20]. Training on images without background or objects and hence employing data augmentation as post processing significantly eases the task for the *GeoConGAN*. Fig. 5 shows some GANerated images.

3.2. Hand Joints Regression

In order to regress the hand pose from a (cropped) RGB image of the hand, we train a CNN, the *RegNet*, that predicts 2D and 3D positions of 21 hand joints. The 2D joint positions are represented as heatmaps in image space, and the 3D positions are represented as 3D coordinates relative to the root joint. We have found that regressing both 2D and 3D joints are complementary to each other, as the 2D heatmaps are able to represent uncertainties, whereas the 3D positions resolve the depth ambiguities.

The *RegNet*, shown in Fig. 6, is based on a residual network consisting of 10 residual blocks that is derived from the *ResNet50* architecture [11], as done in [20]. Additionally, we incorporate a (differentiable) refinement module based on a projection layer (*ProjLayer*) to better coalesce the 2D and 3D predictions. The idea of the *ProjLayer* is to perform an orthographic projection of (preliminary) intermediate 3D predictions, from which 2D Gaussian heatmaps are created (within the layer). These heatmaps are then leveraged in the remaining part of the network (*conv*) to obtain the final 2D and 3D predictions. In Fig. 7a we show that this leads to improved results.

The training is based on a mixture of GANerated (Sec. 3.1) and synthetic images, in conjunction with corresponding 3D ground truth joint positions. The training set contains $\approx 440,000$ samples in total out of which 60% are GANerated. We empirically found that the performance on real test data does not further improve by increasing this percentage. We train the *RegNet* with *relative* 3D joint positions, which we compute by normalizing the absolute 3D ground truth joint positions such that the middle finger metacarpophalangeal (MCP) joint is at the origin and the distance between the wrist joint and the middle MCP joint is 1. Details can be found in the supplementary document.

During test time, *i.e.* for hand tracking, the input to the *RegNet* is a cropped RGB image, where the (square) bounding box is derived from the 2D detections of the previous frame. In the first frame, the square bounding box is located at the center of the image, with size equal to the input image height. Also, we filter the output of *RegNet* with the 1€ filter [2] to obtain temporally smoother predictions.

3.3. Kinematic Skeleton Fitting

After obtaining the 2D joint predictions in the form of heatmaps in image space, and the 3D joint coordinates relative to the root joint, we fit a kinematic skeleton model to this data. This ensures an anatomically plausible hand pose, while at the same time allowing to retrieve the *absolute* hand pose, as we will describe below. Moreover, when processing a sequence of images, *i.e.* performing hand *tracking*, we can additionally impose temporal smoothness.

Kinematic Hand Model: Our kinematic hand model is illustrated as *Skeleton Fitting* block in Fig. 2. The model comprises one root joint (the wrist) and 20 finger joints, resulting in a total number of 21 joints. Note, that this number includes the finger tips as joints without any degree-of-freedom. Let $\mathbf{t} \in \mathbb{R}^3$ and $\mathbf{R} \in \mathrm{SO}(3)$ (for convenience represented in Euler angles) be the global position and rotation of the root joint, and $\theta \in \mathbb{R}^{20}$ be the hand articulation angles of the 15 finger joints with one or two degrees-of-freedom. We stack all parameters into $\Theta = (\mathbf{t}, \mathbf{R}, \theta)$. By $\mathcal{M}(\Theta) \in \mathbb{R}^{J\times 3}$ we denote the *absolute* 3D positions of all J=21 hand joints (including the root joint and finger tips), where we use $\mathcal{M}_j(\Theta) \in \mathbb{R}^3$ to denote the position of the j-th joint. In order to compute the position for non-root joints, a traversal of the kinematic tree is conducted. Note that we take the camera coordinate system as global coordinate frame. To account for bone length variability across different users, we perform a *per-user skeleton adaptation*. The user-specific bone lengths are obtained by averaging

relative bone lengths of the 2D prediction over 30 frames while the users hold their hand parallel to the camera image plane. Up to a single factor due to the inherent scale ambiguity in RGB data, we can determine global 3D results which is important for many applications and not supported by previous work [63]. In addition, we obtain metrically accurate 3D results when provided with the metric length of a single bone. For model fitting, we minimize the energy

$$E(\Theta) = E_{\text{2D}}(\Theta) + E_{\text{3D}}(\Theta) + E_{\text{limits}}(\Theta) + E_{\text{temp}}(\Theta)\,, \quad (1)$$

where the individual energy terms are described below.

2D Fitting Term: The purpose of E_{2D} is to minimize the distance between the hand joint position projected onto the image plane and the heatmap maxima. It is given by

$$E_{\text{2D}}(\Theta) = \sum_j \omega_j \|\Pi(\mathcal{M}_j(\Theta))) - u_j\|_2^2\,, \quad (2)$$

where $u_j \in \mathbb{R}^2$ denotes the heatmap maxima of the j-th joint, $\omega_j > 0$ is a scalar confidence weight derived from the heatmap, and $\Pi : \mathbb{R}^3 \mapsto \mathbb{R}^2$ is the projection from 3D space to 2D image plane, which is based on the camera intrinsics. Note that this 2D term is essential in order to retrieve *absolute* 3D positions since the 3D fitting term E_{3D} takes only root-relative articulation into account, as described next.

3D Fitting Term: The term E_{3D} has the purpose to obtain a good hand articulation by using the predicted *relative* 3D joint positions. Moreover, this term resolves depth ambiguities that are present when using 2D joint positions only. We define E_{3D} as

$$E_{\text{3D}}(\Theta) = \sum_j \|(\mathcal{M}_j(\Theta) - \mathcal{M}_{\text{root}}(\Theta)) - z_j\|_2^2\,. \quad (3)$$

The variable $z_j \in \mathbb{R}^3$ is the user-specific position of the j-th joint relative to the root joint, which is computed from the output of the *RegNet*, x_j, as

$$z_j = z_{p(j)} + \frac{\|\mathcal{M}_j(\Theta) - \mathcal{M}_{p(j)}(\Theta)\|_2}{\|x_j - x_{p(j)}\|_2}(x_j - x_{p(j)})\,, \quad (4)$$

where $p(j)$ is the parent of joint j and we set $z_{\text{root}} = \mathbf{0} \in \mathbb{R}^3$. The idea of using user-specific positions is to avoid poor local minima caused by bone length inconsistencies between the hand model and the 3D predictions.

Joint Angle Constraints: The term E_{limits} penalizes anatomically implausible hand articulations by enforcing that joints do not bend too far. Mathematically, we define

$$E_{\text{limits}}(\theta) = \|\max(\left[\mathbf{0}, \theta - \theta^{\max}, \theta^{\min} - \theta\right])\|_2^2\,, \quad (5)$$

where $\theta^{\max}, \theta^{\min} \in \mathbb{R}^{20}$ are the upper and lower joint angle limits for the degrees-of-freedom of the non-root joints, and $\max : \mathbb{R}^{20\times 3} \mapsto \mathbb{R}^{20}$ computes the row-wise maximum.

Temporal Smoothness: The term E_{temp} penalizes deviations from constant velocity in Θ. We formulate

$$E_{\text{temp}}(\Theta) = \|(\nabla\Theta^{\text{prev}} - \nabla\Theta)\|_2^2\,, \quad (6)$$

where the gradients of the pose parameters Θ are determined using finite (backward) differences.

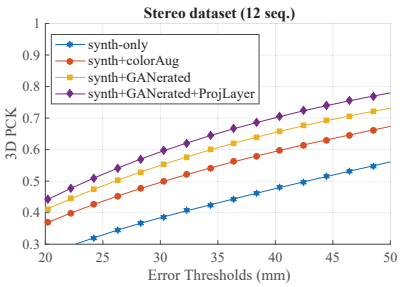
Optimization: In order to minimize the energy in (1) we use a gradient-descent strategy. For the first frame, θ and $\mathbf{t}$ are initialized to represent a flat hand that is centered in the image and 45cm away from the camera plane. For the remaining frames we use the translation and articulation parameters $\mathbf{t}$ and θ from the previous frame as initialization. In our experiments we have found that fast global hand rotations may lead to a poor optimization result corresponding to a local minimum in the non-convex energy landscape (1). In order to deal with this problem, for the global rotation $\mathbf{R}$ we do not rely on the previous value $\mathbf{R}^{\text{prev}}$, but instead initialize it based on the relative 3D joint predictions. Specifically, we make use of the observation that in the human hand the root joint and its four direct children joints of the non-thumb fingers (the respective MCP joints) are (approximately) rigid (cf. Fig. 2, *Skeleton Fitting* block). Thus, to find the global rotation $\mathbf{R}$ we solve the problem

$$\min_{\mathbf{R}\in\text{SO}(3)} \|\mathbf{R}\bar{Z} - \tilde{Z}\|_F^2\,, \quad (7)$$

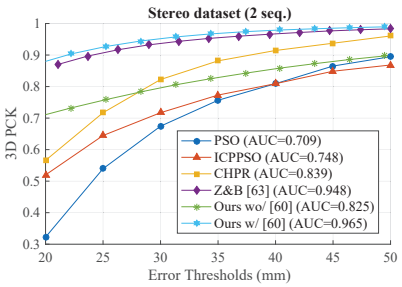
where $\bar{Z}$ contains (fixed) direction vectors derived from the *hand model*, and $\tilde{Z}$ contains the corresponding direction vectors that are derived from the current *RegNet predictions*. Both have the form $Z = \left[y_{j_1}, y_{j_2}, y_{j_3}, y_{j_4}, n\right] \in \mathbb{R}^{3\times 5}$, where the $y_{j_k} = \frac{1}{\|x_{j_k} - x_{\text{root}}\|}(x_{j_k} - x_{\text{root}}) \in \mathbb{R}^3$ are (normalized) vectors that point from the root joint to the respective non-thumb MCP joints $j_1, \ldots, j_4$, and $n = y_{j_1} \times y_{j_4}$ is the (approximate) normal vector of the "palm-plane". To obtain $\bar{Z}$ we compute the y_j based on the x_j of the 3D *model points* in world space, which is done only once for a skeleton at the beginning of the tracking when the global rotation of the model is identity. To obtain $\tilde{Z}$ in each frame, the x_j are set to the *RegNet predictions* for computing the y_j. While problem (7) is non-convex, it still admits the efficient computation of a global minimum as it is an instance of the *Orthogonal Procrustes Problem* [33, 48]: for $U\Sigma V^T$ being the singular value decomposition of $\tilde{Z}\bar{Z}^T \in \mathbb{R}^{3\times 3}$, the global optimum of (7) is given by $\mathbf{R} = U\,\text{diag}(1, 1, \det(UV^T))V^T$.

4. Experiments

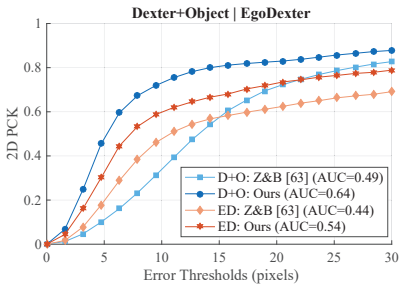
We quantitatively and qualitatively evaluate our method and compare our results with other state-of-the-art methods on a variety of publicly available datasets. For that, we use the Percentage of Correct Keypoints (PCK) score, a popular criterion to evaluate pose estimation accuracy. PCK defines a candidate keypoint to be correct if it falls within a circle (2D) or sphere (3D) of given radius around the ground truth.



(a) 3D PCK on Stereo dataset [60]. GANerated images (yellow) outperforms using only synthetic images (blue). Our projection layer (purple) to enforce 2D/3D consistency further improves accuracy.



(b) 3D PCK on the Stereo dataset [60]. We use 10 sequences for training and 2 for testing, as in [60, 63]. Our method (light blue) achieves the best results.



(c) 2D PCK on the Dexter+Object (D+O) dataset [40] and EgoDexter (ED) [20]. Our method (saturated blue/red) outperforms Z&B [63] on both datasets.

Figure 7: Quantitative evaluation. (a) Ablative study on different training options. (b), (c) 3D and 2D PCK comparison with state-of-the-art methods on publicly available datasets.

4.1. Quantitative Evaluation

Ablative study: In Fig. 7a we compare the accuracy when training our joint regressor *RegNet* with different types of training data. Specifically, we compare using synthetic images only, synthetic images plus color augmentation, and synthetic images in combination with GANerated images, where for the latter we also considered additionally using the *ProjLayer* in *RegNet*. For color augmentation, we employed gamma correction with random $\gamma \in [0.25, 2]$ sampled uniformly. While we evaluated the *RegNet* on the entire Stereo dataset [60] comprising 12 sequences, we did *not train on any* frame of the dataset for this test. We show that training on purely synthetic data leads to poor accuracy (3D PCK@50mm ≈ 0.55). While color augmentation on synthetic images improves the results, our GANerated images significantly outperform standard augmentation techniques, achieving a 3D PCK@50mm ≈ 0.80. This test validates the argument for using GANerated images.

Comparison to state-of-the-art: Fig. 7b evaluates our detection accuracy on the Stereo dataset, and compares it to existing methods [60, 63]. We followed the same evaluation protocol used in [63], *i.e*. we train on 10 sequences and test it on the other 2. Furthermore, [63] align their 3D prediction to the ground truth wrist which we also do for fairness. Our method outperforms all existing methods. Additionally, we test our approach *without* training on any sequence of the Stereo dataset, and demonstrate that we still outperform some of the existing works (green line in Fig. 7b). This demonstrates the generalization of our approach.

Figure 7c shows the 2D PCK, in pixels, on the Dexter+Object [40] and EgoDexter [20] datasets. We significantly outperform Zimmerman and Brox (Z&B) [63], which fails under difficult occlusions. Note that we cannot report 3D PCK since [63] only outputs root-relative 3D, and these datasets do not have root joint annotations.

4.2. Qualitative Evaluation

We qualitatively evaluate our method on three different video sources: publicly available datasets, real-time capture, and community (or vintage) video (*i.e*. YouTube).

Fig. 8 presents qualitative results on three datasets, Stereo [60], Dexter+Object [40] and EgoDexter [20], for both Z&B [63] and our method. We are able to provide robust tracking of the hand even under severe occlusions, and significantly improve over [63] in these cases. While we already outperformed Z&B [63] in our quantitative evaluation (Fig. 7c), we emphasize that this is not the full picture, since the datasets from [40, 20] only provide annotations for *visible* finger tips due to the manual annotation process. Thus, the error of occluded joints is not at all reflected in the quantitative analysis. Since our method is explicitly trained to deal with occlusion—in contrast to [63]—our qualitative analysis in the supplementary video and in columns 3–6 of Fig. 8 highlights our superiority in such scenarios.

We show real-time tracking results in Fig. 9 and in the supplementary video. This sequence was tracked live with a regular desktop webcam in an office environment. Note how our method accurately recovers the full 3D articulated pose of the hand. In Fig. 1 we demonstrate that our method is also compatible with community or vintage RGB video. In particular, we show 3D hand tracking in YouTube videos, which demonstrates the generalization of our method.

5. Limitations & Discussion

One difficult scenario for our method is when the background has similar appearance as the hand, as our *RegNet* struggles to obtain good predictions and thus tracking becomes unstable. This can be addressed by using an explicit segmenter, similar to Zimmermann and Brox [63]. Moreover, when multiple hands are close in the input image, de-

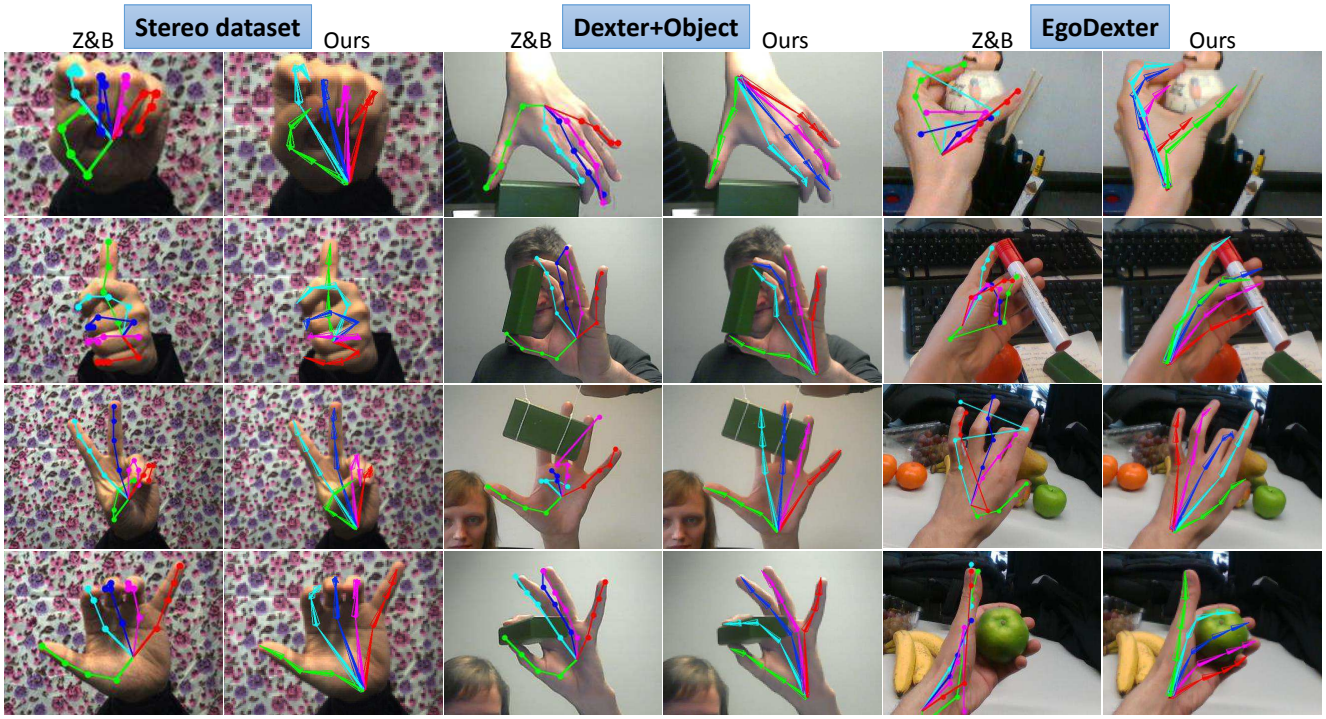


Figure 8: We compare our results with Zimmermann and Brox [63] on three different datasets. Our method is more robust in cluttered scenes and it even correctly retrieves the hand articulation when fingers are hidden behind objects.

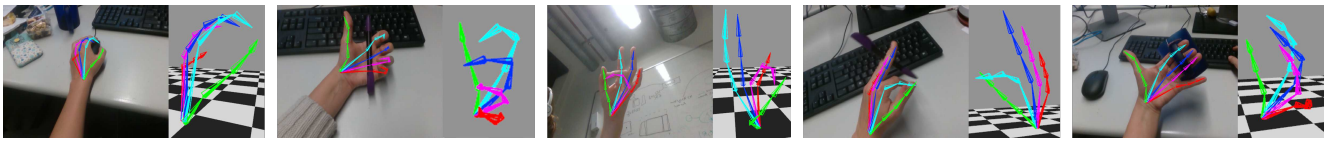


Figure 9: Representative frames of a live sequence captured with an off-the-shelf RGB webcam and tracked in real-time. Our method successfully recovers the global 3D positions of the hand joints. Here, for each input frame, we show the 3D tracked result projected to the camera plane, and the recovered 3D articulated hand skeleton visualized from a different viewpoint.

tections may be unreliable. While our approach can handle sufficiently separate hands—due to our bounding box tracker—tracking of interacting hands, or hands of multiple persons, is an interesting direction for follow-up work.

The 3D tracking of hands in purely 2D images is an extremely challenging problem. While our real-time method for 3D hand tracking outperforms state-of-the-art RGB-only methods, there is still an accuracy gap between our results and existing RGB-D methods (mean error of ≈5cm for our proposed RGB approach vs. ≈2cm for the RGB-D method of [40] on their dataset Dexter+Object). Nevertheless, we believe that our method is an important step towards democratizing RGB-only 3D hand tracking.

6. Conclusion

Most existing works either consider 2D hand tracking from monocular RGB, or they use additional inputs, such as depth images or multi-view RGB, to track the hand motion in 3D. While the recent method by Zimmermann and Brox [63] tackles monocular 3D hand tracking from RGB images, our proposed approach addresses the same problem but goes one step ahead with regards to several dimensions: our method obtains the *absolute* 3D hand pose by kinematic model fitting, is *more robust* to occlusions, and *generalizes better* due to enrichment of our synthetic data such that it resembles the distribution of real hand images. Our experimental evaluation demonstrates these benefits as our method significantly outperforms [63], particularly in difficult occlusion scenarios. In order to further encourage future work on monocular 3D RGB hand tracking we make our dataset available to the research community.

Acknowledgements: This work was supported by the ERC Starting Grant CapReal (335545). Dan Casas was supported by a Marie Curie Individual Fellow, grant 707326.

References

[1] L. Ballan, A. Taneja, J. Gall, L. V. Gool, and M. Pollefeys. Motion Capture of Hands in Action using Discriminative Salient Points. In *European Conference on Computer Vision (ECCV)*, 2012.

[2] G. Casiez, N. Roussel, and D. Vogel. 1€ filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2527–2530. ACM, 2012.

[3] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *International Conference on Computer Vision (ICCV)*, 2017.

[4] C. Choi, S. Ho Yoon, C.-N. Chen, and K. Ramani. Robust hand pose estimation during the interaction with an unknown object. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3123–3132, 2017.

[5] C. Choi, S. Kim, and K. Ramani. Learning hand articulations by hallucinating heat distribution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3104–3113, 2017.

[6] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.

[7] L. Ge, H. Liang, J. Yuan, and D. Thalmann. Robust 3D Hand Pose Estimation in Single Depth Images: from Single-View CNN to Multi-View CNNs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] F. Gomez-Donoso, S. Orts-Escolano, and M. Cazorla. Large-scale multiview 3d hand pose dataset. *arXiv preprint arXiv:1707.03742*, 2017.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[10] H. Hamer, K. Schindler, E. Koller-Meier, and L. Van Gool. Tracking a hand manipulating an object. In *IEEE International Conference On Computer Vision (ICCV)*, 2009.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[12] T. Heap and D. Hogg. Towards 3d hand tracking using a deformable model. In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, pages 140–145. IEEE, 1996.

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[14] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2010.

[15] C. Keskin, F. Kra¸, Y. E. Kara, and L. Akarun. Real time hand pose estimation using depth sensors. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 1228–1234, 2011.

[16] P. Krejov and R. Bowden. Multi-touchless: Real-time fingertip detection and tracking using geodesic maxima. In *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, pages 1–7, 2013.

[17] T. Lee and T. Hollerer. Multithreaded hybrid feature tracking for markerless augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):355–368, 2009.

[18] A. Markussen, M. R. Jakobsen, and K. Hornbæk. Vulture: A mid-air word-gesture keyboard. In *Proceedings of ACM Conference on Human Factors in Computing Systems*, CHI '14, pages 1073–1082. ACM, 2014.

[19] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. *ACM Transactions on Graphics (TOG)*, 36(4), 2017.

[20] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *International Conference on Computer Vision (ICCV)*, 2017.

[21] M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit. Efficiently Creating 3D Training Data for Fine Hand Pose Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a feedback loop for hand pose estimation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3316–3324, 2015.

[23] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2088–2095. IEEE, 2011.

[24] P. Panteleris and A. Argyros. Back to rgb: 3d tracking of hands and hand-object interactions based on short-baseline stereo. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[25] X. Peng and K. Saenko. Synthetic to real adaptation with deep generative correlation alignment networks. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[26] T. Piumsomboon, A. Clark, M. Billinghurst, and A. Cockburn. User-defined gestures for augmented reality. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2013.

[27] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and Robust Hand Tracking from Depth. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1106–1113, 2014.

[28] H. Rhodin, C. Richardt, D. Casas, E. Insafutdinov, M. Shafiei, H.-P. Seidel, B. Schiele, and C. Theobalt. Egocap: egocentric marker-less motion capture with two fisheye cameras. *ACM Transactions on Graphics (TOG)*, 35(6):162, 2016.

[29] G. Rogez, J. S. Supancic, and D. Ramanan. Understanding everyday hands in action from rgb-d images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3889–3897, 2015.

[30] J. Romero, H. Kjellström, and D. Kragic. Hands in action: real-time 3D reconstruction of hands in interaction with objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[31] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[32] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler: Controlling deep image synthesis with sketch and color. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[33] P. H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, Mar. 1966.

[34] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI)*, pages 3633–3642. ACM, 2015.

[35] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[36] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[37] A. Sinha, C. Choi, and K. Ramani. Deephand: robust hand pose estimation by completing a matrix imputed with deep features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4150–4158, 2016.

[38] S. Sridhar, A. M. Feit, C. Theobalt, and A. Oulasvirta. Investigating the dexterity of multi-finger input for mid-air text entry. In *ACM Conference on Human Factors in Computing Systems*, pages 3643–3652, 2015.

[39] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt. Fast and Robust Hand Tracking Using Detection-Guided Optimization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[40] S. Sridhar, F. Mueller, M. Zollhöefer, D. Casas, A. Oulasvirta, and C. Theobalt. Real-time Joint Tracking of a Hand Manipulating an Object from RGB-D Input. In *European Conference on Computer Vision (ECCV)*, 2016.

[41] S. Sridhar, A. Oulasvirta, and C. Theobalt. Interactive markerless articulated hand motion tracking using RGB and depth data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2456–2463, 2013.

[42] S. Sridhar, H. Rhodin, H.-P. Seidel, A. Oulasvirta, and C. Theobalt. Real-time hand tracking using a sum of anisotropic gaussians model. In *3D Vision (3DV), 2014 2nd International Conference on*, volume 1, pages 319–326. IEEE, 2014.

[43] B. Stenger, A. Thayananthan, P. H. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE transactions on pattern analysis and machine intelligence*, 28(9):1372–1384, 2006.

[44] A. Tagliasacchi, M. Schroeder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly. Robust Articulated-ICP for Real-Time Hand Tracking. *Computer Graphics Forum (Symposium on Geometry Processing)*, 34(5), 2015.

[45] D. Tang, H. Jin Chang, A. Tejani, and T.-K. Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3786–3793, 2014.

[46] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *Proc. ICCV*, 2015.

[47] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, et al. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics (TOG)*, 35(4):143, 2016.

[48] J. M. F. Ten Berge. The rigid orthogonal Procrustes rotation problem. *Psychometrika*, 71(1):201–205, 2006.

[49] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33, August 2014.

[50] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017.

[51] D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision (IJCV)*, 2016.

[52] D. Tzionas and J. Gall. 3d object reconstruction from hand-object interactions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 729–737, 2015.

[53] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[54] C. Wan, T. Probst, L. Van Gool, and A. Yao. Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 680–689, 2017.

[55] C. Wan, A. Yao, and L. Van Gool. Hand pose estimation from local surface normals. In *European Conference on Computer Vision*, pages 554–569. Springer, 2016.

[56] R. Wang, S. Paris, and J. Popović. 6d hands: markerless hand-tracking for computer aided design. In *Proc. of UIST*, pages 549–558. ACM, 2011.

[57] R. Y. Wang and J. Popović. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics*, 28(3), 2009.

[58] C. Xu and L. Cheng. Efficient hand pose estimation from a single depth image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3456–3462, 2013.

[59] Q. Ye, S. Yuan, and T.-K. Kim. Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. In *European Conference on Computer Vision (ECCV)*, pages 346–361. Springer, 2016.

[60] J. Zhang, J. Jiao, M. Chen, L. Qu, X. Xu, and Q. Yang. 3d hand pose tracking and estimation using stereo matching. *arXiv preprint arXiv:1610.07214*, 2016.

[61] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei. Model-based deep hand pose estimation. In *IJCAI*, 2016.

[62] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision (ICCV)*, 2017.

[63] C. Zimmermann and T. Brox. Learning to Estimate 3D Hand Pose from Single RGB Images. In *International Conference on Computer Vision (ICCV)*, 2017.