Bare-hand Depth Inpainting for 3D Tracking of Hand Interacting with Object

Woojin Cho* KAIST UVR lab.
Gabyong Park† KAIST UVR lab.
Woontack Woo‡ KAIST UVR lab.

ABSTRACT

We propose a 3D hand tracking system using bare-hand depth inpainting from an RGB-depth image for a hand interacting with an object. The effectiveness of most existing hand-object tracking methods is impeded by the insufficiency of data, which do not include hand data occluded by the object, and their reliance on the information inferred from assuming the specific object type. We generate a sufficiently accurate bare-hand depth image from a hand interacting with an object using a conditional generative adversarial network, which is trained using the synthesized 2D silhouettes of the object to learn the morphology of the hand. We evaluate the proposed approach using a hierarchical particle filter-based hand tracker and prove that our approach utilizing the bare-hand tracker in the hand-object interaction dataset achieve state-of-the-art performance. The generalization of our work will enable visual-tactile interaction that is more natural in various wearable augmented reality applications.

Index Terms: Artificial intelligence—Computer vision—Computer vision problems—Tracking; Human computer interaction (HCI)—Interaction paradigms—Mixed / augmented reality

1 INTRODUCTION

As the augmented reality (AR) technology evolves, the need for natural hand gesture-based interactions is growing. A real-time hand pose estimation can further develop natural hand gesture-based interactions. Although extensive studies have been conducted on the isolated hand pose estimation, scant scholarly attention has been devoted to tracking a hand interacting with an object. A recent study [51] shows that although tracking isolated bare hands has evolved to become sufficiently accurate, using a single camera to track the pose of a hand interacting with an object remains very challenging, especially in terms of object diversity. Owing to the foregoing challenges, a majority of the currently available head mounted display for AR offers hand-based interaction only in bare-handed situations, thereby compelling users to be bare-handed for the desired interaction, regardless of the current hand pose. In this paper, we introduce a novel framework for the 3D tracking of a hand interacting with an object using bare-hand depth inpainting, which is implemented with a conditional generative adversarial network (cGAN). The bare-hand generator was designed to generate a realistic bare-hand depth image from a given RGB-depth (RGB-D) image of the hand interacting with the object. To generalize the proposed approach to various objects, we produced a training dataset consisting of synthesized 2D silhouettes of random geometry. The inpainted bare-hand depth image was transferred to a hierarchical particle filter(HPF)-based tracker, and hand joint poses were computed.

Estimating the hand pose in a 3D space during interaction with objects is extremely challenging. When estimating the pose of an isolated hand, the self-similarity, self-occlusion, and highly articulated joint structure of the hand poses difficulty. In addition to these intrinsic problems, it is necessary to consider the occlusion by the object and the complex geometry of the object when tackling the pose estimation of a hand interacting with an object. Various approaches have been deployed to tackle these challenges. To directly solve several occlusion issues, a multi-camera system-based approach [39, 61] has been proposed. However, this approach tends to be expensive due to the calibration and synchronization required by the system. Using a single camera, a monocular RGB-based method [32, 47], and approaches based on RGB-D input have been proposed. For the RGB-D-based method, [16, 28, 40] developed model-based generative approaches. [17, 46] proposed a learning-based discriminative method, and [34, 50, 56] proposed a hybrid approach, which attempted to merge the advantages of both approaches.

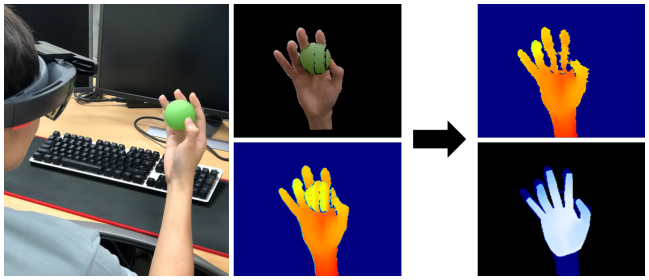

Figure 1: 3D tracking system for hand interacting with an object from RGB-depth input. From the camera input (left), we generated bare-hand depth map (upper right), and estimated full 3D pose of hand (lower right).

The majority of these works aimed to improve the hand tracking results by directly utilizing the information of objects as the elements of the dataset or constraint for hand poses. We conjectured that the strategy of utilizing the object's information was the reason why previous approaches have shown insufficient results to various types of objects that were not predefined or covered in the training sequence. As a solution, we proposed a method in which bare-hand data is generated, regardless of whether the hand is interacting with the object or not. Based on the fact that a hand can hold various objects in identical articulations, we assumed that a bare-hand depth map can be accurately generated based on the partially observed hand region, regardless of the geometry and 3D pose of the object being interacted with. Generally, the pose of the hand can be limited by the type of the object. However, to construct a system that considers daily-life objects, which mostly have a combination of basic geometry or an indefinable dynamic structure, we decided that it was more advantageous to deviate from utilizing the taxonomy of the objects.

The core idea is that the cGAN [21], which has been proposed for accomplishing a domain-to-domain translation, can be adapted to a domain that represents a hand interacting with an object and a bare hand. The critical point is not to create a merely realistic bare hand but one with the same articulation as a hand interacting

*e-mail: woojin.cho@kaist.ac.kr
†e-mail: gypark@kaist.ac.kr
‡e-mail: wwoo@kaist.ac.kr

with an object. Therefore, paired data, which consists of a hand with an object and a bare hand, are required. A similar approach was adopted in [60], where an attempt was made to reconstruct a bare hand point cloud from an object-hand point cloud using an autoencoder. However, because there were no clues to distinguish the object from the hand in this study, the approach was effective only in trained environments. However, we aimed to use RGB data as indirect segmentation information for objects and hand, which makes it possible to omit the hand-object segmentation process from the entire system. Although there are diverse large-scale bare-hand [55, 67] and object-hand [32, 34, 50] datasets, none of them satisfies our requirements.; Therefore, we constructed a new paired dataset by synthetically generating a random 2D silhouette of the object on the bare-hand dataset. In addition, to induce an intensive learning effect for the hand region occluded by the object, the binary mask of the object was extracted during dataset generation and utilized only in the training step.

Our main contributions can be summarized as follows:

- A novel method that supports the robust performance of real-time 3D hand pose estimation for a hand interacting with an object by generating a bare-hand depth image from observed RGB-D data.
- A technique for generalizing the type of the object being interacted with by synthesizing the 2D silhouette of the object, rather than using a 3D model of the object.
- The first real-time hand tracking system that utilizes a complete bare-hand tracker in a hand-object interaction context.

2 Related Work

In this section, we review recent works on bare-hand tracking, hand-object tracking, and depth inpainting domain, which are the key components of the proposed system.

3D Hand Tracking: Existing approaches to bare-hand hand tracking can be classified into three types: generative, discriminative, and hybrid. The generative methods [31, 38, 45, 54] estimate the pose by optimizing the defined objective function, to reduce the discrepancy between the input image and 3D hand model. As they rely on the solution from the previous frame, they may fall into the local minima. The discriminative methods were adopted in [24, 35, 36, 68], where they aim to predict the entire hand joint location from the input directly. As reported in [66], although current discriminative methods are generally effective on large training datasets, they are not generalizable in an unseen environment. The hybrid approaches are an attempt to combine the strengths of the previous two methods. There are several perspectives on how to combine the respective elements. In most studies [33,41,49,52], the strengths of the generative methods have been fused, which can ensure continuous robustness or refine the initial estimation, by harnessing the strengths of the discriminative methods, which exhibit accuracy in a single frame and trainable features.

Although various research fields have recently adopted GANs [15], few studies have adopted them for hand pose estimation. [18, 59] utilized GAN to model the latent statistical relationship between the hand pose and corresponding depth image. Chen et al. [6] utilized GAN as a hand-depth map generator from an RGB input to regularize the 3D hand pose estimation model. Unlike our work, these studies do not utilize GAN to achieve the domain translation of a hand interacting with an object to a bare hand; however, they indicate that mapping to bare-hand depth images is sufficiently learnable and effective.

3D Hand-Object Tracking: The generative approach, which attempts to exclude the object and focus on the model of the hand [17, 47] exhibits limited performance where various articulations are concerned. Another approach [16, 28, 40] treats the hand and object as a respective parametric model. Assuming that the object type is known, Kyriazis et al. [28] proposed a collaborative system with a set of independent trackers to track a hand interacting with multiple objects. To tackle the problem with a dataset consisting of a hand and an object being interacted with, several discriminative approaches [32, 46, 47] for hand-object tracking have been proposed. For challenging situations, such as the first-person viewpoint with large occlusions and cluttered scenes, Mueller et al. [34] proposed a system that combines two convolutional neural networks (CNNs) for localization and estimation. Tekin et al. [53] developed a unified framework to simultaneously track a 3D hand, object, and action classes. To combine the advantages of the generative and discriminative approaches, Sridhar et al. [50] suggested combining discriminative handpart classification and generative pose optimization. Oberweger et al. [37] proposed a method for iteratively updating the pose prediction result using a CNN without a 3D model of the hand or object through a feedback loop. Some methods [13, 16] utilize interacting object as a constraint for the hand pose. Tzionas et al. [56] proposed a framework consisting of a single objective function with discriminatively trained salient points, collision detection, and physics simulation. Choi et al. [7] utilized the grasp classification through a trained CNN for hand pose regression.

All the previous approaches were effective in limited environments under various constraints while performing an additional process to utilize information related to the interacting objects or including a vast feature related to the objects in the latent space to be learned. In contrast, the proposed method, by performing translation from a given observation under a variety of environments to a bare-hand domain, realizes a less challenging tracking environment.

Depth Inpainting: Achieving full coverage scene data estimation in the 3D scene capture system is an unsolved challenge that has received significant attention. Therefore, various approaches to completing, enhancing, and refining acquired images via a secondary data-filling process have been proposed.

Among the specific research objectives, image completion for the plausible synthesis of large spatial areas of color images has also been actively studied. [26, 30] investigated approaches that can be adopted for depth filling. Notably, Criminisi et al. [8] studied exemplar-based inpainting, which is regularly used in the color and depth-filling process [9, 19]. It prioritizes the filling order based on the gradient along the target boundary; however, its performance degraded significantly in scenes that were not front-parallel view. In [42, 64], approaches that utilized CNNs aided by adversarial training were proposed. Pathak et al. [42] demonstrated that well-trained CNNs could generate the contents of an arbitrary image region, which is an objective similar to ours. Studies on removing the object from the scene depth or filling the depth region that is missing due to sensor calibration issues, noise, and artifacts have attracted attention due to significant challenges [5], which is different from inpainting color images. These studies can be broadly classified into spatial-based methods [57, 63, 65], temporal-based methods [4, 22], and spatio-temporal-based methods [43, 62]. However, there are many branches due to the input dependencies and required information domain.

Unlike a majority of the existing depth inpainting studies that focused on natural filling based on the structure and local features in the image, the objective of data filling after object removal is specific to the bare-hand context. Therefore, a technique that can be optimized for the specific domain, rather than a general depth inpainting technique, is required. Consequently, we focused on inpainting studies using recently proposed deep neural networks, which satisfied our objectives. Several methods have made advances in estimating depth from a single monocular color image [11, 14, 27] and performing super-resolution and upscaling for depth image [3, 29, 44]; typically, they learn the spatial and/or temporal feature within the scene to accomplish this task. Because our final goal was to track the 3D

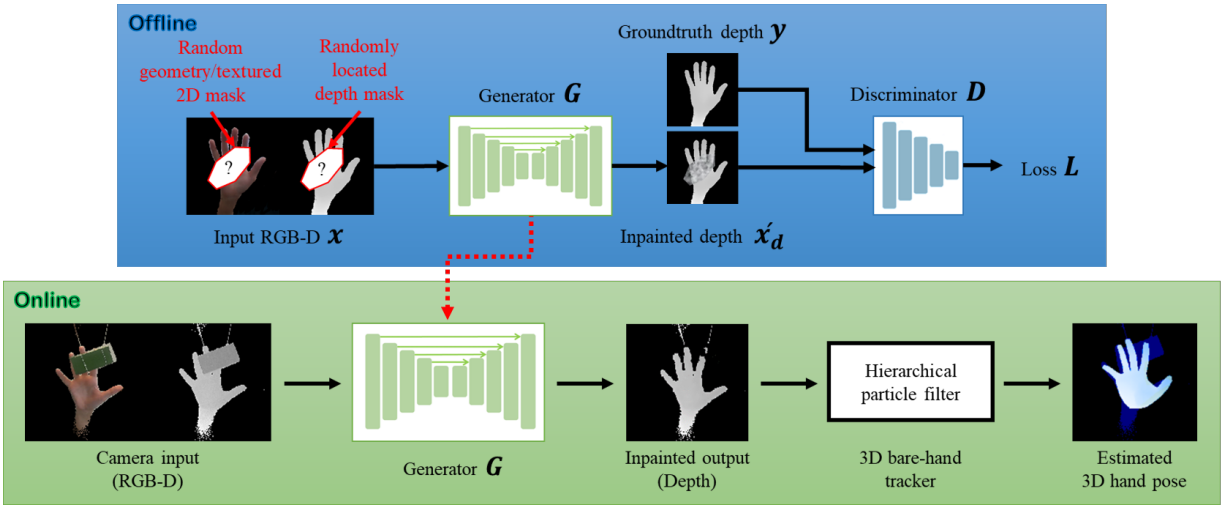

Figure 2: Schematic of proposed system. Generator for depth inpainting was trained using paired dataset consisting of synthesized object-hand input x and bare-hand output y. In the online step, trained generator receives RGB-D input and generates inpainted depth map. Based on this bare-hand domain depth map, bare-hand 3D tracker (HPF) estimates the final 3D hand pose.

hand pose accurately, the observable data corresponding to the real hand must be maintained as much as possible during the depth inpainting process. Therefore, we adopted the cGAN [21], which can directly compare the direct discrepancy between the ground-truth and generated output, rather than learn the mapping between two domains, bare-hand and occluded-hand, using an unpaired dataset.

3 Methodology

3.1 Overview

Our goal is to estimate the 26 degrees-of-freedom (DoF) hand pose by generating the depth image of the bare hand from the hand interacting with the object using a single RGB-D camera. The Intel RealSense SR300 was utilized to obtain 640×480 RGB and depth frames. A pipeline of the proposed method is shown in Fig.2. First, we preprocessed the input RGB and depth images, denoted as $c \in \mathbb{R}^{H \times W \times 3}$ and $d \in \mathbb{R}^{H \times W \times 1}$, from the camera to normalize the data and reduce the effect of the light condition. Then, the trained generator, denoted by G takes the concatenated RGB-D data as an input and generates the depth image, which would have the same pose as a bare hand. The generated output was transferred to the HPF-based tracker [31] to estimate the 26-DoF hand pose per frame.

Hand Model: We employed the 26 DoFs of a parametric hand model; six DoFs for the model's global 3D translation and 3D rotation (encoded as a quaternion); and four DoFs for each finger. Each finger was composed of three joints, a saddle joint at the base with two parameters, and two hinge joints with one parameter. From the 27 parameters of the hand model, we could sufficiently render most of the configuration of the hand. The discrepancy between the estimated hand pose hypothesis and the observed partial hand information was computed using the OpenGL pipeline.

3D Hand Tracker: Although the context of the experiment was a hand-object interaction situation, we exploited the various existing state-of-the-art bare-hand trackers. However, we had to utilize a tracker that relied solely on depth image, not color information, as the proposed method generates a bare-hand depth image from a given input. The HPF for the method proposed by Makris et al. [31] for tracking hand articulations was utilized in our system.

3.2 Loss function

The goal of the bare-hand depth inpainting network is to learn a mapping function between two domains A (hand holding an object) to B (bare hand) by learning how to estimate the overall configuration of the bare hand from the observed partial hand information. We denoted the dataset $\{x_i, y_i\}_{i=1}^{N}$ where input $x_i \in A$ and desired output $y_i \in B$ with dimensionality d. Let $x \sim [x_c, x_d], y \sim [y_d]$, denote that x is a concatenated image of color and depth channel, and y is a single-depth channel.

Our objective contained three types of term: conditional adversarial loss L_{cGAN}, context-conditional loss L_{cc}; and L1-norm loss L_1. L_{cc} was adopted from [10], which was proposed to enhance the training effectiveness for a specific patch of the image. The basic objective of a conditional GAN can be expressed as follows:

$$\mathscr{L}_{cGAN}(G,D) = \mathbb{E}_{x,y}[\log D(x,y)] \tag{1}$$
$$+ \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z)))] \tag{2}$$

The input random vector z was provided in the form of a dropout in the generator, which [21] proved to be more effective than the direct noise vector.

Although the proposed D was activated on the patch-level, rather than on the entire image, we added the context-conditional loss inspired by [10] to specify the training on the region that was occluded by the object. With the synthetic object mask extracted in the process of generating the dataset, D received the combined depth map in which the generated output $G(x)$ is filled in the region corresponding to the object mask on the original input depth image x_d, rather than the entire generated output. If the data generated for the object region was not consistent with the surrounding hand depth of the image, we assumed this value disparity to be a valid cue for the discriminator to preserve global continuity. Formally, let $m \in \mathbb{R}_d$ denote a binary mask of the object region; the operator $\odot$ denotes element-wise multiplication. To express the merging of the image between the output of the generator G and input depth image x_d based on the object mask m, we defined the operator $\otimes$ as follows:

$$m \otimes G(x,z) = m \odot G(x,z) + (1-m) \odot x_d \tag{3}$$

Then, L_{cc} can be formulated as follows:

$$\mathscr{L}_{cc}(G,D) = \mathbb{E}_{x,m,z}[\log(1 - D(x, m \otimes G(x,z)))] \tag{4}$$

Combining the GAN objective with a traditional loss, such as L2 distance [42] or L1 distance [21], has been beneficial for inducing the generator to be near the ground-truth output. Thus, we also added the $L1$ distance loss between the ground-truth output and generated output.

$$\mathscr{L}_{L1}(G) = \mathbb{E}_{x,y,z} \parallel y - G(x,z) \parallel_1 \quad (5)$$

Our final objective was combined as follows,

$$G^* = \arg\min_G \max_D \mathscr{L}_{cGAN}(G,D) + \lambda_1 \mathscr{L}_{cc}(G,D) + \lambda_2 \mathscr{L}_{L1}(G) \quad (6)$$

3.3 Data generation

Training the generator necessitated the use of a pair of RGB-D image, which consists of a bare hand and a hand interacting with an object, with the same hand articulation. As stated in Section 1, no public dataset satisfied the above conditions. Therefore, we acquired a bare-hand RGB-D dataset and synthetically generated random 2D silhouettes, representing the projected form of the object.

Hand-object interaction datasets have been created in previous studies [13, 34] through the augmentation of a 3D model of an object, which is then combined with a virtual hand model or real hand. However, because the variety of objects that can be created are practically limited, only the trained object categories indicated good results during the testing phase. We observed that although the 3D geometric form of the object may vary, the 2D silhouette projected by the camera was significantly less diverse. In other words, if a 3D object corresponds to a parametric space according to the geometric form, the 2D object silhouette can be expressed as a reduced dimension of the 3D object space. Therefore, we opted not to augment 3D objects; rather, we created arbitrary 2D object silhouettes. The generated dataset included a pair in which a synthetic object partially occluded a hand, as well as a pair in which the hand was not occluded at all. Thus, we could satisfy the purpose of generating a bare hand, regardless of whether there was interaction with an object.

For the quantitative evaluation, we acquired the RGB-D images of a hand interacting with various types of real objects selected with reference to [12]. We collected cuboid, spherical, and cylindrical objects. Moreover, intricately formed objects and deformable ones were included to verify the generalizability of the proposed approach.

Real Hand Data Acquisition: We captured several sequences in mid-air, and segmented the hand region using depth thresholding to acquire the bare-hand depth and calibrated color images. We attempted to capture a natural, sufficiently self-occluded hand motion from a 3rd-person viewpoint, rather than a static sequence. Further, the dataset included motions in which the front and back sides of the hand were exposed. Assuming the circumstances surrounding the hand localization were not accurate, an additional sequence was recorded in every peripheral part, based on the triaxial distance from the camera. In total, 6,127 bare-hand frames were captured.

Synthetic Object Generation: A synthetic object-hand dataset was generated by overwriting random texture color and randomly positioned depth maps in the identical pixels of the RGB and depth images of the bare-hand dataset. A schematic of the process is shown in Fig.3; this yields a paired RGB-D dataset consisting of the images of a bare hand and synthetic object-hand. The form of the synthetic 2D silhouette to be created and the location were determined based on the following parameters. Let $f_{uniform}(z_1, z_2)$ be the random number generator with uniform distribution within the range of z_1 to z_2. We used y_c, y_d to denote the RGB and depth image of the bare hand, respectively; o_x, o_y to denote the 2D position of the synthesized object on the image; o_d to denote the depth value of the object; and o_t to denote the color texture of the object. We defined the geometric form of the object using two variables, # of points of polygon N_p and length of each polygonal side $\{l_i\}_{i=1}^{N_p}$.

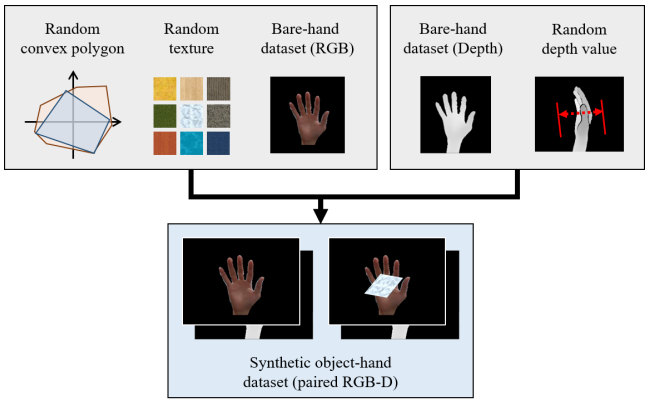

Figure 3: Synthetic object-hand dataset generation schematic. Each 2D object geometry, texture, and depth value were assigned arbitrarily and applied to RGB-D bare-hand dataset.

The formulation can be written as follows:

$$o_x = f_{uniform}(\eta_1,\ W - \eta_1) \quad (7)$$
$$o_y = f_{uniform}(\eta_1,\ H - \eta_1) \quad (8)$$
$$o_d = f_{uniform}(min(y_d) - \eta_2,\ min(y_d) + \eta_2) \quad (9)$$
$$o_t \in S_{texture} \quad (10)$$
$$N_p = f_{uniform}(4,\ 8) \quad (11)$$
$$l_i = f_{uniform}(l_{min},\ l_{max}) \quad (12)$$

where η_1 was set to add a margin to the position of the object, and η_2 was the threshold of the object-depth range, which were selected to evenly represent situations where the object was in front of, behind, or between fingers. $S_{texture}$ is a set of randomly selected 65 texture images prepared in advance. We adjusted the parameters for each length of the side l_{min}, l_{max} empirically by balancing the scale of the object and the bare hand. During synthesis, a binary mask of the object was extracted for the training sequence.

3.4 Training and Inference

Network Architecture: We adopted the architecture for our generative network from pix2pix [21]. Both the generator G and discriminator D used modules of the convolution-batchnorm-ReLu [20]. The generator G takes 4-channel input $x \in \mathbb{R}^{H \times W \times 4}$ and exports 1-channel output $x'_d \in \mathbb{R}^{H \times W \times 1}$. G contains skip connections, based on the general shape of a U-net [48]. For the discriminator D, 70×70 PatchGANs [21], which can be classified at a patch-level with fewer parameters than a full-image discriminator, were utilized.

Training: We trained the network in Tensorflow [1] with a total of $24.5k$ dataset and 45 epochs with a batch size of 1, initialized using the Adam optimizer [25], at the learning rate of 1×10^{-5}, $\beta_1 = 0.5, \beta_2 = 0.999, \lambda_1 = 2.0, \lambda_2 = 100$.

Inference: FFor inference, we conducted preprocesses to generate the network input x from the acquired camera input c, d. To localize the hand region of interest (ROI), we extracted the hand position roughly by segmenting the hand using a simple color-based threshold. As accurate hand segmentation, which can be achieved by several existing approaches [2, 23, 58], was not our focus, we assumed that the observable skin color had a small variation, which could be roughly segmented through manual thresholding. After localizing the ROI of the hand, we normalized the brightness elements in the dataset. The c, which was the RGB color space, was converted to a HSV color space, and the brightness value V was

equalized. Then, both c, d were resized to 256×256 and normalized in the range of $(-1, 1)$. The final network input x was generated by concatenating the preprocessed c and d. As the output of bare-hand generator $x'_d = G(x, z)$ had been scaled on $256 \times 256 \times 1$, we applied reverse normalization to the x'_d based on original minimum & maximum depth value and re-scaling with localized region information.

4 Experiments

As the proposed system was composed of the independent modules, i.e., the bare-hand inpainting network and hand pose tracker, we evaluated the accuracy of depth inpainting and 3D hand pose tracking through the entire process.

Test Dataset: The quantitative bare-hand depth inpainting experiment was conducted with a $2.4k$ synthetic object-hand test dataset, which was not included in the training dataset. For the hand pose tracking, we evaluated our method on the DexterHO dataset [50]. The dataset consisted of six sequences of RGB-D data and calibration parameters. It contained both 2D and 3D location annotations for the hand and object. However, because the hand only made slight contact with the object in four out of the six sequences, no situation existed in which the object directly occluded the hand. Conversely, although our proposed method was clearly effective for directly occluded sequences, in the other cases, the performance of the utilized bare hand tracker (HPF) was affected. Therefore, to verify the actual performance improvement of the proposed method, we only utilized the "occlusion" and "rotate" sequences for the quantitative experiment. We also utilized a modified form of the annotation by M. Oberweger et al. [37], which reported some erroneous annotation of the original data and made the corrected data available.

For qualitative results, we evaluated our system on the DexterHO and self-generated hand interaction sequences using various real objects. Specifically, we trained the bare-hand depth inpainting network on a synthetic object-hand dataset that did not contain any 3D model of the real object and none of the data from the test dataset, including DexterHO.

Evaluation Methods: To evaluate the inpainting accuracy, we computed the L1 norm average and median and the peak signal-to-noise ratio (PSNR). Additionally, we adopted the metric patch-based normalized cross-correlation (PNCC) proposed by A. Baruhov [3]. This measurement was proposed to show the correspondence between two images by computing the normalized cross-correlations between the local patches with overlap. Formally, the normalized cross-correlation function denotes $\rho(\cdot,\cdot)$, and N the total number of patches and two images, X and Y. The similarity, $PNCC(X,Y)$, can be defined as follows:

$$PNCC(X,Y) = \frac{1}{N} \sum_{i,j \in [0,s,2s,\ldots]} \rho(X^b_{i,j}, Y^b_{i,j}) \tag{13}$$

where $I^b_{i,j}$ is the patch of image I starting at pixel (i, j) and extending to $(i+b-1, j+b-1)$. We used $b = 16$, and $s = 4$, similar to the reference.

To evaluate the hand pose tracking accuracy, we used the percentage of correct keypoints (PCK) score on 2D and 3D. It defines based on keypoints if, compared with the ground-truth, the result satisfies the distance threshold as a circle (2D, pixel) or sphere (3D, *mm*). WWe compared our method with the state-of-the-art methods on the DexterHO dataset, using the 2D and 3D PCK results from Zimmermann et al. [68], Mueller et al. [32], Sridhar et al. [50] and Oberweger et al. [37]. [37,50] reported the pose estimation result for each sequence of the DexterHO dataset, and the comparison with this result is shown in Table.2.

Quantitative Evaluation: For the depth inpainting accuracy on the synthetic object-hand test dataset, both cases resulted in highly accurate inpainting, as shown in Table.1. In the case of the PNCC metric referenced in the [3], which was based on generating a low-quality depth to a high-quality depth image, the value outputted by the state-of-the-art method was 0.633. Based on this value, we demonstrated that our method generated an accurate bare-hand depth image. In addition, it was proven that the overall accuracy improved when L_{cc} was included.

Metric	Base	Base+L_{cc}
L1 norm avg.	5.836 mm	5.467 mm
L1 norm median.	0.656 mm	0.3487 mm
PSNR	28.71	31.12
PNCC	0.8967	0.9126

Table 1: Depth inpainting accuracy

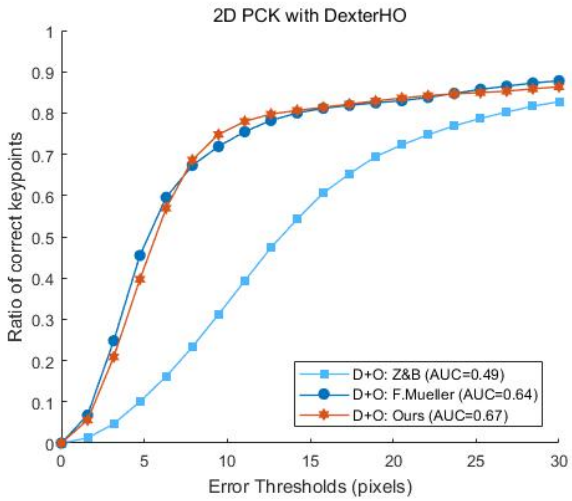

Figure 4: Quantitative results on DexterHO with 2D PCK. Z&B represents result from [68] and F.Mueller represents result from [32].

Method	Data type	Occlusion	Rotate	Average
Sridhar et al. [50]	RGB+Depth	17.5 mm	**16.3 mm**	16.9 mm
Oberweger et al. [37]	Depth	16.3 mm	17.9 mm	17.1 mm
Our work	RGB+Depth	**14.3 mm**	17.8 mm	**16.0 mm**

Table 2: 3D fingertip error comparison on DexterHO.

Fig.4 and 5 show the quantitative results on DexterHO with 2D PCK and 3D PCK. Because the ground-truth provided by the DexterHO dataset does not have data for occluded tips, the effect of depth inpainting on the joints that were directly occluded by the object was not reflected in the quantitative results. Nevertheless, both results showed comparable tracking accuracy with the state-of-the-art methods. The average 3D fingertip error is shown in Table.2; the proposed method significantly outperformed the other methods. The proposed approach successfully generated an effective bare-hand depth image from the given data. Note that the pose tracker we utilized was designed for the bare-hand situation.

Qualitative Evaluation:

The qualitative experiment was conducted for the DexterHO dataset and self-generated object-hand sequences. As shown in Fig.6, a sufficiently accurate inpainted depth image enabled an HPF to successfully track the pose of the hand interacting with the object. From the result of inpainting in the second row, we can observe a failure to completely generate the index finger. However, this

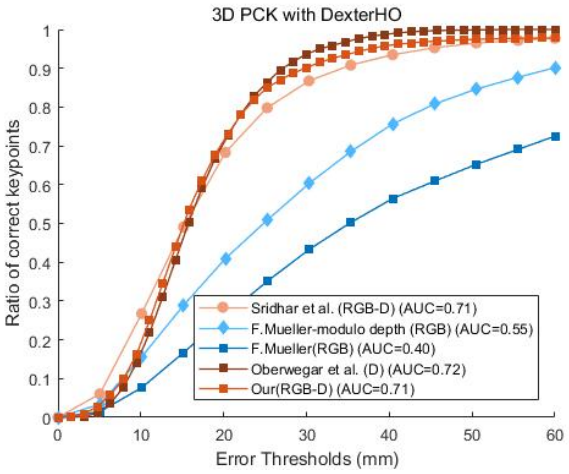

Figure 5: Quantitative results on DexterHO using 3D PCK. Sridhar et al. represents result from [50], F.Mueller represents result from [32] and Oberwegar et al., from [37].

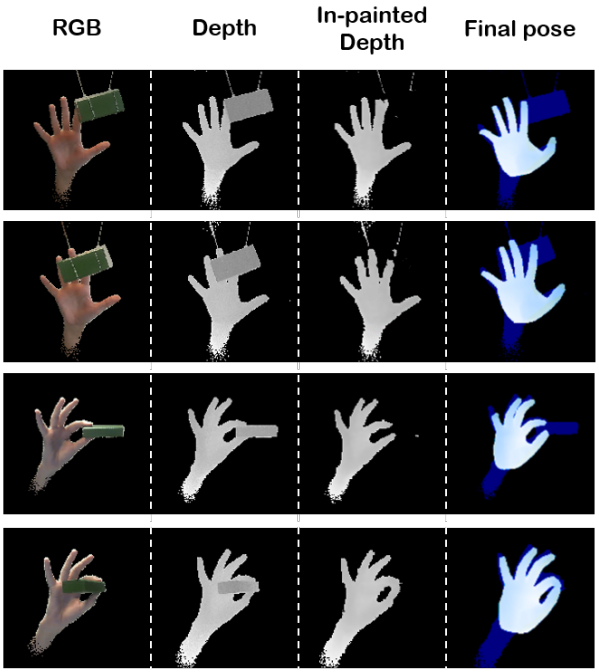

Figure 6: Qualitative results on DexterHO dataset.

incomplete depth region still served as an adequate cue for the bare-hand tracker, because of which the pose in the previous frame was maintained.

Fig.7 shows the qualitative results on sequences of interaction with simple objects that did not contain the ground-truth bare-hand image. The trained image consisted of the randomly generated 2D silhouette of the object; we found that the proposed method also produced an accurate bare-hand depth data for real objects. Fig.8 is the qualitative results for challenging objects with complex shape or texture or erroneous depth data. Although the inpainted depth was relatively inaccurate, some effort to eliminate the region corresponding to the object and generate depth data consistent with the observed region of the hand can be observed.

Performance: The overall system was implemented in Python using Tensorflow with an embedded c++ module, Intel Core i7, 32 GB of RAM, and an Nvidia GeForce GTX 1080 Ti GPU. The total runtime was 32 ms, 12 ms for depth inpainting and 20 ms for HPF. Thus, the proposed method ran at over 30 fps on a single GPU.

Limitation: The proposed method failed in situations that were not considered in the design of the system. For the specular object surface, erroneous data distribution occurred, as the depth data of the object was completely lost. As the synthesized paired training dataset had less real-world hand-object interaction pose distribution, including an egocentric viewpoint, the tracking results were not sufficiently accurate in the unseen environment. Furthermore, because the proposed framework was newly performed every frame without temporal information, we found that the inpainting results lack temporal continuity as an image was generated, regardless of the previous frame. Finally, in the case of a bare-hand situation in which there is no interaction with an object, although the depth inpainting result revealed an attempt to generate the same bare hand, unnecessary inference was observed in the raw depth data.

5 Conclusion

We presented a novel approach for the 3D tracking of a hand interacting with an object. Our approach utilized the cGAN to generate bare-depth images from RGB-D inputs, regardless of whether it was a bare hand or a hand interacting with an object. To generalize the method for various daily objects, we synthesized a 2D arbitrary geometry depth mask on the self-generated bare-hand dataset, to represent the projected silhouette of the object. Quantitative and qualitative experiments demonstrated the effectiveness of our approach in a situation where the hand interacted with various objects. With the system using a bare-hand tracker (HPF), the proposed method exhibited comparable performance to the state-of-the-art of the existing hand- object tracking methods. By enabling robust hand tracking during interaction with various unknown objects, the proposed method offered a higher DoF of interaction for AR users and facilitated the assumption of bare-handedness by developers, regardless of the object manipulated by the user. Our work can be further developed by adopting a semi-supervised learning technique and leveraging advanced GANs to perform an enhanced cross-domain transformation.

Acknowledgments

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-01270, WISE AR UI/UX Platform Development for Smartglasses) and Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (NRF- 2017M3C4A7066316).

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] A. A. Argyros and M. I. Lourakis. Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *European Conference on Computer Vision*, pp. 368–379. Springer, 2004.

[3] A. Baruhov and G. Gilboa. Unsupervised enhancement of real-world depth images using tri-cycle gan. *arXiv preprint arXiv:2001.03779*, 2020.

[4] Y. Berdnikov and D. Vatolin. Real-time depth map occlusion filling and scene background restoration for projected-pattern based depth cameras. In *Graphic Conf., IETP*, 2011.

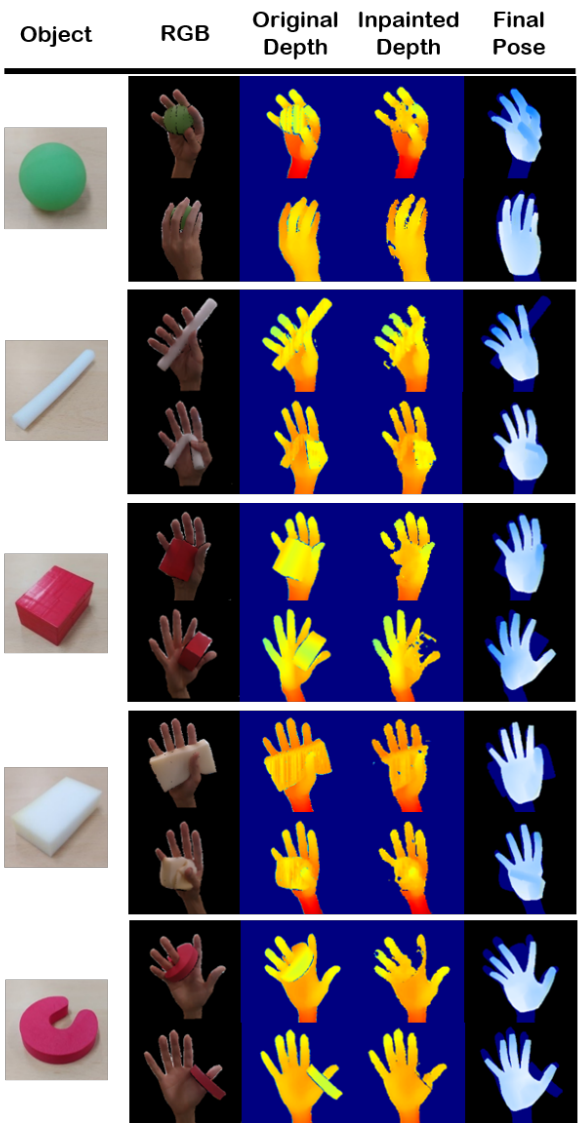

Figure 7: Qualitative results on simple object-hand sequence.

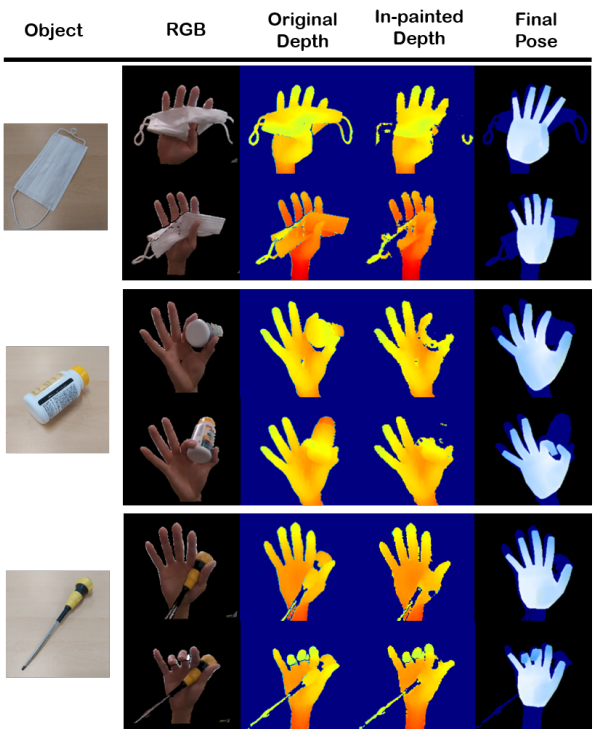

Figure 8: Qualitative results on complex object-hand sequence.

[5] T. P. Breckon and R. B. Fisher. Amodal volume completion: 3D visual completion. *In Computer Vision and Image Understanding*, 99(3):499–526, 2005.

[6] L. Chen, S.-Y. Lin, Y. Xie, Y.-Y. Lin, W. Fan, and X. Xie. DGGAN: Depth-image guided generative adversarial networks for disentangling RGB and depth images in 3D hand pose estimation. In *The IEEE Winter Conference on Applications of Computer Vision*, pp. 411–419, 2020.

[7] C. Choi, S. H. 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*, pp. 3123–3132, 2017.

[8] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *In IEEE Transactions on Image Processing*, 13(9):1200–1212, 2004.

[9] I. Daribo and H. Saito. A novel inpainting-based layered depth video for 3dtv. *In IEEE Transactions on Broadcasting*, 57(2):533–541, 2011.

[10] E. Denton, S. Gross, and R. Fergus. Semi-supervised learning with context-conditional generative adversarial networks. *arXiv preprint arXiv:1611.06430*, 2016.

[11] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2650–2658, 2015.

[12] T. Feix, I. M. Bullock, and A. M. Dollar. Analysis of human grasping behavior: Object characteristics and grasp type. *In IEEE Transactions on Haptics*, 7(3):311–323, 2014.

[13] Y. Gao, Y. Wang, P. Falco, N. Navab, and F. Tombari. Variational object-aware 3D hand pose from a single RGB image. *In IEEE Robotics and Automation Letters*, 4(4):4239–4246, 2019.

[14] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pp. 740–756. Springer, 2016.

[15] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye. A review on generative adversarial networks: Algorithms, theory, and applications. *arXiv preprint arXiv:2001.06937*, 2020.

[16] H. Hamer, J. Gall, T. Weise, and L. Van Gool. An object-dependent hand pose prior from sparse training data. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 671–678. IEEE, 2010.

[17] H. Hamer, K. Schindler, E. Koller-Meier, and L. Van Gool. Tracking a hand manipulating an object. In *IEEE 12th International Conference on Computer Vision*, pp. 1475–1482. IEEE, 2009.

[18] W. He, Z. Xie, Y. Li, X. Wang, and W. Cai. Synthesizing depth hand images with GANs and style transfer for hand pose estimation. *In Sensors*, 19(13):2919, 2019.

[19] A. Hervieu, N. Papadakis, A. Bugeau, P. Gargallo, and V. Caselles. Stereoscopic image inpainting: distinct depth maps and images inpainting. In *20th International Conference on Pattern Recognition*, pp. 4101–4104. IEEE, 2010.

[20] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[21] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134, 2017.

[22] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli,

J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pp. 559–568, 2011.

[23] B. Kang, K.-H. Tan, N. Jiang, H.-S. Tai, D. Tretter, and T. Nguyen. Hand segmentation for hand-object interaction from depth map. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 259–263, 2017.

[24] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun. Real time hand pose estimation using depth sensors. In *Consumer Depth Cameras for Computer Vision*, pp. 119–137. Springer, 2013.

[25] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] V. Kumar, J. Mukherjee, and S. K. D. Mandal. Image inpainting through metric labeling via guided patch mixing. *In IEEE Transactions on Image Processing*, 25(11):5212–5226, 2016.

[27] Y. Kuznietsov, J. Stuckler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6647–6655, 2017.

[28] N. Kyriazis and A. Argyros. Scalable 3D tracking of multiple interacting objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3430–3437, 2014.

[29] J. Lei, L. Li, H. Yue, F. Wu, N. Ling, and C. Hou. Depth map super-resolution considering view synthesis quality. *In IEEE Transactions on Image Processing*, 26(4):1732–1745, 2017.

[30] Y. Liu and V. Caselles. Exemplar-based image inpainting using multiscale graph cuts. *IEEE Transactions on Image Processing*, 22(5):1699–1711, 2012.

[31] A. Makris, N. Kyriazis, and A. A. Argyros. Hierarchical particle filtering for 3D hand tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 8–17, 2015.

[32] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt. Generated hands for real-time 3D hand tracking from monocular RGB. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 49–59, 2018.

[33] 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. *In ACM Transactions on Graphics (TOG)*, 38(4):1–13, 2019.

[34] 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 *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 1284–1293, 2017.

[35] M. Oberweger and V. Lepetit. Deepprior++: Improving fast and accurate 3D hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 585–594, 2017.

[36] M. Oberweger, P. Wohlhart, and V. Lepetit. Hands deep in deep learning for hand pose estimation. *arXiv preprint arXiv:1502.06807*, 2015.

[37] M. Oberweger, P. Wohlhart, and V. Lepetit. Generalized feedback loop for joint hand-object pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[38] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Efficient model-based 3D tracking of hand articulations using kinect. In *BmVC*, vol. 1, p. 3, 2011.

[39] 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 *2011 International Conference on Computer Vision*, pp. 2088–2095, 2011.

[40] P. Panteleris, N. Kyriazis, and A. A. Argyros. 3D tracking of human hands in interaction with unknown objects. In *BMVC*, pp. 123–1, 2015.

[41] G. Park and W. Woo. Hybrid 3D hand articulations tracking guided by classification and search space adaptation. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 57–69, 2018.

[42] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544, 2016.

[43] C. Richardt, C. Stoll, N. A. Dodgson, H.-P. Seidel, and C. Theobalt. Coherent spatiotemporal filtering, upsampling and rendering of RGBZ videos. In *Computer Graphics Forum*, vol. 31, pp. 247–256. Wiley Online Library, 2012.

[44] G. Riegler, D. Ferstl, M. Rüther, and H. Bischof. A deep primal-dual network for guided depth super-resolution. *arXiv preprint arXiv:1607.08569*, 2016.

[45] K. Roditakis, A. Makris, and A. A. Argyros. Generative 3D hand tracking with spatially constrained pose sampling. In *BMVC*, vol. 1, p. 2, 2017.

[46] 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*, pp. 3889–3897, 2015.

[47] J. Romero, H. Kjellström, and D. Kragic. Hands in action: real-time 3d reconstruction of hands in interaction with objects. In *2010 IEEE International Conference on Robotics and Automation*, pp. 458–463. IEEE, 2010.

[48] 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*, pp. 234–241. Springer, 2015.

[49] 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 the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 3633–3642, 2015.

[50] S. Sridhar, F. Mueller, M. Zollhöfer, 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*, pp. 294–310. Springer, 2016.

[51] J. S. Supančič, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan. Depth-based hand pose estimation: methods, data, and challenges. *International Journal of Computer Vision*, 126(11):1180–1198, 2018.

[52] 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. *In ACM Transactions on Graphics (TOG)*, 36(6):1–12, 2017.

[53] B. Tekin, F. Bogo, and M. Pollefeys. H+ o: Unified egocentric recognition of 3D hand-object poses and interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4511–4520, 2019.

[54] A. Tkach, A. Tagliasacchi, E. Remelli, M. Pauly, and A. Fitzgibbon. Online generative model personalization for hand tracking. *In ACM Transactions on Graphics (ToG)*, 36(6):1–11, 2017.

[55] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *In ACM Transactions on Graphics (ToG)*, 33(5):1–10, 2014.

[56] D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall. Capturing hands in action using discriminative salient points and physics simulation. *In International Journal of Computer Vision*, 118(2):172–193, 2016.

[57] K. R. Vijayanagar, M. Loghman, and J. Kim. Real-time refinement of kinect depth maps using multi-resolution anisotropic diffusion. *In Mobile Networks and Applications*, 19(3):414–425, 2014.

[58] T. Vodopivec, V. Lepetit, and P. Peer. Fine hand segmentation using convolutional neural networks. *arXiv preprint arXiv:1608.07454*, 2016.

[59] 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*, pp. 680–689, 2017.

[60] H. Wang. Hand pose estimation for hand-object interaction cases using augmented autoencoder. 2019.

[61] Y. Wang, J. Min, J. Zhang, Y. Liu, F. Xu, Q. Dai, and J. Chai. Video-based hand manipulation capture through composite motion control. *In ACM Transactions on Graphics (TOG)*, 32(4):1–14, 2013.

[62] K. Xu, J. Zhou, and Z. Wang. A method of hole-filling for the depth map generated by kinect with moving objects detection. In *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, pp. 1–5. IEEE, 2012.

[63] X. Xu, L.-M. Po, C.-H. Cheung, L. Feng, K.-H. Ng, and K.-W. Che-

ung. Depth-aided exemplar-based hole filling for DIBR view synthesis. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, pp. 2840–2843, 2013.

[64] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6721–6729, 2017.

[65] J. Yang, X. Ye, K. Li, C. Hou, and Y. Wang. Color-guided depth recovery from RGB-D data using an adaptive autoregressive model. *IEEE Transactions on Image Processing*, 23(8):3443–3458, 2014.

[66] S. Yuan, G. Garcia-Hernando, B. Stenger, G. Moon, J. Yong Chang, K. Mu Lee, P. Molchanov, J. Kautz, S. Honari, L. Ge, et al. Depth-based 3D hand pose estimation: From current achievements to future goals. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[67] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T.-K. Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4866–4874, 2017.

[68] C. Zimmermann and T. Brox. Learning to estimate 3D hand pose from single RGB images. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4903–4911, 2017.