# Variational Object-aware 3D Hand Pose from a Single RGB Image

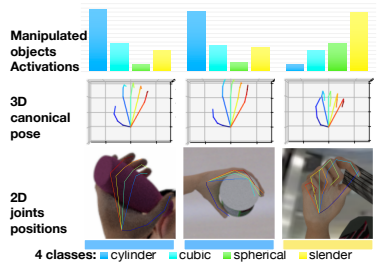
Yafei Gao[1,*], Yida Wang[1,*], Pietro Falco[2], Nassir Navab[1] and Federico Tombari[1,3]

***Abstract*—We propose an approach to estimate the 3D pose of a human hand while grasping objects from a single RGB image. Our approach is based on a probabilistic model implemented with deep architectures, which is used for regressing, respectively, the 2D hand joints heat maps and the 3D hand joints coordinates. We train our networks so to make our approach robust to large object- and self-occlusions, as commonly occurring with the task at hand. Using specialized latent variables, the deep architecture internally infers the category of the grasped object so to enhance the 3D reconstruction, based on the underlying assumption that objects of a similar category, i.e. with similar shape and size, are grasped in a similar way. Moreover, given the scarcity of 3D hand-object manipulation benchmarks with joint annotations, we propose a new annotated synthetic dataset with realistic images, hand masks, joint masks and 3D joints coordinates. Our approach is flexible as it does not require depth information, sensor calibration, data gloves, or finger markers. We quantitatively evaluate it on synthetic datasets achieving state-of-the-art accuracy, as well as qualitatively on real sequences.**

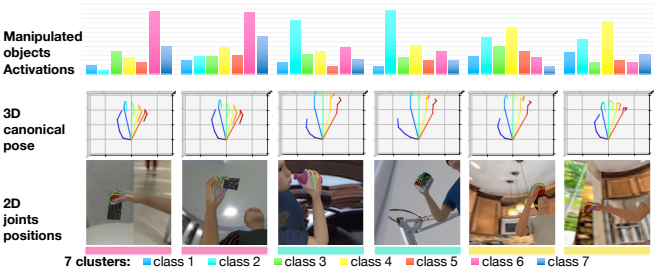
***Index Terms*—variational inference, triplet**

## I. Introduction

HAND pose estimation is now a required technology for many emerging consumer applications such as virtual and augmented reality (VR, AR), robotics, gaming, and human-machine interface. Concerning robotics, a key problem in the scientific community is to program both stationary robots and modern mobile manipulators without strong technical background. The classical programming techniques can be optimal in industrial production lines where the environment is completely structured. However, in applications that require human-robot collaboration and in new areas of service robotics such as logistics, healthcare, and house automation, robotic systems have to be reprogrammed in an intuitive and easy way by nonexpert users. An effective way to instruct robots in an intuitive fashion is programming by demonstration, where the robot observes humans performing a task and learns to reproduce it. A key bottleneck, especially for manipulation tasks, is the observation of human hands while grasping and manipulating objects, as this presents the challenge of occlusion, since parts of the hand are dynamically occluded by the grasped object. Notably, also the other aforementioned applications such as AR, VR and gaming need to deal with hand-object interaction and occlusion.




(a) training with object labels




(b) training without object labels

Fig. 1: 3D hand pose estimation from monocular under heavy occlusion. The canonical hand joint configurations for objects of the same category are similar. Hence, we leverage specialized latent features to regress accurate hand pose based on the detected object's information.

In this paper, we address the problem of estimating the hand pose in 3D space while grasping objects using a single RGB camera. 3D hand pose means the 3D positions of the hand joints with respect to a frame fixed to the wrist. As it consumes data only from a monocular camera, our system does not require any depth sensor, sensor calibration, data gloves as in [1], or finger markers [2]. To the best of our knowledge, this is the first work that addresses discriminative hand pose estimation using monocular image as input while the hand interacts with an object. We propose a method based on two stages, each carried out via a deep architecture which are released here [1]. Given an RGB image, the first step is to filter the hand out of the environment and to generate a heat map that highlights the joints' location. To deal with the frequent occlusions of the hand caused by the grasped objects, a deep

Manuscript received: February, 24, 2019; Revised: May, 31, 2019; Accepted: July, 2, 2019.

This paper was recommended for publication by Editor Dongheui Lee upon evaluation of the Associate Editor and Reviewers' comments.

[1]CAMP, Technische Universität München, Boltzmannstr. 3 85748 Garching yafei.gao@tum.de, yida.wang@tum.de, navab@cs.tum.edu, tombari@in.tum.de

[2]Deptôf Automation Solutions, ABB Corporate Research, Forskargränd 7, 72178 Västerås, Sweden pietro.falco@se.abb.com

[3]Google Zurich

* The first two authors contributed equally to this work.

Digital Object Identifier (DOI): see top of this page.

[1] *https://github.com/wangyida/VO-handpose*

architecture with variational latent feature is purposely trained to reconstruct the occluded parts. Then, the 3D coordinates of the joints are estimated based on the generated heat map. In order to enhance the 3D pose estimation procedure, we introduce additional features in the latent space that specialize to the shape and size of the grasped object to influence the final hand pose regression. The underlying assumption for this is that, for grasping similar objects in terms of shape and size (e.g., from the same category), the joint (knuckle) configurations of the two grasping poses are similar to each other (see Fig. 1). In addition, given the current lack in literature of 3D hand-object manipulation benchmarks with joint annotations, we have generated a new annotated synthetic dataset that includes realistic images, hand masks, joint masks and 3D joints coordinates.

To summarize, the main contributions of this work are: i) a novel variational deep architecture to reconstruct a 2D hand mask in presence of large occlusions and to identify a heat-map of hand joints' locations using a monocular image; ii) a deep network that estimates 3D hand pose from the 2D heat map, leveraging latent features that identify the object category to accurately estimate the hand pose; iii) a novel synthetic dataset of hands grasping objects with rich annotations.

## II. Related Works

### A. 3D Bare-Hand Pose Estimation

Approaches for 3D bare hand pose estimation can be sorted into two categories: discriminative (appearance-based) and generative (model-based). While generative approaches classify input $X$ with expected output $Y$ by learning a model of joint probability $P(X, Y)$ and using Bayes rule to compute $P(Y|X)$, discriminative methods classify $X$ by learning a direct map between $X$ and $Y$ or directly estimating the posterior probability.

Most generative approaches for hand pose try to fit a parametric model to the data by generating model hypotheses and evaluating them on the observed data [3], [4], [5]. Instead, discriminative approaches directly learn a forward-pass function from training data and establish a mapping from image to hand pose [6], [7]. In this case, a critical factor is represented by finding a suitable learning model which has the ability to handle a large amount of features and possible hand poses [8], [9]. As deep architectures handle more complex tasks, convolutional neural networks (CNNs) fully utilise stacked convolutional operations to predict 2D joint locations for real-time continuous pose recovery from a single depth image [10]. Depth images alse help estimate 3D poses [11], [12], [13]. With input of RGB images, 3D models are also used to produce 2D samples to learn 3D Orientation of objects [14]. Recently, Zimmermann et al. [15] propose a concatenated architecture to estimate hand segmentation, joints position and 3D poses sequentially.

### B. Hand Pose Estimation with Object Interaction

Due to substantially increased occlusion caused by the objects, related work in this field primarily falls within the generative category and either assumes simplified working conditions (e.g., empty background) or employs additional input modalities (e.g., multi-view or depth data). A differentiable objective function for pose estimation is proposed in [16] where edges, optical flow, salient points and collisions are used to capture the motion of two hands interacting with an object on an empty background. Kyriazis [17] suggests an ensemble of collaborative trackers to handle multi-object scenarios based on RGB-D data as input. As for discriminative approaches, Romero proposes a hand pose retrieval approach for RGB images, where nearest neighbors from a large hand pose database are retrieved based on object shape information [18]. A discriminative top-down approach is proposed in [19] using CNNs, able to estimate the hand joints and object locations from a depth camera. First object pixels are segmented out, then a two-channel image containing both the input depth map and the masked depth map are used to regress the 3D joint position. However, experiments are proposed where only a tennis ball is used as interacting object, which exhibits a similar silhouette from diverse observation perspectives, consequently only simplified occlusion cases are taken into consideration. Another discriminative approach based on depth data is proposed by Choi [20], which uses parallel deep architectures and embeds object shape information into latent features. Two networks share intermediate observations produced from different perspectives to create a more informed representation. Instead of processing low-level data to detect or remove occluded regions, it exploits a CNN-based framework to extract grasp estimates from those regions. Interestingly, Choi's approach and our work share the assumption that there is a strong correlation between the object category and the grasping pose. Differently from Choi, we aim to solve the task using only monocular data.

### C. Hand-Object Datasets

To the best of our knowledge, there exist the following fully-annotated hand-object datasets: Hand-Sphere Dataset [19], SynthHands Dataset [21], GANerated Hands Dataset [22], First-Person Hand Action [23], and Stereo Dataset [24]. Hand-Sphere [19] captures hands grasping spherical objects using a Kinect sensor and providing both hand segmentation and pose estimation. For segmentation task, paired depth maps and RGB images are provided with 5635 samples for training and 1042 for testing, while the pose estimation dataset consists of 3986 samples for training and 745 for testing. Hand-Sphere [19] lacks diversity of manipulated objects since the object has a similar silhouette from different viewpoints, hence provides limited types of occlusion. Also, this characteristic does not allow to evaluate the assumption that hand poses are correlated to the shape and category of the grasped object. SynthHands [21] is a synthetic dataset for hand pose estimation from depth and color data, with and without object interaction. It uses a merged reality approach to capture and synthesize large amounts of annotated data of natural hand interaction in cluttered scenes. Occluded hand and interacting objects are directly observed. However, to implement grasping hand pose estimation task, many interactions within this dataset are physically invalid. Recently, GANerated Hands Dataset [22] was

proposed, containing more than 330,000 color images of hands with 2D and 3D annotations of 21 keypoints on a synthetic hand model. The use of GANs help increase the realism of the dataset, nevertheless physically invalid joint configurations still exist. In this dataset, the hand pixels are already segmented and extracted. Stereo Dataset [24] is composed by 18000 stereo image pairs and 18000 depth images captured from different scenarios and the ground-truth 3D positions of palm and finger joints obtained from the manual label.

Our dataset differs in two main aspects. First, it includes a variety of objects with labels. Second, it is designed to realistically simulate hand-object interaction, hence facilitating the use of parametric models trained on our dataset on real data. A dataset with object labels is important as it allows us to test approaches based on object category information.

## III. Methodology

This section describes the proposed approach for 3D hand pose estimation from a single image, devised to deal with large hand occlusion. Inspired by [15], we first segment the complete hand and regress joint locations as heatmaps, then use such joint locations to guide 3D pose estimation by regressing the 3D canonical and relative hand poses. As a reminder, 3D canonical poses are a set of absolute joint coordinates, while 3D relative poses depend from a specific viewpoint. In [15], three sub-architectures are used sequentially, namely HandSegNet, PoseNet and PosePrior. Since this approach estimates the pose of bare hands, it is not designed to deal with occlusions that occurr during hand-object interaction. To overcome this limitation, as depicted in Fig. 2, we propose to replace the discriminative model in [15] with a generative one based on two encoding-decoding networks and two learned latent spaces ($z_h$ and $z_j$ in Fig. 2), so to regress more robustly joint configurations and hand poses. Since the latent space is a compact representation of the input domain, we believe this approach enforces a large number of unrealistic poses to be dropped during inference, increasing the accuracy of the outcome.

In particular, first we replace HandSegNet (i.e., the network in [15] for hand segmentation) with a variational convolutional architecture based on an encoder-decoder pair and trained to regress hand masks and joint location heatmaps. This network is concatenated with another decoder, i.e. PoseNet in [15]. Then, we replace PosePrior, which estimates in parallel the canonical pose and the rotation matrix and obtains the relative pose by multiplying the canonical coordinate by the rotation matrix, with another variational auto-encoder, which is trained via triplet learning to regress 3D canonical and relative joint coordinates from the inferred joint heatmaps. Note that, similarly to [15], hand side is also necessary for pose inference. To describe more formally our pipeline in the following, let $X$ be the input RGB image, while the outputs are a hand segmentation mask ($Y_h$), a multi-channel joint heatmap $Y_j$ and a set of estimated 3D joint coordinates $Y_p$. Two latent features $z_h$ and $z_j$, which are extracted from $X$ and $Y_j$, are used to regress 2D maps $Y_h, Y_j$ and 3D poses $Y_p$. We apply, on both sub-architectures, variational inference and triplet training.

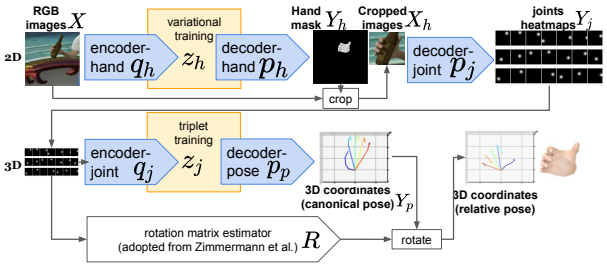



Fig. 2: Proposed architecture for 3D hand pose estimation.

### A. 2D Hand Mask and Joint Heatmap

Supervised CNNs for segmentation such as DCN [25] and stacked CNN [26] can effectively perform hand segmentation only on those parts of the hand that are visible, so HandSegNet [15] can not segment well the hand in presence of occlusion even if trained with complete hand masks, this leading to errors nearby hand joints. As shown in Fig. 5, the segmented hand region generated by [15] mostly contains visible hand parts, resulting in disconnected components in presence of occluding objects. To solve this problem, we apply variational inference on a latent feature $z_h$ of input $X$ and split the 2D estimator into an encoder-decoder pair. Instead of using loss functions targeting hand segmentation directly, we set two constraints to enhance the ability to obtain the 2D hand mask in presence of occlusion:

- the decoder $p_h(z_h)$ always generates a complete hand from latent feature $z_h$;
- the encoder $q_h(X)$ generates a latent feature $z_h^{'}$ which is likely to produce a complete hand.

First, the input image $X$ is encoded into the latent features $z_h$ by encoder $q_h$, then used to generate a hand mask $Y_h$ at $256 \times 256$ resolution via decoder $p_h$. Here the grasped object is removed once the hand mask is generated. Then, for the aim of extracting hand joint information in form of pixel-wise heat maps, $Y_j$ are generated from a decoder $p_j()$ with a masked hand $X_h$ as input. This means that $p_j$ is concatenated after $p_h$, so we can also generate $p_j$ directly from the latent features $z_h$ via a combined decoder $p_j(p_h)$.

$p_j$ is an extended convolutional architecture with 24 layers, that outputs a 21-channel heatmap of size $32 \times 32 \times 21$, each channel associated to one of the 21 joints. Hence, we combine the encoder and the 2 decoders together as $p_j(p_h(q_h))$, generating the 2D pose as the heat map $Y_j$ from the image $X$. Both the encoder $q_h(X)$ and the decoder $p_h(z_h)$ are simultaneously optimised with a variational constraint [27] for latent variables. The Kullback-Leibler (KL) divergence $D[Q_h(z|X)||P_h(z|Y)]$ between the posterior $P_h(z|Y)$ and a likelihood $Q_h(z|X)$ is used to evaluate the capability of the encoder to generate latent features which are likely to produce the expected target $Y$:

$$E_{z\sim Q}[\log Q_h(z|X) - \log P_h(Y|z) - \log P(z)] + \log P(Y) \ . \quad (1)$$

Since the likelihood term $Q_h(z|X)$ is hardly tractable, variational inference [27] solves this problem by redefining a specific encoding function $q_h(\cdot)$ with latent features $z = q_h(X)$ following a Gaussian distribution $N(0, I)$, such that the

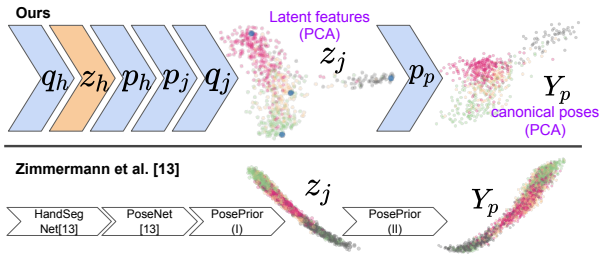



Fig. 3: Difference of data distribution after triplet training is applied for 3D pose estimation. Different colors represent different categories.

associated probability density function $Q_h(z|X)$ is expected to fit the posterior $P_h(z|Y)$. We modify equation (1) as follows to obtain the cost function for our encoder and decoder:

$$\begin{aligned}\mathcal{L}_{enc-hand} &= D[Q_h(z|X)||P(z)] \\ \mathcal{L}_{gen-hand} &= -E_{z\sim Q}[\log P_h(Y_h|z)] \ .\end{aligned} \tag{2}$$

The loss function in (2) can be interpreted as: the conditional log likelihood $\log P_h(X|z)$ used in variational auto-encoder [28] is replaced with $\log P_h(Y|z)$ where the expected network output is different from the input, so that the decoding process can be carried out for different targets $Y$.

Once the completed hand segmentation network is trained with equation (2), we use the generated hand mask to crop the original image $X$. In this case, the hand is centered in the image while all fingers are included for further processing. We can define a loss function for the decoder simply by replacing $p_h(z_h)$ with $p_j(p_h(z_h), X)$, i.e. the likelihood $P_h(Y_h|z)$ to generate hand masks is changed to generate joints maps $P_j(Y_j|z, X)$; thus the final loss function used to regress the 2D joint heatmap is

$$\mathcal{L}_{gen-joint} = -E_{z\sim Q}[\log P_j(Y_j|z, X)] \ . \tag{3}$$

Finally, we train hand segmentation and joint estimation together with the following loss:

$$\mathcal{L}_{2d} = \mathcal{L}_{enc-hand} + \mathcal{L}_{gen-hand} + \mathcal{L}_{enc-joint} + \mathcal{L}_{gen-joint} \ . \tag{4}$$

According to the architectural design, to get those latent features we apply 3 fully-connected layers with dimensions $[256, 128, 128]$ respectively, followed by ReLU [29] to process the output of the 18th convolutional layer in HandSegNet [15] and the output of the first fully-connected layer in PosePrior [15]. The decoder of our 2D hand joint estimator relies on the convolutional architecture used for people detection in [30], with which is concatenated with the decoder for hand segmentation.

### *B. Object-aware 3D Hand Pose Estimation*

As the last step of our model, we estimate the canonical and real 3D hand poses based on the images cropped by the generated 2D joint heatmaps $Y_j$. The canonical 3D hand pose [31] defines a 3D pose which is rotation invariant and independent of the camera view, so that a set of 2D hand joints can be projected into the same 3D canonical pose even if their real poses are different. Here we also apply variational inference on the latent features to make the predicted 3D pose more stable. Since humans grasp objects with a strategy that depends on the size and shape of the object, the 3D canonical hand poses should be similar to each other when grasping similar objects. Fig. 3 shows that the distribution of 3D hand poses (trained without using any explicit object label) becomes correlated to object categories. We then conclude that 3D hand pose estimation in presence of objects can benefit from knowing the object category, should such information be available in advance. However, at test time we want to predict the hand pose just from an RGB image, without any additional information. Therefore, instead of using object labels as an additional input, we implicitly add category information to the learned latent feature.

When estimating the 3D hand pose from the heatmap $Y_j$ using the combined encoder-decoder architecture $q_j, p_p$, we propose to use triplet training to optimize the latent features $z_{pose}$ produced by $q_j$, so that they form clusters driven by object categories. Since most datasets do not provide object labels for training, we introduce an approach to learn object-related latent clusters, by introducing additional latent variables which are used as cluster centers. Our unsupervised clustering could be used within stochastic gradient descent (SGD) [32] to train a deep network, which would not be possible for other clustering methods such as K-means [33].

**Object-driven Latent Features via Triplet Training** In case the training dataset contains annotations regarding the category of the manipulated object (such as for the proposed HOP dataset described in Sec. IV), we want to learn latent features that are similar when the category of the grasped object is the same. To push latent features to cluster together driven by object categories, we use metric learning. Metric learning optimises feature distributions with relative information between training samples: the distribution of latent features changes so that the features which are expected to produce similar outputs are also close to each other. Triplet training [34] was initially applied to face recognition [35]. We apply a triplet cost function inspired from [36] together with a pairwise term. The triplet loss function is computed from triplets, i.e. three instances of the same feed-forward network with shared weights [34]. Each triplet is composed of a reference sample, a positive sample and a negative sample. We use $z^{ref}$ to denote the feature of reference input $X^{ref}$ which is processed by function $f$.

In our case, $f$ is the concatenated architecture of joint estimator $p_j(p_h(q_h))$ and encoder $q_p$. $f(X^{pos})$ and $f(X^{neg})$ denote, respectively, the positive (same label as $X^{ref}$) and negative (different label as $X^{ref}$) anchors of the triplet. As metric for training we use the Euclidean distance. As an example, imagine a triplet with 3 hands holding, respectively, a bottle, a mug and a tomato. The reference sample is grasping a bottle. As hands have a similar configuration when grasping a bottle and a mug, and a significantly different one when holding a tomato, the hand with mug is labeled as positive sample while the hand holding the tomato is regarded as a negative sample.

During training, positive features $z^{pos}$ are those belonging to

the same category as the reference feature $z^{ref}$, while negative features $z^{neg}$ are those belonging to different ones. We set a loss function to make squared Euclidean metric $||z^{ref} - z^{neg}||_2^2$ larger than $||z^{ref} - z^{pos}||_2^2$:

$$\mathcal{L}_{triplet} = \sum \ln(\max(1, 2 - \frac{||z^{ref} - z^{neg}||_2^2}{||z^{ref} - z^{pos}||_2^2 + m})) + \sum ||z^{ref} - z^{pos}||_2^2 \ , \quad (5)$$

where $m$ is the margin for triplet. A pair-wise term $|z^{ref} - z^{pos}|^2$ is used for moving latent features on the boundaries of two categories, since the triplet term does not influence a lot for those samples. The dimensionality of both of the latent features $z_h$ and $z_p$ is 256. We choose it empirically through a grid search approach from $2^4$ to $2^{10}$ to balance the fitting capacity for data and time efficiency.

**Unsupervised Latent Feature Optimisation** In case the training data does not provide object category labels, we adopt an unsupervised clustering approach which could be used with SGD [32] aiming at forming clusters similar to those obtained via triplet training. Although K-means [33] clustering is simple enough to form clusters without using additional information, it can only be applied on a set of features without changing values. Since latent features are continuously updated during training, K-means would not converge easily. To solve this problem, we introduce $N$ additional latent variables $\{c_1, c_2, \ldots, c_N\}$ acting as cluster centers, having the same dimension as the latent features. Suppose that batch size is $M$, latent variables $\{z_1, z_2, \ldots, z_M\}$ are labeled with the index of its nearest center $c_m \in \{c_1, c_2, \ldots, c_N\}$ during training. The loss function $\mathcal{L}_{cluster}$ enforces those variables holding the same label to stay close to their cluster center. At the same time, the center itself moves towards the region of space where variables with the same label are present. The clustering loss function $\mathcal{L}_{cluster}$ is determined as below:

$$\mathcal{L}_{cluster} = \sum_{m=1}^{M} ||z_m - c_n||_2^2 \ . \quad (6)$$

The additional variables $\{c_1, c_2, \ldots, c_N\}$ are randomly initialized from a Gaussian distribution. Once this unsupervised clustering method is applied, triplet training still works when the indices of latent variables are almost fixed. Thus, the supervised and unsupervised loss functions can be used together for metric learning:

$$\mathcal{L}_{metric} = \mathcal{L}_{triplet} + \mathcal{L}_{cluster} \quad (7)$$

We use the squared Euclidean distance $\mathcal{L}_{pose} = ||Y_p - p_p(z_j)||_2^2$ between the expected canonical 3D pose and the output $Y_p$ of decoder $p_p$ as regression loss for the pose estimation. The overall loss function hence becomes:

$$\mathcal{L}_{3d} = \mathcal{L}_{triplet} + \mathcal{L}_{cluster} + \mathcal{L}_{pose} \quad (8)$$

Another advantage of enforcing such cluster centers is that the network can be trained with categorical supervision as cluster center of every category. It is then possible to exploit these centers at test time to infer the category of the grasped object, e.g. by comparing the distances between the latent feature and each cluster center. Since the predicted canonical poses are camera-view independent, we apply a rotation matrix $R$ to change $Y_p$ into a camera related pose, and we apply fully-connected layers with linear activation based on input $Y_j$. We adopt the architecture proposed by [15] by adding loss function $||R_{gt} - R||_2^2$ to $\mathcal{L}_{3d}$. In the end, the relative 3D poses are optimised with

$$\mathcal{L}_{3d_r} = \mathcal{L}_{pose} + ||R_{gt} - R||_2^2 + \mathcal{L}_{cluster} + \mathcal{L}_{triplet} \quad (9)$$

where the $\mathcal{L}_{triplet}$ term is optional and used only when object labels are available.


(a) Scenes (b) 21 joint (c) Objects (4 categories)

Fig. 4: Characteristics of the proposed HOP dataset

## IV. Hand-object Dataset for 3D Pose Estimation (HOP)

The datasets described in Sec. II-C are not sufficient for training an object-oriented hand pose estimation network due to the lack of the necessary variations of objects and shapes. We propose a new dataset dubbed Hand-Object Pose (HOP), which contains 11,820 pairs of RGB images and masks at $320 \times 320$ resolution, with 800 samples to be used for testing. Hand poses include 21 3D joints, which are manually created according to physical grasping pose and degrees of freedom (DoF) of each joint. Then they are used to precisely annotating CAD models. Images encompass 5 female and 5 male subjects who grasp 30 different objects with 600 randomly rendered background images. We assign category labels for each image based on the characteristics of the object present therein. The dataset includes 4 object categories: i) cylindrical objects (e.g., bottle, can, milk carton), ii) bars and sticks (e.g., pencil, fork, chopsticks), iii) cubical objects (e.g., book, smart phone, cutting board), iv) spherical objects (e.g., orange, ball, tomato).

**Dataset Generation**: With the help of MakeHuman [2], an open source computer graphics software for prototyping of photo realistic 3D avatars, we obtained 3D models with skeleton in different body shapes, skin colors and ages. 3D object models are obtained from *TurboSquid*[3] and human activities dataset [37]. We imitate physical human hand movement and manually create series of interaction animations between a human and an object using Blender[4]. Hand animations were captured from different viewpoints. Two point lamps randomly placed in each scene ensure the diversity of illumination conditions. After fixing the location of camera and light sources, background images, which exclude people or animals

[2] *http://www.makehumancommunity.org*
[3] *https://www.turbosquid.com*
[4] *https://www.blender.org*

in the scene, are selected from www.pexels.com. Mask images of both isolated hand and isolated object for each scene are generated by Blender as well, which may be useful for future research in object segmentation. We used Cycles Renderer[5], a physical-based unbiased path tracing engine designed for animations, to produce photo-realistic renders.

**Annotations**: As we placed a standing human model in the origin, 3D hand joint coordinates and object locations are automatically obtained according to the relative position of the center of the base of support (BoS). Object categories are manually defined. The dataset also provides intrinsic camera matrix and 3D keypoint positions in camera coordinate system as well. The synthetic dataset and the corresponding source code (python-blender) will be publicly released.

## V. Experiments

We evaluate our work on both bare hand datasets and our object manipulation dataset, via quantitative and qualitative results. For ablation purposes, we first independently test the 2D joint estimation and 3D pose estimation stages. When testing the 3D pose estimation stage alone, we feed ground-truth heatmaps as input. In addition, we also test the whole architecture composed of all proposed stages.

### A. 2D Hand Segmentation

As we are analysing our 2D processing architectures, only $\{q_h, p_h, p_j\}$ are optimised and tested. We compare performance in terms of completed hand segmentation on synthetic data using HandSegNet in [15] and our variational 2D hand estimator. We use 30000 samples from the rendered hand dataset (RHD) [15] and 10220 samples from the proposed HOP dataset together to train the 2D hand segmentation network. The network is randomly initialized and trained using ADAM [38] optimiser for 120,000 iterations. The learning rate is $1 \times 10^{-5}$ for the first 60,000 iterations, $1 \times 10^{-6}$ for the following 30,000 iterations and $1 \times 10^{-7}$ until the end. We quantitatively evaluate the performance of our methods on completed hand segmentation and joint estimation compared to HandSegNet and PoseNet in [15]. We show results in Fig. 5 under different challenging factors: (1) huge occlusion, (2) small occlusion, (3) complex background, (4) skin interference and (5) data migration from synthetic to real scenes.

The estimated 2D joints, hand masks, and estimated 3D joints (zoomed in) are shown in Fig. 5. Our approach overcomes occlusions from grasped objects better than [15]. One common problem of completed hand segmentation networks is skin interference, which means that such networks tends to segment other body parts out of the background instead of the hand. Fig. 5 also shows that our method performs much better even when face and hand overlaps. Fig. 5, bottom shows that our method works for real scenarios even when the parametric model is trained on synthetic data only, thanks to the realism of the proposed dataset (Sec. IV), as well as the capability of our inference model to handle the domain shift from synthetic to real data. The accuracy of the estimated hand masks proves that our variational training for the encoder-decoder architecture is effective, also avoiding to split the hand into several components around the object. One limitation highlighted by these results are the recurring blurred boundaries of the hand mask, which sometimes affects the segmentation of fingers.

[5] *https://www.cycles-renderer.org/*

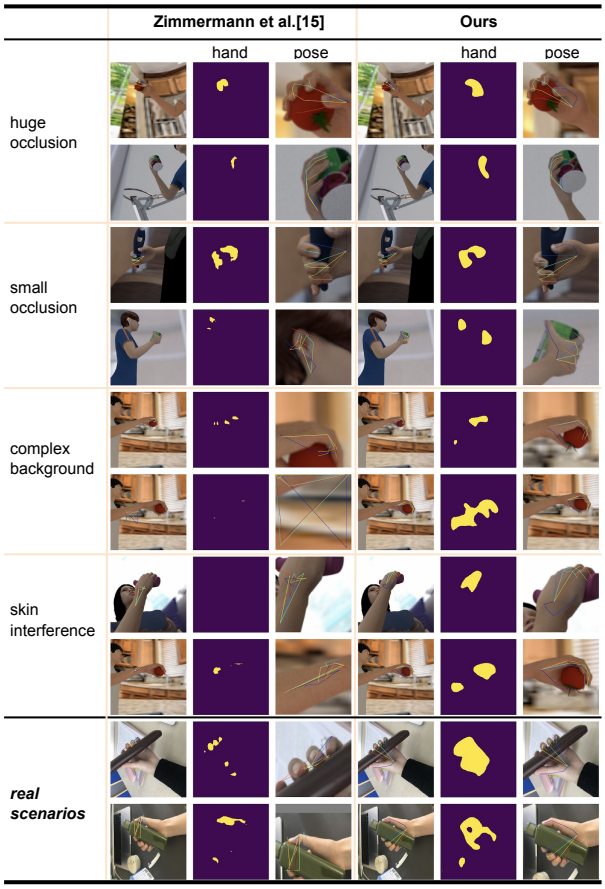



Fig. 5: Qualitative results of hand segmentation masks and poses of cropped hands using masks on HOP dataset.

TABLE I shows quantitative results for 2D joint estimation on the RHD [15] and the HOP datasets. We report the area under the curve (AUC) on the percentage of correct keypoints (PCK) with 20 pixels as threshold, the median value of endpoint error (EPE median) in pixels and the average endpoint error (EPE mean) in pixels. The Table shows how *Our-PoseNet* is effective on both HOP and RHD [15] datasets, achieving the best results with an AUC of, respectively, 0.775 and 0.704. The purpose of testing on RHD is to show that we also have good performance on bare hand pose estimation. In this case, triplet training is not adopted as there are no object labels, though we can still use the proposed variational 2D joint estimation architecture and the 3D pose estimator with the cluster loss function.

Compared to HS-PoseNet [15], our variational embedding yields a 0.032 improvement on AUC. If we use Hourglass [39], i.e. a deep architecture initially applied for human joint estimation, as a replacement for PoseNet in [15], the AUC increases to 0.741. Accordingly, we replaced PoseNet with

| data | method | AUC | EPE median | EPE mean |
|---|---|---|---|---|
| HOP (ours) | HS-PoseNet [15] | 0.722 | 4.285 | 13.392 |
| | HS-Hourglass [39] | 0.741 | 5.159 | 10.709 |
| | Our-PoseNet | 0.754 | 4.024 | 10.707 |
| | Our-Hourglass | **0.775** | **3.791** | **8.496** |
| RHD [15] | HS-PoseNet [15] | 0.635 | 6.745 | 18.741 |
| | Our-PoseNet | **0.704** | **4.215** | **17.520** |

TABLE I: 2D joint estimation by HS-PoseNet (i.e., HandSegNet + PoseNet as proposed in [15]) and by our approach (i.e., the proposed architecture $q_h - p_h$.) EPE are in pixels.

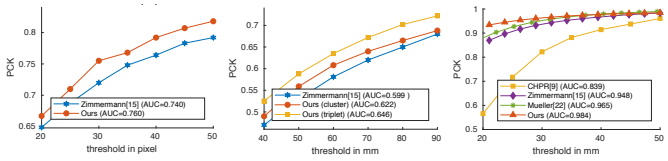
| method | cylinder | slender | cubical | sphere | *Avg.* |
|---|---|---|---|---|---|
| HS-PoseNet [15] | 0.694 | 0.773 | 0.767 | 0.701 | 0.734 |
| Ours-PoseNet | **0.743** | **0.812** | **0.828** | **0.721** | **0.776** |

TABLE II: Categorical joint estimation results on HOP dataset. HS-PoseNet is the joint HandSegNet + PoseNet architecture in [15]. Ours-PoseNet is the combination of $q_h, p_h$ architecture and PoseNet ($p_j$). The AUC is calculated over the error range in 0 to 20$mm$.

Hourglass in our architecture as well (i.e., *Our-Hourglass*), obtaining an increased accuracy of 0.775, hence proving once again the effectiveness of our variational approach. Although RHD does not include object occlusion, our method still performs better than HandSegNet [15], since there are still self-occlusions in the free hand case. Fig. 6(a) illustrates the AUC on PCK for which the error thresholds range from 20 pixels to 50 pixels. For the evaluation on the HOP dataset, we tested the performance on each of the 4 categories separately. Both HandSegNet [15] and our work are trained under the same conditions from scratch. Our approach performs better than HandSegNet [15] in each category, with a category-wise average of AUC 0.776, which is 0.032 higher than HandSegNet.

### *B. 3D Pose Estimation*

To compare the performance with a focus on 3D pose estimation only, we optimise only $\{q_j, p_p\}$ and feed ground-truth heatmaps as input, in order to unbias the comparison from the other stages of the pipeline. Given the correlation between the grasping pose and the object shape, first we train the proposed architecture $q_j$ and $p_p$ based on our HOP dataset with object labels. TABLE III shows AUC, median, and mean EPE for the RHD [15], GANeratedHands [22], Stereo [24] and our HOP datasets. For completeness, we tested both triplet training and unsupervised clustering. Since most available datasets lack object labels, we trained this variational model on HOP without labels. By learning the latent variables in an unsupervised way, we force the network to create hyper-clusters in the latent space. The number of clusters is not constant and is adjusted for each dataset. Specifically, we choose a number of clusters in the range of 4 - 10, eventually picking 5 for HOP, 7 for RHD [15], 10 for GANeratedHands [22] and Stereo [24]. Generally speaking, a high number of clusters tends to reduce the advantages brought in by clustering and metric learning, while quite small number of clusters tend to make the network a variational




(a) HOP (2D) (b) HOP (3D) (c) Stereo [24] (3D)

Fig. 6: Accuracy for 2D and 3D joint estimation as PCK over a threshold (pixels for 2D and mm for 3D) on the joint (RHD+HOP) dataset.

| data | method | AUC | EPE median | EPE mean |
|---|---|---|---|---|
| RHD [15] | PosePrior [15] | 0.555 | 18.932 | 28.804 |
| | Ours (cluster) | **0.587** | **16.301** | **27.569** |
| GANeratedHands [22] | PosePrior [15] | 0.977 | 7.665 | 8.790 |
| | Ours (cluster) | **0.981** | **7.197** | **8.243** |
| Stereo [24] | CHPR [9] | 0.839 | - | - |
| | PosePrior [15] | 0.948 | 9.543 | 11.064 |
| | GANeratedHands [22] | 0.965 | - | - |
| | Ours (cluster) | **0.984** | **7.606** | **8.943** |
| HOP (ours) | PosePrior [15] | 0.534 | 19.728 | 30.860 |
| | Ours (triplet) | **0.597** | **15.901** | **27.326** |
| | Ours (cluster) | 0.583 | 16.741 | 28.018 |

TABLE III: Results on RHD [15], GANeratedHands [22], Stereo [24] and our HOP with triplet training (triplet) and unsupervised clustering (cluster). EPE are in millimeters.

inference model. TABLE III shows the result of unsupervised training latent variables on those datasets respectively. The AUC ranges from 0.981 to 0.984 when we choose clusters between 4 and 10 in Stereo [24] dataset. Fig. 6(b) compares the PCK curve among PosePrior, our approach with clustering loss, and our approach with Triplet training, all trained on the HOP dataset. Combining the results shown in these tables, we can observe that (1) the proposed network efficiently improves the performance with respect to the original one, and (2) triplet training on latent variables using label information leads to better results than hyper clustering on latent space when labels are not available.

### *C. Comparison on the entire pipeline*

Finally, we compare the performance of the whole pipeline based on all parametric models $\{q_h, p_h, p_j, q_j, p_p\}$. We evaluate the results by training on HOP. We report the AUC on the PCK for 20 millimeters threshold, the EPE median in millimeters, and the EPE mean in millimeters. TABLE IV shows that the proposed approach with variational inference and metric learning has a good AUC of 0.580 which is 0.016 higher than previous work [15]. When we are training the whole architecture and only apply unsupervised clustering on the latent features from $q_j$, the AUC improves to 0.669, which is more than 0.100 higher than [15]. Note that both supervised and unsupervised learning exploit object categories. In supervised learning categories are given by the user as labels, while in the unsupervised approach, they are clustered automatically. If we have labels and the generated hand mask is accurate enough, supervised learning performs better. If the generated hand mask has low quality, it is better to use clustering to automatically infer object categories.

| method | AUC | EPE median | EPE mean |
|---|---|---|---|
| Zimmermann et al. [15] | 0.543 | 32.003 | 45.581 |
| Ours w/o variational | 0.564 | 30.193 | 44.466 |
| Ours (cluster) | **0.669** | **23.280** | **36.052** |
| Ours (triplet) | 0.580 | 29.485 | 41.012 |

TABLE IV: Evaluation on joint training with 2D ($q_h$,$p_h$,$p_j$) and 3D ($q_j$,$p_p$) processing with triplet training (triplet) and unsupervised clustering (cluster). AUC is calculated over error range from $0mm$ to $50mm$.

## VI. Conclusion and Future Work

Our proposed variational network with metric learning estimates the 3D hand pose while grasping an object from a single RGB image. We leverage the correlation between the hand pose and the category of the grasped object to design an effective architecture that does not require input 3D data. Since available datasets often do not include object category labels, a clustering method is introduced to group objects in an unsupervised fashion. Notably, in our approach the object category is the only information used for the objects. Both for supervised and unsupervised training, its validity is based on the assumption that hand poses for objects of the same category are similar. If a user grasps the same object in different and less natural ways, the performance of our architecture would decrease as this would invalidate such assumption. However, we believe that in many robotic applications (i.e., programming by demonstration) this assumption holds, since users typically train robots by grasping similar objects with similar hand configurations. An interesting future direction regards the use of contact information to correct the estimated 3D hand pose, guaranteeing consistency between pose and contact [40].

## References


[1] J.-H. Kim, N. D. Thang, and T.-S. Kim, "3-d hand motion tracking and gesture recognition using a data glove," in *Int. Symp. Industrial Electronics (ISIE)*. IEEE, 2009.

[2] P. Falco, G. De Maria, C. Natale, and S. Pirozzi, "Data fusion based on optical technology for observation of human manipulation," *Int. J. Optomechatronics*, vol. 6, no. 1, pp. 37–70, 2012.

[3] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient model-based 3d tracking of hand articulations using kinect." in *BMVC*, vol. 1, no. 2, 2011, p. 3.

[4] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, "Realtime and robust hand tracking from depth," in *CVPR*, 2014.

[5] A. Tagliasacchi, M. Schröder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly, "Robust articulated-icp for real-time hand tracking," in *Computer Graphics Forum*, 2015.

[6] M. Oberweger, P. Wohlhart, and V. Lepetit, "Training a feedback loop for hand pose estimation," in *ICCV*, 2015.

[7] Y. Cai, L. Ge, J. Cai, and J. Yuan, "Weakly-supervised 3d hand pose estimation from monocular rgb images," in *ECCV*, 2018.

[8] D. Tang, T.-H. Yu, and T.-K. Kim, "Real-time articulated hand pose estimation using semi-supervised transductive regression forests," in *ICCV*, 2013.

[9] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," in *CVPR*, 2015.

[10] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *TOG*, vol. 33, no. 5, p. 169, 2014.

[11] 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 *CVPR*, 2018, pp. 2636–2645.

[12] S. Li and D. Lee, "Point-to-pose voting based hand pose estimation using residual permutation equivariant layer," in *CVPR*, 2019, pp. 11 927–11 936.

[13] G. Moon, J. Yong Chang, and K. Mu Lee, "V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map," in *CVPR*, 2018.

[14] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3d orientation learning for 6d object detection from rgb images," in *ECCV*, 2018.

[15] C. Zimmermann and T. Brox, "Learning to estimate 3d hand pose from single rgb images," in *ICCV*, 2017, pp. 4903–4911.

[16] L. Ballan, A. Taneja, J. Gall, L. Van Gool, and M. Pollefeys, "Motion capture of hands in action using discriminative salient points," in *ECCV*. Springer, 2012.

[17] N. Kyriazis and A. Argyros, "Scalable 3d tracking of multiple interacting objects," in *CVPR*, 2014.

[18] J. Romero, H. Kjellström, C. H. Ek, and D. Kragic, "Non-parametric hand pose estimation with object context," *Image and Vision Computing*, vol. 31, no. 8, pp. 555–564, 2013.

[19] D. Goudie and A. Galata, "3d hand-object pose estimation from depth with convolutional neural networks," in *FG*. IEEE, 2017.

[20] C. Choi, S. H. Yoon, C.-N. Chen, and K. Ramani, "Robust hand pose estimation during the interaction with an unknown object," in *CVPR*, 2017.

[21] 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 *ICCV*, 2017.

[22] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "Ganerated hands for real-time 3d hand tracking from monocular rgb," in *CVPR*, 2018.

[23] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations," in *CVPR*, 2018.

[24] 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.

[25] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *ICCV*, 2015.

[26] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," *ICANN*, pp. 52–59, 2011.

[27] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[28] M.-N. Tran, D. J. Nott, and R. Kohn, "Variational bayes with intractable likelihood," *JCGS*, vol. 26, no. 4, pp. 873–882, 2017.

[29] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[30] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *CVPR*, 2016.

[31] A. Fossati, M. Dimitrijevic, V. Lepetit, and P. Fua, "From canonical poses to 3d motion capture using a single camera," *PAMI*, vol. 32, no. 7, pp. 1165–1181, 2010.

[32] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola, "Parallelized stochastic gradient descent," in *NIPS*, 2010.

[33] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," *TPAMI*, vol. 24, no. 7, pp. 881–892, Jul 2002.

[34] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International Workshop on Similarity-Based Pattern Recognition*. Springer, 2015, pp. 84–92.

[35] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *ECCV*, 2016.

[36] P. Wohlhart and V. Lepetit, "Learning descriptors for object recognition and 3d pose estimation," in *CVPR*, 2015.

[37] A. Pieropan, G. Salvi, K. Pauwels, and H. Kjellström, "A dataset of human manipulation actions," in *ICRA Workshop on Autonomous Grasping and Manipulation*, 2014.

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

[39] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *ECCV*. Springer, 2016.

[40] P. Falco, C. Natale, and R. Dillmann, "Ensuring kinetostatic consistency in observation of human manipulation," *RAS*, vol. 61, no. 5, pp. 545 – 553, 2013.