IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 1, 2020, accepted December 24, 2020, date of publication January 11, 2021, date of current version January 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3050624

Hand Pose Estimation in the Task of Egocentric Actions

MAREK HRÚZ[1,2], JAKUB KANIS[2], AND ZDENĚK KRŇOUL[1]

[1]Department of Cybernetics, Faculty of Applied Sciences, University of West Bohemia in Pilsen, 306 14 Pilsen, Czech Republic
[2]New Technologies for the Information Society, Faculty of Applied Sciences, University of West Bohemia in Pilsen, 30100 Pilsen, Czech Republic

Corresponding author: Zdeněk Krňoul (zdkrnoul@kky.zcu.cz)

This work was supported in part by the Ministry of Education of the Czech Republic under Project LTARF18017, and in part by the European Regional Development Fund through the AI&Reasoning Project under Grant CZ.02.1.01/0.0/0.0/15 003/0000466.

ABSTRACT In this article we tackle the problem of hand pose estimation when the hand is interacting with various objects from egocentric viewpoint. This entails a frequent occlusion of parts of the hand by the object and also self-occlusions of the hand. We use a Voxel-to-Voxel approach to obtain hypotheses of the hand joint locations, ensemble the hypotheses and use several post-processing strategies to improve on the results. We utilize models of prior hand pose in the form of Truncated Singular Value Decomposition (SVD) and the temporal context to produce refined hand joint locations. We present an ablation study of the methods to show the influence of individual features of the post-processing. With our method we were able to constitute state-of-the-art results on the HANDS19 Challenge: Task 2 - Depth-Based 3D Hand Pose Estimation while Interacting with Objects, with precision on unseen test data of 33.09 mm.

INDEX TERMS 3D convolutional neural network, egocentric, hand pose, TruncatedSVD, volumetric data.

I. INTRODUCTION

Devices capturing images or videos from first person (egocentric) view have recently become more common (e.g. Magic Leap One, Microsoft HoloLens, Google Glass). New smart applications can analyze such scenes, including hand gestures or poses, and help the user to improve his/her daily activities. Computer vision and machine learning are the main means how to process such data. They can extend the devices and applications with intelligent algorithms that understand what users are doing and how they interact with their surroundings [1]. In general, hand pose estimation is being addressed in many fields - robotics [2], medicine, automotive, or sign language processing [3]. Moreover, hand pose estimation presents beneficial cue for action recognition [4], [5]. The egocentric data has also impact on human machine interaction applications. Recently, these data have proven to be useful for learning to imitate man to robot [6], [7] or object detection and action recognition/ anticipation [8]. The problem of hand gesture detection and hand pose estimation is crucial for such interactive augmented or virtual reality applications. The human hand has a relatively large number of degrees of freedom with frequent self-occlusions of fingers hence the hand pose estimation leads to a highly nonlinear regression task. The task is even more challenging and still not solved for the egocentric view with more frequent occlusions of fingers caused by forearm and/or grasped object.

There are two main approaches for solving the problem of hand pose estimation. The first one is based on the synthesis of the pose from a known parametric model of a hand and its comparison with a given image of a hand. The goal of these so called top-down approaches is to find such values of the model parameters that describe the observed image the best. This approach is based on a handcrafted energy function for measuring the magnitude of difference between the synthetic and real image of the hand [9]–[14] or selected local [15]–[19] or holistic features [20] for discriminative estimation. The second approach called bottom-up is based on classification respectively regression of a given input hand image into a chosen representation of a hand model. These machine learning approaches are mostly based on deep neural networks (mainly convolutional) and in principle need a large amount of training data to achieve satisfactory precision [21]–[28].

These individual approaches can be further divided based on the contextual temporal information: methods for detecting/tracking the hand from a single image and methods

The associate editor coordinating the review of this manuscript and approving it for publication was You Yang.

that make use of several consecutive frames. The top-down approaches perform better when the contextual temporal information is available and the bottom-up approaches are well suited for the isolated frames. We use a bottom-up approach originally designed for the isolated frames and ensemble the results based on the temporal context.

The multiple learning algorithms aim at having a better predictive performance than the single learning algorithms alone [29], [30], if we assume significant diversity among the individual models. These are mostly known as an ensemble model researched on classification tasks. It is known that the number of learning algorithms affects the accuracy of prediction. Despite there are only a limited number of studies dealing with this problem. The file size and data volume in the classification task is mostly selected a priori and statistical testing is then used to determine the correct number of ensemble components. In this article, we explore the effect of ensemble components and ensemble cardinality in a regression task at several levels to increase the estimation accuracy of the egocentric hand pose estimation task.

Our main contributions include:

- Finding the optimal width of the hand pose estimating neural network.
- Utilizing the probabilistic output of the network by extracting several highly probable joint location hypotheses to produce a more precise location.
- Using several hand pose prior models in an ensemble fashion.
- Using temporal context to produce more robust hand pose estimations.
- Constituting state-of-the-art results in the HANDS19 Challenge: Task 2 - Depth-Based 3D Hand Pose Estimation while Interacting with Objects [31].

The rest of this article is divided as follows; Section II deals with related work in the field of hand pose estimation. In Section III we describe our approach to hand pose estimation. We divide the pipeline into several distinct parts; (a) hand region detection, (b) hand data representation, (c) hand joints localization, and (d) post-processing. In Section IV we report experiments on different aspects of our solution in a form of an ablation study. At last, we provide a Conclusion.

II. RELATED WORK

In this section, we focus on reviewing the related work on recent hand detection and pose estimation methods. We refer readers to [32], [33] for an overview of previous methods, mainly [32] for data-driven and [33] for generative methods, and will only deal with the more recent works, where majority of them is CNN based.

The hand region detection is usually the first step in hand pose estimation task. In [21] the authors train a randomized decision forest classifier that provides per-pixel hand/background segmentation. More recently, hand detection and pose estimation is designed as a single pipeline [34]–[36]. In [34] the 2D object detection method uses convolutional feature maps from the entire image and generates several bounding box candidates. In [35] the authors use localization network to detect hand object from one high-resolution depth image of the whole scene.

On the contrary in [5], there is no assumption of the bounding box, the image is divided to equidistant cells within hand pose estimation task. In [36] the authors introduce an approach where the hand location is estimated in a hierarchical way. The Hierarchical Hand Localization Networks and hand pose is presented as a joint framework that utilizes cascade processing from coarse to fine resolution. This hierarchical structure is first applied at a low-resolution octree of the whole image to produce a coarse hand region. Then a high-resolution octree is constructed on the region for fine location estimation.

Regression of 3D joint locations directly by a 3D CNN seems to benefit from learning 3D features in a single pass [32], [37]–[40]. In [37] the authors introduced V2V-PoseNet. They show that the hand pose estimation from depth data is inherently a 3D problem and should be handled in the 3D domain. At that time they achieved state-of-the-art results on several datasets including NYU [21] and won the Hands2017 Challenge[1] (HANDS2017) [41]. They used a voxel representation of both input and target data, and 3D fully convolutional neural network (FCNN) to estimate 3D locations of the joints. The input of FCNN is obtained straightforwardly from the depth data and the targets are represented as 3D Gaussian heat-maps with the mean value in the location of the target joint and a constant variance.

The encoder/decoder architecture of V2V-PoseNet is a hourglass shape similar to U-Net [42] or Segnet [43] with skip connections implemented as residual connections [44]. Regarding the negative side of this approach; although the network has a relatively small number of parameters ($\sim 2.5M$) the processing of the data is quite time and memory consuming since 3D feature maps are produced. Our method is inspired by these works and advance the research in the domain of egocentric view and huge occlusions due to grasped objects. The relevant factors are: number of parameters and hand pose prior.

There is an approach [39] that differs in absence of encoder/decoder structure in comparison to the original V2V-PoseNet architecture or ours. Instead of the encoder/decoder part, they use an additional loss function to steady the skeleton of the hand in a simple form of the ratio of each finger and the ratio of each bone's length. However, learning with such 3D hand pose constraint does not prove to be beneficial for regression task observed on the benchmark datasets (NYU, ICVL [16]).

On the other hand, leveraging the complete hand surface as intermediate supervision for learning was introduced in [32]. More specifically, authors extend deep framework with full hand surface estimation dealing in data-driven manner and

[1] http://icvl.ee.ic.ac.uk/hands17/

estimate complete hand surface. In experiments, they show improvement of the estimation accuracy of 3D hand pose with such intermediate hand surface completion step. Deep learning estimation of hand pose jointly with a full 3D hand triangular mesh representation is introduced in [45] for a single depth image.

There are 2D convolution approaches more convenient for RGB input images that recently perform well also for depth maps [46], [47]. By taking a step back to the 2D representation of the input depth data they address the computational complexity which is brought up to par with classical image recognition approaches. The main idea lies in the representation of the target. In [46], the network uses anchors evenly distributed in the input depth image to predict the offsets of the joints. There is a separate branch to handle the spatial offsets and another one to handle the depth offset. Furthermore, anchor informativeness is predicted to suppress the influence of irrelevant anchors on the position of a given joint. The authors show that when a pre-trained model is used as the backbone network (in their case ResNet-50 trained on ImageNet dataset) the system improves significantly.

In [47], the spatial structure of depth map is modeled by a 2D CNN as differentiable re-parameterization module to construct spatial-aware representations from joint coordinates directly. 3D offsets of the pixels relative to the hand joints coordinates are calculated as 3D heat maps and unit vector fields. It reflects the closeness and directions from pixel to the target joints, respectively. The estimator reaches state-of-the-art performance for NYU.

More recent approaches to hand pose estimation from a single depth map decomposes the task into several sub-tasks (for palm and fingers) [48], [49]. In [48], the authors adopt two-branch cross-connection structure to share the beneficial *complementary information* between the sub-tasks. Authors of [36] are also using 2D convolution method to perform fine location estimation based on DeconvNets [50]. However they did not outperform 3D (point cloud or voxel) based methods.

The third category of the approaches can be classified between 2D and 3D approaches. The methods directly take the 3D point cloud as input to infer 3D hand joint locations [51]–[53]. Point-to-Point regression - PointNet [51] models 3D spatial information in the point cloud and outputs point-wise estimations in the form of heat-maps and unit vector fields defined on the input point cloud. Thereby it defines the closeness and the direction of every point in the point cloud to the hand joint. The PointNet uses a stacked network architecture trained in an end-to-end fashion. To model the hand pose, the point-wise estimations are used with the weighted fusion. In work [52], deep learning based hand pose estimation method from unordered point cloud is used. Different from them, our method aims on egocentric view.

In [54]–[56], the focus is to observe the role of synthetic data in the task of hand pose estimation. Supported by experiments, it is shown, that the synthetic data enable the models to generalize better to real-world test data. In [55], the authors propose a self-supervision method for learning the 3D hand pose from an unlabeled depth map. The method is initialized by synthetic data in a supervised manner and fine-tuned on real depth maps in unsupervised manner. This shows benefits for multiple view scenarios, for example the front and the side view of the hand. However, it cannot be simply applied to the task of egocentric hand pose and single view estimation.

On the other hand, the synthetic data generated for the egocentric view were used to train a CNN in [57]. They combine synthetic data with a generative hand model to track hands interacting with objects from RGB-D videos. Moreover, in [54], they used the synthetic data to learn 3D meshes of objects grasped by the hand. Very beneficial *feature mapping* for learning 3D hand pose from synthetic depth maps was introduced in [56]. They used 5M synthetic images of randomly generated hand poses that are rendered online during training and currently it has still the best performance on NYU. They train a regression network that utilizes a mapping of features of the real images into the synthetic feature space domain and input this mapping to the network predicting the 3D pose from image features.

The prediction of an estimate of the 3D hand pose interacting with an object was introduced in [5], [33], [54]. In [5], the authors propose an approach for predicting simultaneously the 3D hand and the 6D object pose, object classes and action categories from image sequences of an egocentric RGB camera. In this joint learning scenario, they model the hand pose and confidence of the joint location in the sub-regions of the image and use an RNN mapping to learn the explicit dependencies between the hand and object poses. They demonstrate state-of-the-art performance of the algorithm on First-Person Hand Action (F-PHAB) dataset [4]. However, the contribution of such complex model on RGB-D input for better precision of the hand pose estimation was not proved. A different approach is in [54], where the authors introduced a method jointly estimating the hand and object 3D meshes to be able to infer interactions with objects. In [33], the feedback loop was used for training a CNN. They learn the CNN to generate images of the hand as a feedback by first synthesizing depth images of the hand and the object, and then merge them together. In [47], the authors take the spatial-aware representations as intermediate features, stack multiple regression modules to repeatedly predict joint coordinates, which allows the estimator to infer the 3D spatial structure of depth data and reevaluate the initial estimations using the 3D information and multi-joint spatial context.

The principal component analysis (PCA) is used for the hand pose and shape regularization in a training phase [58] or as a post-processing step [51]. In [38] the authors extended V2V-PoseNet architecture to a structure-aware network to model skeleton constraints of the hand pose treated as an intermediate supervision on hand bones. The authors incorporate skeleton constraints of hand pose into detection network with encoder/decoder structure without considering an explicit post-processing step. Moreover the authors introduced a method of combining more best locations from the

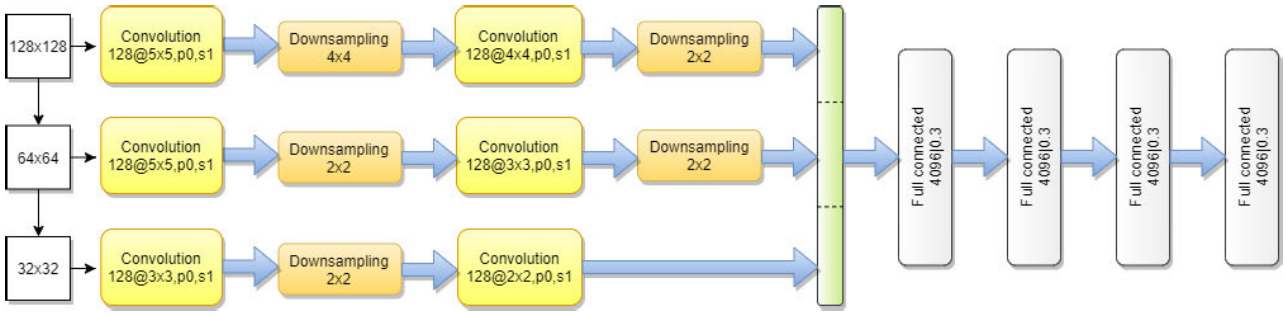


FIGURE 1. Network architecture for the hand region refinement. The input consists of three depth images in different scales. Each branch consists of two blocks of convolutional layers followed by a max pooling. Final feature maps are flattened and concatenated into a vector representation. The vector is processed by four fully connected layers and one final layer with three neurons producing the u_x, u_y, d offsets.

heat-map to obtain the sub-voxel precision of the position of the joint.

In this work we expand on this method by providing an ablation study on the number of locations and their utilization. Moreover, we perform the experiments on much more benchmark datasets and show the universality of our approach. Our approach differs from previous works in *cascaded-linear embedding*. We introduce TruncatedSVD hand pose model and combine it with the ensembles of multiple heat-map hypotheses, N-best joint locations from single heat-maps voting. Moreover our approach can integrate time smoothing over the egocentric action. Unlike [40], the temporal context is used only to cope with a low confident hand pose (or a particular joint) estimation primarily caused by their occlusion with the object. In contrast to [37], [38], our approach benefits from the integration of a single detector of the original V2V-PoseNet in a multiple learning framework with post-processing steps. To the best of our knowledge, we are the first ones to apply post-processing methods of a hand pose prior and temporal context modeled as an ensemble of TruncatedSVDs to the problem of hand pose estimation from egocentric viewpoint when interacting with objects.

III. METHODS

Our approach is based on machine learning techniques, namely deep learning. We adopt the approach introduced in [37] as the Voxel-to-Voxel PoseNet and present an ablation study on a recent benchmark dataset HANDS19 Challenge: Task 2 - Depth-Based 3D Hand Pose Estimation while Interacting with Objects [31]. We believe this dataset to be the most challenging depth based egocentric view hand pose estimation dataset available to this date.

Our system can be divided into several parts:

1) hand region detection;
2) hand data representation;
3) hand joints location estimation;
4) post-processing to obtain the refined final hand pose.

We look into all the individual aspects of the system and analyze the impact of different methods and their parameters.

A. HAND REGION DETECTION

In this work, we follow [4] and initialize our hand region detection method from rough a priori 2D regions provided with the data. In our system, we use a hand region detection as a first step in the processing chain. We expect an approximate region of the depth map in which the hand lies. We use a region refinement network similarly to [58], however we provide a more robust network with three different scales of the input depth image to be processed by three independent branches as depicted in Figure 1. This idea was put forward by Thompson *et al.* in [21] as a multi-resolution CNN for the hand pose estimation. Thus, we consider a rough initialization and each branch observes the hand at a different resolution and hence can learn features at different scales. Unlike [58], we cannot rely on thresholding of the depth image, since we are considering egocentric view of the hand.

In our implementation of the region refinement network, we use blocks consisting of a convolution layer, max pooling, and batch normalization [59]. The resulting feature maps are concatenated channel-wise and flattened into a vector representation. This vector is then processed by fully connected layers and finally the offsets are produced. In our case the offsets are in the domain of the depth map - specifically u_x, u_y, d. Here, the vector (u_x, u_y) is in the metric of pixels and d is in the metric of millimeters. Using the offsets we determine the center of the hand region in the image domain and also the expected depth of the hand.

B. HAND DATA REPRESENTATION

The hand is represented by a set of voxels in a 3D cube (see Figure 2). The cube is defined in the metric of millimeters. We fit this cube to the approximated hand region center and given the known camera parameters we compute the projection of this cube into the depth image. This allows us to crop the depth image in both the image and the depth coordinates. Next, we need to represents the observed depth pixels in the coordinate system defined by the 3D cube. This is again achieved by using the known camera parameters. Finally, the coordinates are discretized into an $N \times N \times N$ grid, so that we obtain a Boolean representation of the 3D cube. In literature this process is referred to as *voxelization* [37].

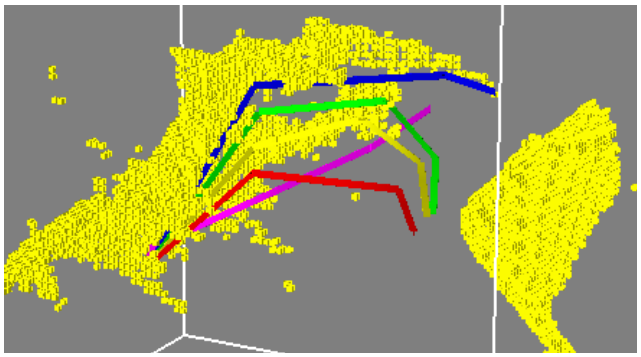


FIGURE 2. Visualization of one frame of the data converted to voxels (yellow points) with target labels plotted as a skeleton.

C. HAND JOINTS LOCATION ESTIMATION

To estimate the locations of hand joints we use variants of the model proposed in [37]. The voxelized representation of the hand is inputted into a 3D FCNN with an hourglass structure (see Figure 5). At the output there are 3D heat-maps representing the likelihood of the joint being located in the particular voxel. The likelihood is modeled as an unnormalized Gaussian. For each joint there is one output heat-map. From the estimated heat-maps, we are able to compute the location of the joints in the coordinate system of the output 3D cube. Then in combination with the refined position of the hand region and known camera parameters, we are able to transform the results into the target millimeter metric. The absolute location in the metric of millimeters is used in evaluation. If needed, we are able to transform the locations into the depth image domain. We achieve this by using the known location of the crop from which the 3D cube was obtained and the known camera parameters to project the 3D coordinates into the image domain.

Expanding on the work [37], we explore three different algorithms to estimate the location of the joints from the output heat-maps. Firstly, we estimate the location as a weighted average of all the voxels of the output heat-map. This can be seen as a maximum likelihood estimation when we try to reconstruct the original target Gaussian for a given joint as:

$$\vec{x}_j = \frac{1}{\sum \omega_i} \sum_{x=1}^{N} \sum_{y=1}^{N} \sum_{z=1}^{N} \omega_i \, [x, y, z]^T, \tag{1}$$

where $\vec{x}_j$ is the location of the j^{th} joint defined in the coordinate system of the output 3D cube (with sub-voxel precision), ω_i is the estimated likelihood of a joint being located in the i^{th} voxel (as a linear index), and x, y, z is the location in the output 3D cube of size $N \times N \times N$.

Secondly, we use a straightforward approach of estimating the joint location as the maximum likelihood in the output 3D cube:

$$\vec{x}_j = \underset{x,y,z}{\operatorname{argmax}} \, \omega_{(x,y,x)}, \tag{2}$$

where $\omega_{(x,y,x)}$ is the estimated likelihood of a joint being located in the position x, y, z. This approach is limiting in the sense that it can produce only a discretized value of the location. The discretization is given by the number of voxels in the output 3D cube and hence the precision of the result is dependent on the size of the cube in millimeters.

Thirdly, we use a smoothed version of the argmax approach. We determine the $\vec{x}_j$ as in Equation 2, denoting it $\hat{\vec{x}}_j$. Next, we compute the final location as a weighted average of a $k \times k \times k$ cube centered on the voxel $\hat{\vec{x}}_j$.

$$\vec{x}_j = \hat{\vec{x}}_j + \frac{1}{\sum \omega} \sum_{x \in \varphi} \sum_{y \in \varphi} \sum_{z \in \varphi} \omega_{(x,y,z)+\hat{\vec{x}}_j} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{3}$$

where $\sum \omega$ is the sum of likelihoods in the $k \times k \times k$ cube, over which the individual summations are performed, and $\varphi = \left\langle -\frac{k-1}{2}; \frac{k-1}{2} \right\rangle \in \mathbb{N}$, implying that k must be odd.

D. POST-PROCESSING TO OBTAIN THE FINAL HAND POSE

To refine the final position of the joint, we adopt several post-processing methods.

1) EPOCH ENSEMBLE

The method takes the outputs of the network from individual epochs and combines them into a final decision [37]. By our system, we explore two types of combinations. We either combine the heat-maps or the computed joint locations. The combination is performed as averaging. This technique is not yet clearly understood, since intuitively the model from a new epoch should outperform the model from a prior epoch. In the context of the problem of hand pose estimation and our solution, we take into consideration several assumptions. (1) On the large dataset, the network's performance is quite reasonable even after the first epoch and it allows to use all the epochs in the final decision. (2) After several epochs the training loss of the network starts to fluctuate. We assume that the network is adapting to different subset of the training data after each epoch and in a joint decision the final accuracy can improve.

2) PREDICTION ENSEMBLE

This method utilizes the form of the predictions that the network makes. Since it produces likelihood heat-maps, we are able to take N best results and combine them into the final decision. The reasoning behind this is that we want to make use of the whole heat-map and not just a single point in it. We assume that the decision of the network can be flat Gaussian or Gaussian mixture. The total maximal likelihood can be mainly for occluded joints only slightly better than the next best, and so on. Thus the final joint location is obtained as an average of N best locations in a given heat-map.

3) POSE PRIOR

Given that the hand pose is restricted by a relatively low number of DoF of individual joints, we hypothesize that a model of prior hand pose should be beneficial for the final accuracy of the system. The hand model we use in our

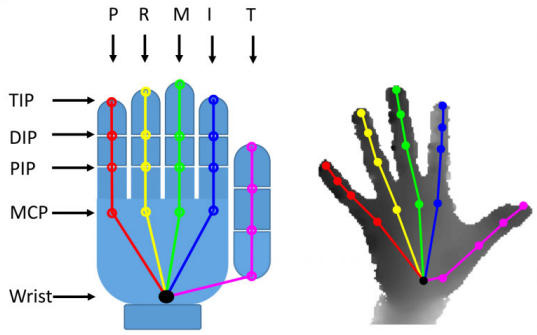


FIGURE 3. Hand model definition [62], on the left: T, I, M, R, P denote Thumb, Index, Middle, Ring, Pinky fingers. MCP, PIP, DIP, TIP denote Metacarpal, Proximal, Distal bones and Tip of the finger; on the right: the depth map with the ground truth annotation.

work has 21 joints as seen in Figure 3. The theoretically maximal number of DoF is 21 joints × 3 spacial coordinates which equals to 63. In accordance with [60], the hand pose is anatomically restricted just to 27 DoFs. This discrepancy encourages the usage of dimension reduction methods.

Although these methods are well established in the field, we are the first ones to provide a comprehensive study of their effect on hand pose estimation problem from egocentric viewpoint.

When dealing with hand pose data representing object handling, the number of possible hand poses is further restricted by the shape of the objects. When the object occludes the hand, the hand shape prior further helps to determine the true hand pose, i.e. to estimate the location of unseen joints. In this approach we used TruncatedSVD [61], which is a variant of singular value decomposition (SVD) that only computes the k largest singular values, where k is user specified. Unlike PCA it uses the data matrix directly as opposed to computing the covariance matrix.

The pose prior is computed using the ground truth data (GT). Specifically, we use the target locations of joints in millimeters expressed in the coordinate system of the input 3D cube stacked into vectors to train the TruncatedSVD model. The anatomical and/or object restrictions of the hand pose are implicitly encoded in the target data and the SVD model should reflect this. We train several models with different number of components and use them in post-processing by computing the latent representation of the 63 dimensional pose vector obtained from the output heat-maps.

4) TEMPORAL CONTEXT

Temporal context can be applied to sequential data. The assumption is that the hand pose will not change dramatically on a frame-to-frame basis. This is especially true when dealing with sequences of object handling. We can then use the predictions from different time instances to augment the hand pose in the middle time instance. We compute the pose as a weighted average of individual joint locations. The size of the context can be chosen. We use a window of n frames before and after the current frame.

IV. EXPERIMENTS

In this section we provide details about the conducted experiments and implementation. The task is to predict the locations of hand joints as accurately as possible. The evaluation is reported as the average error per joint in millimeters. The network weights and biases are initialized by random numbers from the normal distribution with zero mean and variance of 0.001. We use RMSProp as the optimizer with learning rate of 0.00025. The batch size is dependent on the size of the network and the memory capacity of GPUs we used for the training (NVIDIA GeForce GTX1080 Ti 11GB). The experiments are coded in Python with Chainer [63] as the framework for the neural network computing.

A. DATA

Given the relative high difficulty of labeling hands in 3D, there are not many datasets available for training large models of machine learning. There are some well established datasets from 3rd person's view, but there is a lack of well described datasets from first-person view with object interaction. Actually, the first large benchmark dataset (F-PHAB) that enables the study of the first-person hand actions with the use of full 3D hand poses was introduced [4] recently. This dataset includes dynamic hand action sequences with more than 100k RGB-D frames annotated with 3D hand poses, see Figure 4, using six magnetic sensors attached to the fingertips and inverse kinematics to obtain the ground-truth data.

We train and evaluate our system on the data provided in the HANDS19 Challenge: Task 2 - Depth-Based 3D Hand Pose Estimation while Interacting with Objects [31], [64]. This task builds on the F-PHAB dataset [4], where objects are being manipulated by a subject in an egocentric viewpoint, see Figure 4. Some hand shapes and objects are strategically excluded from the training set in order to measure interpolation and extrapolation capabilities of solutions. Training set contains images from four different subjects performing 45 different actions involving 26 different objects. The test set contains images from four different subjects performing 71 different actions involving 37 different objects. Forty-five actions, 26 objects and two users overlap with the training set. Some frames appear in the the original F-PHAB dataset release and some are unreleased (new) frames. The downside of this dataset is that there are no official validation data available.

The data are provided with approximate 2D hand regions. We apply refinement of this region as described in Section III-A. The depth image crop of constant size 200 × 200 pixels is centered on the provided rough region enclosing the hand object and is extracted as the input for the refinement network. We keep this size of the region constant over all experiments.

We empirically verified that the region is sufficient enough for resolution and focal length of used Intel RealSense

FIGURE 4. Example of the training and testing data, the illustration taken from [64]. They consist of egocentric view on hands interacting with various objects. The depth data are accompanied with ground truth labels obtained from magnetic sensors.

SR300 depth sensor and the egocentric capture scenario. For our experiments, we assume a minimal distance of 475 mm from the depth sensor, in which case 200 pixels capture all hand shapes of approximate size 200 mm and smaller. Next, we assume maximal distance of 600 mm for hand poses captured in situations when an adult performer's arm can be fully stretched forward. The constant hand region is resized to three scales of (1) 128 × 128 px, (2) 64 × 64 px, and (3) 32 × 32 px, see Figure 1.

The position of this region is refined by the network. The offsets resulting by the network are scaled back up to the original resolution and applied to the original center of the hand region. In practice this means that the constant 200 × 200 px hand region is shifted to a new position. This region serves as the basis for the process of voxelization (Section III-B). In [32] considering both the estimation accuracy and the real-time performance, for the experiments, they used 32 × 32 × 32 volume as the input resolution. However in our study we use a 3D cube of the original size 88 × 88 × 88 voxels since we optimize the precision. All depth pixel appear as "1" in the cube and background is "0".

B. HAND REGION REFINEMENT NETWORK

The training data for the hand region refinement network consist of cropped depth image, the approximate 2D hand region, and a point in the depth image domain representing an explainable/constant point on the observed hand. The point is obtained from 3D target joint locations. We experiment with two options. First, we use the center of mass of the target labels (CoM) as the target point and second, we use the location of the root of the middle finger (RoM) as the target point (see Figure 3, the point M-MCP). Since the joint locations are in the millimeter metric, we need to transform them into the image domain.

The architecture of the refinement network is depicted in Figure 1. The high resolution branch (128 × 128 px) begins with a convolutional layer with 128 kernels of size 5 × 5 with a stride of 1. A max-pooling layer of size 4 × 4 with a stride of 4 follows. A Batch Normalization layer is applied after max-pooling. These three layers form one block, and several blocks comprise the network. The outputs of the three convolutional branches are concatenated and flattened. Four fully connected layers with 4096 neurons follow and the fifth fully connected layer has three neurons that output the offsets: (u_x, u_y, d).

TABLE 1. Average joint error [mm] of the refinement network on training and validation data. RoM is the Root of the Middle finger, CoM is the Center of Mass.

Method	error on train data	error on val. data
RoM	4.51 mm	4.85 mm
CoM	5.64 mm	5.76 mm

For this learning task, we kept aside $> 5k$ depth images as validation data. We train the refinement network for 100 epochs and select the epoch with the best performance on the validation data. We use a batch size of 64, Adam optimizer [65] with beta 0.9, weight decay 10^{-5}, and learning rate 5×10^{-5}. The employed data augmentation scheme includes following transformations: (1) in-plain rotation uniformly and randomly drawn from an interval $\langle -35, 35 \rangle$ degrees, (2) scale drawn from a distribution $\mathcal{N}(1.0, 0.04)$, and (3) translation drawn from a distribution $\mathcal{N}(0, 25)$ pixels. The augmentation is performed with a probability of 0.3 for each data sample and each individual transformation is performed with probability 0.5. The results can be seen in Table 1. The network with the RoM target is able to train with a higher precision. In the next experiments we also analyze the impact of the two approaches of hand region refinement on the final precision of the joint location estimation.

C. DATA AUGMENTATION

A data augmentation is a well established method for reducing overfitting of deep models. In our system we use several geometric transformations to augment the training data for the joint location estimation. Namely translation, rotation, and scale. The augmentations are performed on-the-fly directly on the volumetric data. The translation is randomly chosen from an interval of $[-8; 8]$ voxels for each (x, y, z) dimension independently. The translation augmentation should cover for the mistakes done by the hand region refinement network.

The rotation is preformed only in the image plain (represented by the xy plane) around the center of the 3D voxel cube. Other rotation axes would create unrealistic/not possible data since the depth sensor observes only unoccluded surfaces due to the projection. We use a linear interpolation to minimize the loss of voxels. The angle of rotation is drawn from an interval of $[-40; 40]$ degrees.

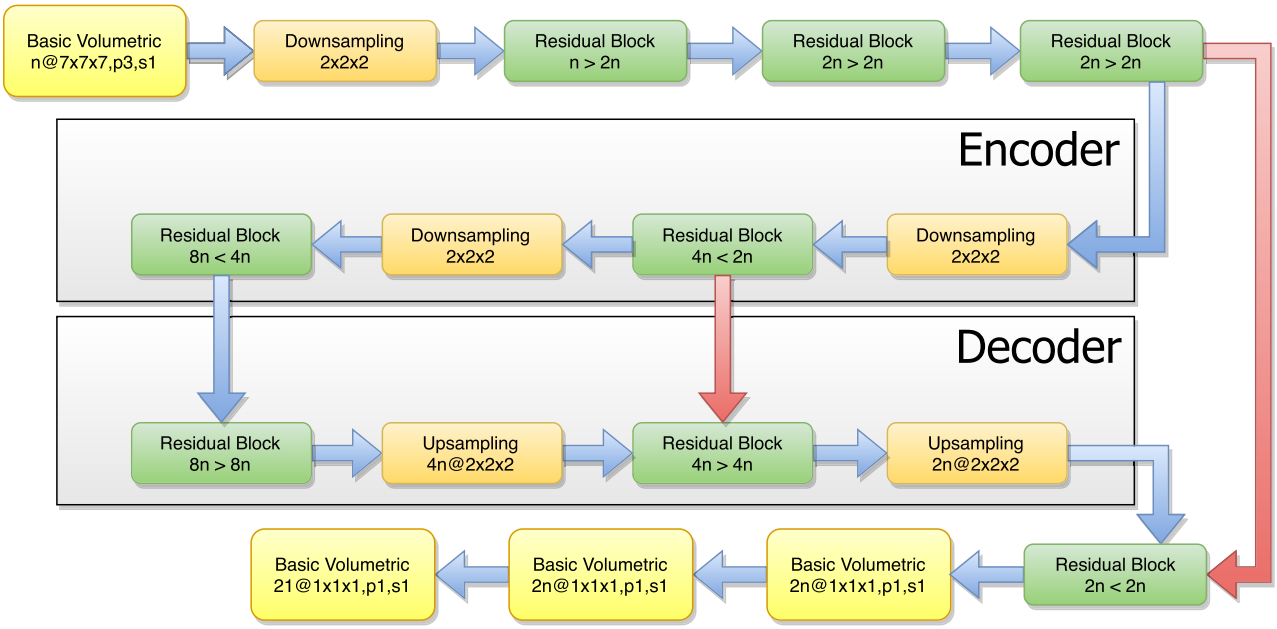


FIGURE 5. The V2V-PoseNet-like architecture used in this work. Basic Volumetric Block is a 3D convolutional layer with notation *number_of_channels@kernel_size,padding,stride* followed by Batch Normalization. The Down-sampling Block is a 3D Max-pooling operation with given size. The Residual Block has a notation of *input_number_of_channels* > *output_number_of_channels*. The Up-sampling Block is implemented as Transposed Convolution (Deconvolution) with notation *number_of_channels@kernel_size*. In the original work [37], *n* was set to 16.

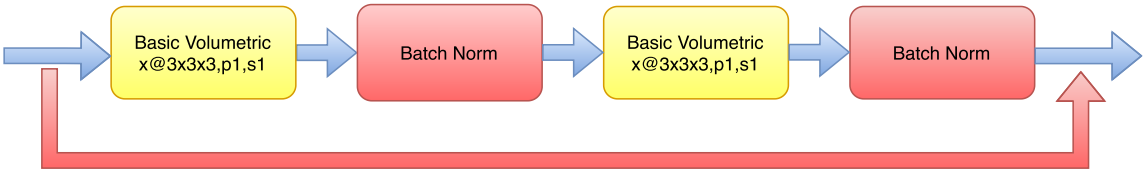


FIGURE 6. The Residual Block of the network. Here the parameter *x* is the number of output channels. The skip connection is summed element-wise with the output. If the input and output number of channels do not agree, we use a 1 × 1 × 1 convolutional layer to re-project the input into the output space.

The scale is selected randomly from an interval of [0.8; 1.2] for each axis independently. Hence, the transformation can actually change the shape of the hand to help with the hand shape extrapolation. Again, we use the linear interpolation, so that after the scaling up we do not create holes in the volumetric data. The transformations are applied independently with a probability of 0.5. The order of transformation is (1) scale, (2) rotation, and (3) translation so that we do not transform the voxels out of bounds too often. The same transformations are applied to the target locations.

D. NETWORK ARCHITECTURE

In this section we explore different architectures of the neural network. We focus on the width parameter of the model, which is essentially the number of kernels in the convolutional layers. The base model introduced in [37] can be seen in Figure 5. The input is of size 88 × 88 × 88 and the output is of size 44 × 44 × 44. The Basic Volumetric block is a 3D convolutional layer followed by Batch Normalization. Down-sampling block is a 3D max-pooling layer, Residual block is depicted in Figure 6. It consists of a Basic Volumetric block, followed by the batch normalization, repeated two times. The input is added element-wise to the output. If the number of channels of the input data is inconsistent with the number of channels of the output, the input data are re-projected using a 1 × 1 × 1 convolutional layer with the appropriate number of channels. The Upsampling block is realized as a Deconvolution (Transposed Convolution) layer [50]. The skip connections are summed element-wise with the other input to the layer.

The width parameter is noted as *n* in the Figure 5. It represents the number of channels in the first convolutional layer. We can see that the model has eight times more channels in the bottleneck as in the input. We keep the ratio of the number of channels constant throughout the experiments and we change the base number of channels *n*.

In Table 2 we can see the results of the basic model with different widths. We fixed CoM as the basis for the process of voxelization and S-max for location estimation (Equation 3). The best performing model has width of 64 (4x

TABLE 2. Impact of the width of the network on the final precision on the test data. With CoM refinement and S-max location estimation.

Width n	Epoch	Precision [mm]
16	10	40.04
32	10	38.18
64	10	38.05
84	10	71.99
16	20	39.61
32	20	38.03
64	**20**	**37.11**
84	20	70.34

wider). The difference is even more notable after 20 epochs, which suggests that the learning capacity of this model is higher than of the thinner ones. The trend in the downward direction remains the same. The model with the width parameter of 16 performs worse than the wider models. After increasing the width parameter 6-fold to 84 the model drops in accuracy dramatically, even though the training loss still remains low. This can be explained either by a classical view, that the model is overfitted and lost the ability to generalize, or according to new studies [66] the model has reached an interpolation threshold in the critical regime. Unfortunately the computational capacities of our facilities do not allow us to explore this hypothesis further by increasing the width parameter and hence no definite conclusion can be made at this point.

E. POST-PROCESSING

In this section we report the results of our experiments comparing different approaches to the post-processing.

1) HAND JOINTS LOCATION ESTIMATION

In this experiment we compare the impact of the different types of joint location estimation from the resulting heat-maps. We fixed the width of the network to $n = 16$. The size of the neighbourhood for the smooth max computation (Equation 3) was set to $k = 3$. The results are reported in Table 3. We observe that the reconstruction of the original Gaussian yields the best results. This is exclusive to the egocentric viewpoint of hands handling objects. In other task (i.e. 3^{rd} person viewpoint, unobstructed hand) the best results were obtained using the S-max function (Equation 3). This can be explained by the frequent occlusions of hand joints by the objects. When the joints are not directly observed the Gaussian approximation gives the lowest error on average. The downside of using this approximation is that only one prediction per heat-map is obtainable. Hence, no other ensemble from the heat-map is possible.

TABLE 3. Impact of the refinement network and location estimation on the final precision. Refinement methods consist of RoM, i.e. the target is the Root of the Middle Finger, and CoM, i.e. the target is the Center of Mass of the joint locations. The location estimation is Gauss, i.e. Equation 1, Max, i.e. Equation 2, S-max (smooth-max), i.e. Equation 3. The precision is in millimeters.

Refinement	Epoch	Max	S-max	Gauss
RoM	10	42.78	42.70	41.19
CoM	10	40.13	40.04	38.97
RoM	20	42.05	41.97	39.86
CoM	20	39.69	39.61	**38.08**

2) HAND REGION REFINEMENT

In Table 3 we can also see the impact of the refinement network on the final precision of the system. Although we were able to train the network predicting RoM with higher precision, see Table 1, the joint location estimation is more precise when using the refinement network trained on CoM. The results are consistent through out the epochs and the method of location estimation from heat-maps. It is not completely clear why this behavior occurs. We observe that in some cases the root of the middle finger is in an extreme position relative to the other hand joints which can result in cutting of vital information when performing the process of voxelization. Simply put, some depth pixels of the hand do not fit into the 3D cube when it is centered on the RoM. A safer option is to estimate the CoM of the joints, to minimize this cut off. Finally, if the z coordinate of the CoM is higher than the maximal allowed distance, see Section IV-A, we set it to an experimentally determined value of 130 mm.

3) PREDICTION ENSEMBLE

In the next experiment we wanted to utilize the properties of the heat-maps which represent the likelihood function of the 3D position of joints. In Table 4 we measure the precision of a system that was trained for 20 epochs and (1) uses only one best location, (2) uses 100 best locations, (3) ensembles 100 best locations from each epoch, and (4) ensembles 100 best locations from a subset of epochs. It needs to be noted that when we apply this technique, the Gaussian estimation of location from heat-maps (Equation 1) is no longer viable and furthermore, it is outperformed. Hence, in the following experiments we use the S-max approach (Equation 3) to obtain the sub-voxel locations of joints.

When we use more locations to estimate the final prediction we use a simple averaging of these sub-voxel locations. Experiments that used the confidence from the heat-map in a weighted averaging scheme resulted in a slightly worse precision. At first glance it is clear that when an epoch ensemble is used the performance improves. This is consistent with the findings in [37]. It hints at an assumption that in each epoch the network captures different aspects of the problem and by averaging the results we are able to obtain the best precision.

TABLE 4. Results for different methods of prediction ensemble on the test data.

Epoch Ensemble	# N-best	Precision [mm]
20	-	37.17
20	100	36.14
1-20	100	35.18
3-best	100	34.88
5-best	100	**34.85**
7-best	100	34.90

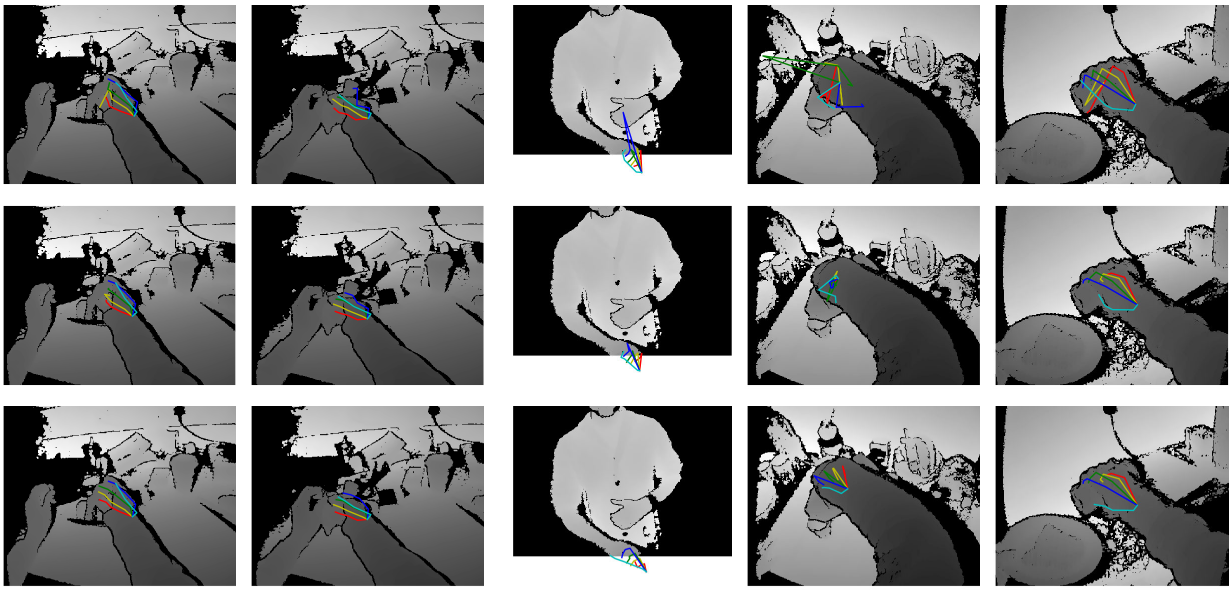


FIGURE 7. Example of the testing data prediction. The first row represents the raw output of the pose network obtained by the S-max approach (best epoch on the test set). The second row represents the effect of the ensemble (5 best epochs, N-best locations). The third row represents the ensemble + post-processing effect (TruncatedSVD pose prior, temporal context).

As result, the best performing system uses five epochs that achieved the lowest error on the test data. We did not explore the performance after each epoch due to none available validation data in the benchmark dataset. We did not want to cut off the validation data from training data, so that we do not put ourselves at a disadvantage in the challenge. Minimizing the number of epochs to ensemble is good for shortening the run-time. However, when using three best epochs the results are less precise. On the other hand, increasing the number of epochs does not lead to better results as demonstrated when using 7 best epochs.

Next, we want to see what is the optimal number of the best locations to use for the averaging. In Table 5 we see that by raising the number of location candidates we obtain better results. Although the differences are very subtle and by further raising the number they get smaller. For the rest of the experiments we fixed the number of candidate locations to 100.

TABLE 5. Results for different number of best locations used in ensemble on the test data.

Epoch Ensemble	# N-best	Precision [mm]
1-20	1	35.26
1-20	10	35.21
1-20	20	35.20
1-20	50	35.18
1-20	100	**35.18**

4) POSE PRIOR

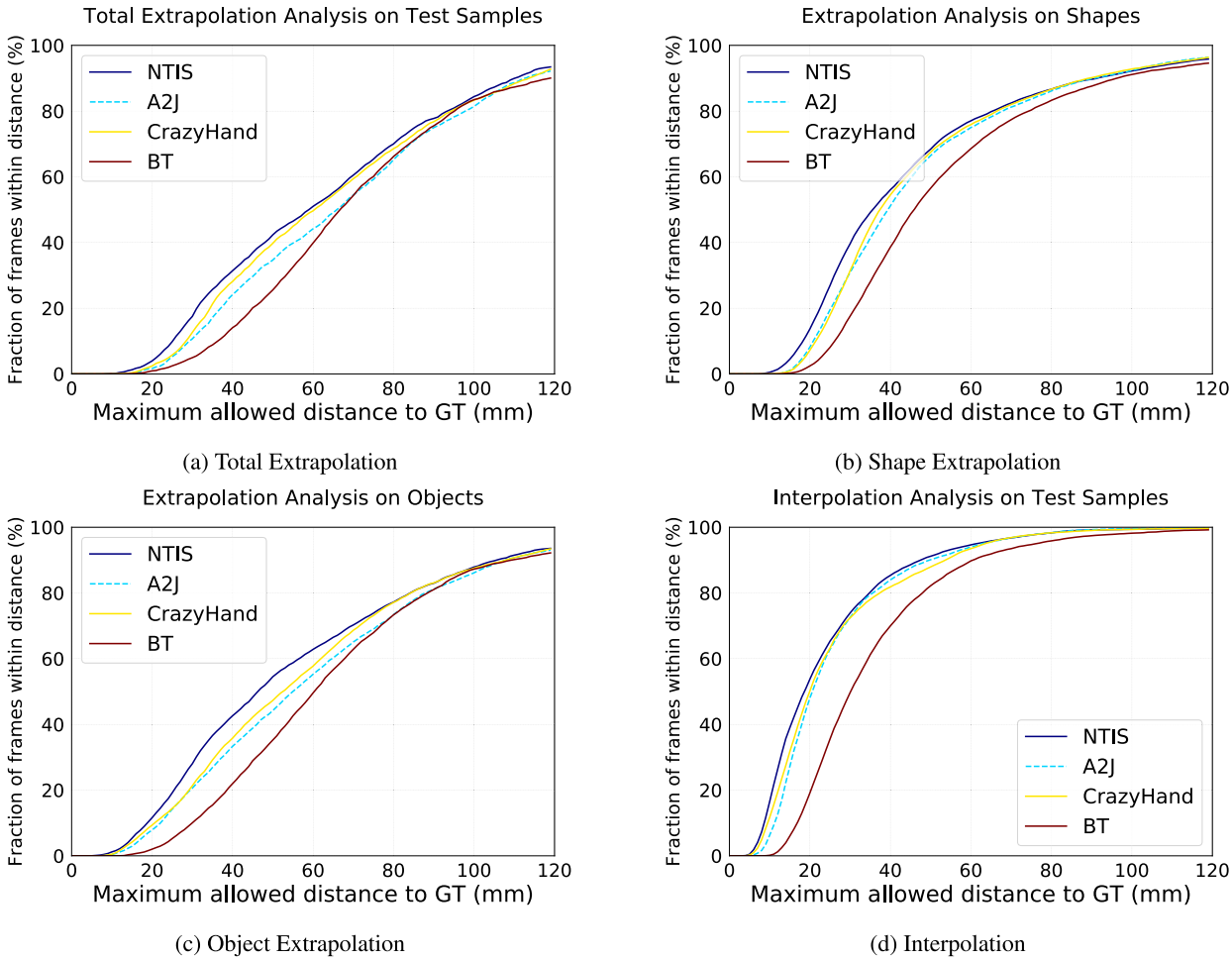
We apply the pose prior by re-projecting the obtained pose vector using a pretrained TruncatedSVD model. We use five models with number of components $n_i \in \{5, 10, 15, 20, 25\}$ to obtain five different poses. These poses are combined by using weighted averaging using weights $w_i \in \{0.05, 0.35, 0.6, 0.1, 0.1\}$. During the experimentation we found out that the occluded joints (by the object or joint self-occlusion) are predicted with low confidence and well visible joints have high confidence. That is why we leave the highly probable joint locations estimations as they are and substitute only the locations with low confidence. The confidence threshold was experimentally set to 0.158. By applying this procedure we obtain a precision of **33.32 mm**. The setup is as in experiment summarized in Table 4; ensemble of five best epochs, and 100 best locations from each epoch.

5) TEMPORAL CONTEXT

In the following experiment we study the impact of the size of the temporal context on the precision. We report the findings in Table 6. The results indicate that the best precision is achieved when we use 3 frames before and 3 frames after the current frame. That is 7 frames in total. The individual predictions from different frames are first processed by the pose prior approach and the results are then averaged. We also tried to first average the raw predictions from different frames

TABLE 6. Results for different sizes of temporal context on the test data.

Epoch Ensemble	N-best	Window Size	Precision [mm]
5-best	100	+1, -1	33.23
5-best	100	+3, -3	**33.11**
5-best	100	+5, -5	33.12
5-best	100	+7, -7	33.16
5-best	100	+10, -10	33.19



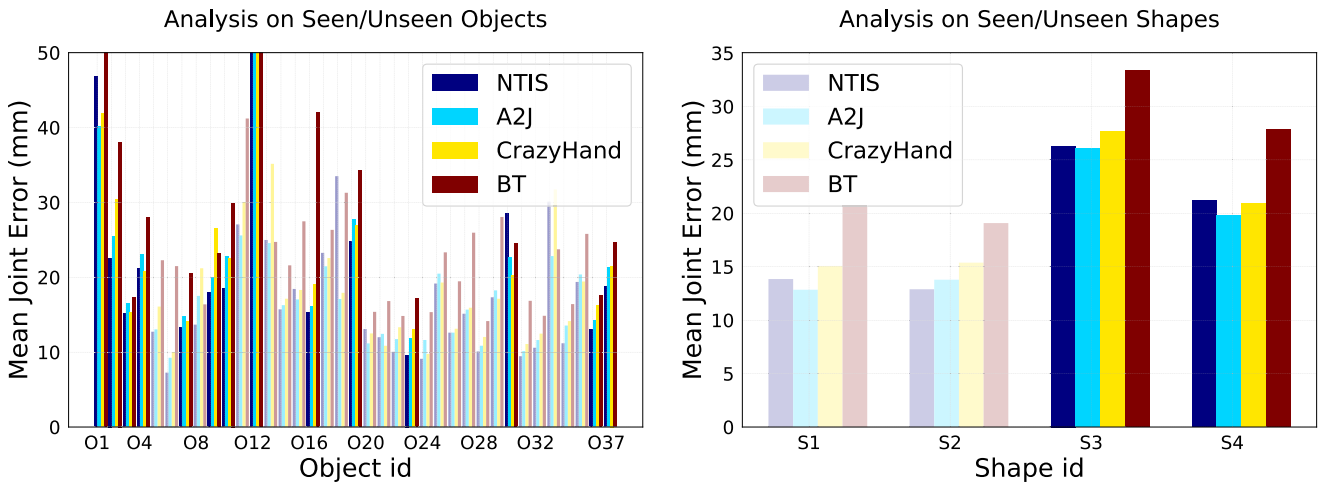
(a) Total Extrapolation

(b) Shape Extrapolation

(c) Object Extrapolation

(d) Interpolation

FIGURE 8. Analysis of different approaches to hand pose estimation. The graphs from [31] represent fraction of frames from test sets that predicted joint locations with maximum error below given threshold. It can be seen that our approach (NTIS) outperforms the others, especially in the domain of lower error thresholds. Other approaches of the challenge (A2J, CrazyHand and BT) are summarized in [31].



(a) Mean joint error on unseen (solid) and seen (transparent) objects.

(b) Mean joint error on unseen (solid) and seen (transparent) hands.

FIGURE 9. Analysis of different approaches to hand pose estimation, our approach is NTIS. The graphs from [31] represent the average joint location error in mm. Different approaches (A2J, CrazyHand and BT) are summarized in [31].

and then using the pose prior. The results were very similar (**33.09 mm**).

F. RESULTS

The progressive improvement of the precision is reported in Table 7. The notation in the table follows the standard convention presented in this article: 5Nbest is the ensemble of 5 best performing epochs with 100 best location averaging, TrSVD is the pose prior post-processing, tem. context is the temporal context post-processing. When we compare our solution to the re-implementation of the original V2V-PoseNet system, we achieve a relative improvement of 17.6%.

We present results of our method for the HANDS19 Challenge: Task 2 - Depth-Based 3D Hand Pose Estimation while Interacting with Objects. In the challenge the results were categorized into several groups. (1) Shape - represents unique hands not present in the training data, (2) Object - represents unique objects being manipulated not present in the training data, (3) Interpolation - both shape and object are present in the training data, and (4) Extrapolation - both the shape and object are not seen in the training data. The overall results are reported on category (4) Extrapolation. We summarize the categorical results in Table 8.

We show qualitative results and effect of individual post-processing methods in Figure 7. Finally, we present the quantitative results generated at the time of the deadline by the organizers of the HANDS19 Challenge in Figure 8 and Figure 9 [31].

1) OTHER DATASETS

In this section we report results for other recent datasets, see Table 9. Since there is no other widely acknowledged benchmark dataset for depth data of hands interacting with objects from egocentric view, we chose other standard benchmark datasets: NYU, HO3D, and HANDS2017 to test the performance of our solution on data that are not exclusively egocentric.

HO3D [67] is a dataset of RGB-D images of hands manipulating with objects from a 3^{rd} person view. Annotations of the pose of the hand and the object are available. The dataset is composed of sequences of 10 people handling 10 different objects. For the training of our system we used 66,034 depth images with 21 annotated hand joints from 55 sequences and for evaluation we used 11,524 depth images from 13 sequences. Some people and objects are deliberately left out of the training set. The evaluation was realized through the Codalab[2] competition system.

NYU contains 72,757 training and 8,252 testing RGB-D images and 3D annotation of the hand pose. The training set is generated by one person while the test set is generated by two people (one is the same as in the train set). For the training, we used the depth data from the frontal view. For evaluation, we used 14 out of 36 joints, which is a standard for this dataset. To be able to compare our results with prior works, we had to use the same practice of applying two different sizes of the voxelization cubes for the two different performers.[3]

The HANDS2017 dataset [41] is one of the largest benchmark dataset for 3D hand pose estimation. It is composed of 957k training and 295k testing RGB-D images that are sampled from the BigHand2.2M [68] and F-PHAB dataset. The training set is composed of five people and the testing set is composed of 10 people, including five unseen people. The hand pose is defined as 3D positions of 21 joints. The evaluation was realized through the Codalab[4] competition system.

For most datasets we used our own re-implementation of the V2V-PoseNet system, since there might be some unreported nuances of the implementation of the original system that might influence the resulting precision. For example, we were not able to achieve the numbers reported in the original article for the NYU dataset, unless we used the ground truth CoM location for the voxelization cube. The original and re-implemented V2V models use 10 epoch ensemble. We were able to significantly improve the precision on every tested dataset. In the last row of Table 9 the GTHandCrop

TABLE 7. Effect of 4x wider architecture, 5Nbest ensemble and post-processing on the test data.

Method	Precision [mm]
Baseline: v2v 1-10 epoch ens.	40.17
+ 1-20 epoch ens.	37.66
+ width 64	35.26
+ 5Nbest	34.85
+ TrSVD	33.26
+ tem.context	33.09

TABLE 8. Results for different categories on the HANDS19 Challenge: Task 2 test data. Deadline of the challenge was 28/10/2019.

Category	Precision before deadline [mm]	Precision after deadline [mm]
Shape	23.62	23.32
Object	29.07	29.81
Interpolation	17.42	17.84
Extrapolation	33.48	**33.09**

TABLE 9. Results for other datasets.

Dataset	Method	Precision [mm]
HO3D	v2v (our original impl.)	36.2
	v2v (5Nbest-TrSVD-tem.context)	**31.3**
HANDS2017	v2v (HANDS2017 Challenge)	9.95
	v2v (5Nbest-TrSVD)	**7.51**
NYU	v2v (our original impl.)	9.3
	v2v (5Nbest-TrSVD-tem.context)	**8.6**
NYU GTHandCrop	v2v (our original impl.)	8.3
	v2v (5Nbest-TrSVD-tem.context)	**7.5**

[2]https://competitions.codalab.org/competitions/22485
[3]250 mm and 300 mm respectively.
[4]https://competitions.codalab.org/competitions/17356

means we omit the CoM refinement network and use the ground truth CoM for hand localization.

2) COMPUTATIONAL TIME

We investigate the relative increase in computational time of V2V-PoseNet prediction when we employ our method. We focus on the time needed for predicting each input image for standard V2V-PoseNet architecture and our 4x wider architecture. We measure the time without employing epoch ensambling (which has a linear burden on the prediction time) and we use one GPU.[5] Next, we state that the finding of the N-best maximum values in the resulting heatmap has negligible impact on computational time when compared to finding only one maximum value. The resulting times are then 12 frames per second for the standard V2V architecture and $\approx$ 3 frames per second for our 4 times wider architecture. This implies a 4-fold increase in computational time, which corresponds to the increase in number of parameters of the wider model. To accommodate for the speed, we also computed the precision of our system when it has a comparable run-time to the original V2V-PoseNet epoch ensemble. For this purpose we ensemble only 3 best networks and achieve a precision of 33.19 mm for the HANDS2019 dataset, which considerably outperforms the original ensemble of V2V-PoseNet with 10 epoch ensemble (40.17 mm).

V. CONCLUSION

In this work we present an computationally extensive study on the current state-of-the-art method for the hand pose estimation from the egocentric viewpoint when interacting with objects. We analyze the architecture of a Voxel-to-Voxel PoseNet [37], and find that the base width of 64 gives the best results.

We introduce several schemes to obtain the location of the joint from the predicted heat-maps. We find out that when we use the weighted averaging of the whole output heat-maps we obtain the best results. However, when we use different techniques that allow us to obtain more hypotheses from the heat-maps and we produce the results as an ensemble of these hypotheses, we improve the precision. Furthermore, we apply several post-processing techniques to obtain more precise results. The post-processing is based on incorporating a prior model of hand poses in the form of the TruncatedSVD, and using the epoch ensemble. Finally, we show that using the temporal context when available yields even better results.

For each method we perform an ablation study to find the optimal parameters. When all the techniques are applied we achieve a mean joint location error of 33.09 mm, which is the best resulting score for the HANDS19 Challenge: Task 2 - Depth-Based 3D Hand Pose Estimation while Interacting with Objects. It should be noted that in the time of the deadline of the challenge we achieved a precision of 33.48 mm, which was still the best score at that time. Furthermore, our system also achieved the second best ever precision of 7.51 mm (up to May 2020) on HANDS2017 dataset.

[5]Nvidia GeForce GTX1080 Ti

ACKNOWLEDGMENT

Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA LM2018140) provided within the program Projects of Large Research, Development and Innovations Infrastructures. The authors would also like to thank the organizers of the HANDS 2019 Challenge for their service provided to all involved participants.

REFERENCES

[1] D. Ryumin, D. Ivanko, A. Axyonov, I. Kagirov, A. Karpov, and M. Zelezny, "Human-robot interaction with smart shopping trolley using sign language: Data collection," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2019, pp. 949–954.

[2] T. Feix, J. Romero, C. H. Ek, H.-B. Schmiedmayer, and D. Kragic, "A metric for comparing the anthropomorphic motion capability of artificial hands," *IEEE Trans. Robot.*, vol. 29, no. 1, pp. 82–93, Feb. 2013.

[3] C. Zimmermann and T. Brox, "Learning to estimate 3D hand pose from single RGB images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4903–4911.

[4] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 409–419.

[5] B. Tekin, F. Bogo, and M. Pollefeys, "H+O: Unified egocentric recognition of 3D hand-object poses and interactions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4511–4520.

[6] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, "Combining self-supervised learning and imitation for vision-based rope manipulation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 2146–2153.

[7] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 5628–5635.

[8] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Scaling egocentric vision: The EPIC-KITCHENS dataset," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 720–736.

[9] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Efficient model-based 3D tracking of hand articulations using kinect," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 101.1–101.11.

[10] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun, "Realtime and robust hand tracking from depth," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1106–1113.

[11] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, D. Freedman, P. Kohli, E. Krupka, A. Fitzgibbon, and S. Izadi, "Accurate, robust, and flexible real-time hand tracking," in *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst.*, Apr. 2015, pp. 3633–3642.

[12] S. Sridhar, F. Mueller, M. Zollhoefer, D. Casas, A. Oulasvirta, and C. Theobalt, "Real-time joint tracking of a hand manipulating an object from RGB-D input," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 294–310.

[13] D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall, "Capturing hands in action using discriminative salient points and physics simulation," *Int. J. Comput. Vis.*, vol. 118, no. 2, pp. 172–193, Jun. 2016.

[14] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, E. Soto, D. Sweeney, J. Valentin, B. Luff, A. Topalian, E. Wood, S. Khamis, P. Kohli, T. Sharp, S. Izadi, R. Banks, A. Fitzgibbon, and J. Shotton, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," in *Proc. ACM Trans. Graph. (TOG), ACM SIGGRAPH*, vol. 35, Jul. 2016, pp. 1–12.

[15] D. Tang, T.-H. Yu, and T.-K. Kim, "Real-time articulated hand pose estimation using semi-supervised transductive regression forests," in *Proc. IEEE Int. Conf. Comput. Vis.* Washington, DC, USA: IEEE Computer Society, Dec. 2013, pp. 3224–3231.

[16] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim, "Latent regression forest: Structured estimation of 3D articulated hand posture," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3786–3793.
[17] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 824–832.
[18] C. Wan, A. Yao, and L. Van Gool, "Hand pose estimation from local surface normals," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2016, pp. 554–569.
[19] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton, "Opening the black box: Hierarchical sampling optimization for estimating human hand pose," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3325–3333.
[20] C. Choi, A. Sinha, J. H. Choi, S. Jang, and K. Ramani, "A collaborative filtering approach to real-time hand pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2336–2344.
[21] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Trans. Graph.*, vol. 33, no. 5, pp. 1–10, Sep. 2014, doi: 10.1145/2629500.
[22] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands deep in deep learning for hand pose estimation," in *Proc. Comput. Vis. Winter Workshop*, 2015, pp. 1–10.
[23] M. Oberweger, P. Wohlhart, and V. Lepetit, "Training a feedback loop for hand pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3316–3324.
[24] A. Sinha, C. Choi, and K. Ramani, "DeepHand: Robust hand pose estimation by completing a matrix imputed with deep features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4150–4158.
[25] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Robust 3D hand pose estimation in single depth images: From single-view CNN to multi-view CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3593–3601.
[26] Q. Ye, S. Yuan, and T.-K. Kim, "Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation," in *Proc. Eur. Conf. Comput. Vision. (ECCV)*. Cham, Switzerland: Springer, 2016, pp. 346–361.
[27] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua, "Structured prediction of 3D human pose with deep neural networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 1–11.
[28] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "3D convolutional neural networks for efficient and robust hand pose estimation from single depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1991–2000.
[29] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Intell. Res.*, vol. 11, no. 1, p. 169–198, Jul. 1999.
[30] H. Bonab and F. Can, "Less is more: A comprehensive framework for the number of components of ensemble classifiers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2735–2745, Sep. 2019.
[31] A. Armagan, G. Garcia-Hernando, S. Baek, S. Hampali, M. Rad, Z. Zhang, S. Xie, M. Chen, B. Zhang, F. Xiong, and Y. Xiao, "Measuring generalisation to unseen viewpoints, articulations, shapes and objects for 3D hand pose estimation under hand-object interaction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 1–32.
[32] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Real-time 3D hand pose estimation with 3D convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 956–970, Apr. 2019.
[33] M. Oberweger, P. Wohlhart, and V. Lepetit, "Generalized feedback loop for joint hand-object pose estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 1898–1912, Aug. 2020.
[34] T.-Y. Chen, M.-Y. Wu, Y.-H. Hsieh, and L.-C. Fu, "Deep learning for integrated hand detection and pose estimation," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 615–620.
[35] C. Choi, S. Kim, and K. Ramani, "Learning hand articulations by hallucinating heat distribution," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3104–3113.
[36] Y. Che, Y. Song, and Y. Qi, "A novel framework of hand localization and hand pose estimation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 2222–2226.
[37] J. Y. Chang, G. Moon, and K. M. Lee, "V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5079–5088.
[38] F. Huang, A. Zeng, M. Liu, J. Qin, and Q. Xu, "Structure-aware 3D hourglass network for hand pose estimation from single depth image," in *Brit. Mach. Vis. Conf. (BMVC)*. London, U.K.: BMVA Press, 2018, p. 289. [Online]. Available: http://bmvc2018.org/contents/papers/1133.pdf
[39] P.-W. Ting, E.-T. Chou, Y.-H. Tang, and L.-C. Fu, "Hand pose estimation based on 3D residual network with data padding and skeleton steadying," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, C. Jawahar, H. Li, G. Mori, and K. Schindler, Eds. Cham, Switzerland: Springer, 2019, pp. 293–307.
[40] F. Guo, Z. He, S. Zhang, X. Zhao, and J. Tan, "Attention-based pose sequence machine for 3D hand pose estimation," *IEEE Access*, vol. 8, pp. 18258–18269, 2020.
[41] S. Yuan, G. Garcia-Hernando, B. Stenger, G. Moon, J. Y. Chang, K. M. Lee, P. Molchanov, J. Kautz, S. Honari, L. Ge, and J. Yuan, "Depth-based 3D hand pose estimation: From current achievements to future goals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2636–2645.
[42] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI* (Lecture Notes in Computer Science), vol. 9351. Cham, Switzerland: Springer, 2015, pp. 234–241. [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a
[43] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.
[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[45] J. Malik, A. Elhayek, F. Nunnari, K. Varanasi, K. Tamaddon, A. Heloir, and D. Stricker, "DeepHPS: End-to-end estimation of 3D hand pose and shape by learning from synthetic depth," in *Proc. Int. Conf. 3D Vis. (3DV)*, Verona, Italy, Sep. 2018, pp. 110–119.
[46] F. Xiong, B. Zhang, Y. Xiao, Z. Cao, T. Yu, J. T. Zhou, and J. Yuan, "A2J: Anchor-to-joint regression network for 3D articulated pose estimation from a single depth image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 793–802.
[47] P. Ren, H. Sun, Q. Qi, J. Wang, and W. Huang, "SRN: Stacked regression network for real-time 3D hand pose estimation," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2019, p. 112.
[48] K. Du, X. Lin, Y. Sun, and X. Ma, "CrossInfoNet: Multi-task information sharing based hand pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9896–9905.
[49] C.-H. Yoo, S. Ji, Y.-G. Shin, S.-W. Kim, and S.-J. Ko, "Fast and accurate 3D hand pose estimation via recurrent neural network for capturing hand articulations," *IEEE Access*, vol. 8, pp. 114010–114019, 2020.
[50] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2528–2535.
[51] L. Ge, Z. Ren, and J. Yuan, "Point-to-point regression pointnet for 3D hand pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 475–491.
[52] S. Li and D. Lee, "Point-to-pose voting based hand pose estimation using residual permutation equivariant layer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11927–11936.
[53] X. Chen, G. Wang, C. Zhang, T.-K. Kim, and X. Ji, "SHPR-Net: Deep semantic hand pose regression from point clouds," *IEEE Access*, vol. 6, pp. 43425–43439, 2018.
[54] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid, "Learning joint reconstruction of hands and manipulated objects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11807–11816.
[55] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Self-supervised 3D hand pose estimation through training by fitting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10853–10862.
[56] M. Rad, M. Oberweger, and V. Lepetit, "Feature mapping for learning fast and accurate 3D pose inference from synthetic images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4663–4672.
[57] 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 *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 1284–1293. [Online]. Available: https://handtracker.mpi-inf.mpg.de/projects/OccludedHands/

[58] M. Oberweger and V. Lepetit, "DeepPrior++: Improving fast and accurate 3D hand pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 585–594, doi: 10.1109/ICCVW.2017.75.
[59] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, p. 448–456.
[60] G. ElKoura and K. Singh, "Handrix: Animating the human hand," in *Proc. ACM SIGGRAPH/Eurographics Symp. Comput. Animation*. Vienna, Austria: Eurographics Association, 2003, pp. 110–119.
[61] P. C. Hansen, "The truncatedSVD as a method for regularization," *BIT*, vol. 27, no. 4, pp. 534–553, Dec. 1987.
[62] S. Yuan, Q. Ye, G. Garcia-Hernando, and T. Kim, "The 2017 hands in the million challenge on 3D hand pose estimation," *CoRR*, vol. abs/1707.02237, pp. 1–7, Jul. 2017. [Online]. Available: http://arxiv.org/abs/1707.02237
[63] S. Tokui, H. Y. Vincent, R. Okuta, T. Akiba, Y. Niitani, T. Ogawa, S. Saito, S. Suzuki, K. Uenishi, and B. Vogel, "Chainer: A deep learning framework for accelerating the research cycle," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2019, pp. 2002–2011, doi: 10.1145/3292500.3330756.
[64] *5th International Workshop on Observing and Understanding Hands in Action*. Accessed: Oct. 28, 2019. [Online]. Available: https://sites.google.com/view/hands2019/
[65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
[66] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, "Deep double descent: Where bigger models and more data hurt," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, Addis Ababa, Ethiopia, 2020, pp. 1–24.
[67] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit, "HOnnotate: A method for 3D annotation of hand and object poses," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3196–3206.
[68] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2605–2613.

MAREK HRÚZ received the Ph.D. degree in computer science from the Faculty of Applied Sciences, University of West Bohemia in Pilsen, Czech Republic, in 2012.

He is currently an Assistant Professor with the Department of Cybernetics, Faculty of Applied Sciences, University of West Bohemia in Pilsen, and a Senior Researcher with the New Technologies for the Information Society. His main research interests are computer vision, machine learning, deep learning, signal processing, and multi-modal signal processing.

Mr. Hrúz's awards and honors include the first prize of the international competition Hands 2019—Challenge in the task of Depth-Based 3D Hand Pose Estimation while Interacting with Objects.

JAKUB KANIS received the Ph.D. degree in computer science from the Faculty of Applied Sciences, University of West Bohemia in Pilsen, Czech Republic, in 2009.

He is currently an Assistant Professor with the Department of Cybernetics, Faculty of Applied Sciences, University of West Bohemia in Pilsen, and a Senior Researcher with the New Technologies for the Information Society. His main research interests concern artificial intelligence, machine learning, computer vision, and natural language processing.

Mr. Kanis' awards and honors include the first prize of the international competition Hands 2019—Challenge in the task of Depth-Based 3D Hand Pose Estimation while Interacting with Objects.

ZDENĚK KRŇOUL was born in Rokycany, Czech Republic, in 1979. He received the Ph.D. degree in computer science from the Faculty of Applied Sciences, University of West Bohemia in Pilsen, Czech Republic, in 2008.

Since 2016, he has been an Assistant Professor with the Department of Cybernetics, Faculty of Applied Sciences, University of West Bohemia in Pilsen, and a Senior Researcher with the New Technologies for the Information Society. His main research interests concern 3-D computer vision, data analysis, and synthesis methods, object and pose tracking and detection, and 3-D graphic and animation of human body.

Mr. Krňoul's awards and honors include the first prize of LIPS 2008—Visual Speech Synthesis Challenge in the Intelligibility category and the first prize of the international competition Hands 2019—Challenge in the task of Depth-Based 3D Hand Pose Estimation while Interacting with Objects in conjunction with the ICCV 2019 workshop.