Learning to Estimate 3D Hand Pose from Single RGB Images

Christian Zimmermann, Thomas Brox
University of Freiburg
{zimmermann, brox}@cs.uni-freiburg.de

Abstract

Low-cost consumer depth cameras and deep learning have enabled reasonable 3D hand pose estimation from single depth images. In this paper, we present an approach that estimates 3D hand pose from regular RGB images. This task has far more ambiguities due to the missing depth information. To this end, we propose a deep network that learns a network-implicit 3D articulation prior. Together with detected keypoints in the images, this network yields good estimates of the 3D pose. We introduce a large scale 3D hand pose dataset based on synthetic hand models for training the involved networks. Experiments on a variety of test sets, including one on sign language recognition, demonstrate the feasibility of 3D hand pose estimation on single color images.

1. Introduction

The hand is the primary operating tool for humans. Therefore, its location, orientation and articulation in space is vital for many potential applications, for instance, object handover in robotics, learning from demonstration, sign language and gesture recognition, and using the hand as an input device for man-machine interaction.

Full 3D hand pose estimation from single images is difficult because of many ambiguities, strong articulation, and heavy self-occlusion, even more so than for the overall human body. Therefore, specific sensing equipment like data gloves or markers are used, which restrict the application to limited scenarios. Also the use of multiple cameras severely limits the application domain. Most contemporary works rely on the depth image from a depth camera. However, depth cameras are not as commonly available as regular color cameras, and they only work reliably in indoor environments.

In this paper, we present an approach to learn full 3D hand pose estimation from single color images without the need for any special equipment. We capitalize on the capability of deep networks to learn sensible priors from data in order to resolve ambiguities. Our overall approach consists of three deep networks that cover important subtasks on the way to the 3D pose; see Figure 2. The first network provides a hand segmentation to localize the hand in the image. Based on its output, the second network localizes hand keypoints in the 2D images. The third network finally derives the 3D hand pose from the 2D keypoints, and is the main contribution of this paper. In particular, we introduce a canonical pose representation to make this learning task feasible.

Another difficulty compared to 3D pose estimation at the level of the human body is the restricted availability of data. While human body pose estimation can leverage several motion capture databases, there is hardly any such data for hands. To train a network, a large dataset with ground truth 3D keypoints is needed. Since there is no such dataset with sufficient variability, we created a synthetic dataset with various data augmentation options.

The resulting hand pose estimation system yields very promising results, both qualitatively and quantitatively on existing small-scale datasets. We also demonstrate the use of 3D hand pose for the task of sign language recognition. The dataset and our trained networks are available online.

2. Related work

2D Human Pose Estimation. Spurred by the MPII Human Pose benchmark and the advent of Convolutional
Figure 2: Our approach consists of three building blocks. First, the hand is localized within the image by a segmentation network (HandSegNet). Accordingly to the hand mask, the input image is cropped and serves as input to the PoseNet. This localizes a set of hand keypoints represented as score maps c. Subsequently, the PosePrior network estimates the most likely 3D structure conditioned on the score maps. This figure serves for illustration of the overall approach and does not reflect the exact architecture of the individual building blocks.

Neural Networks (CNN) this field made large progress in the last years. The CNN architecture of Toshev and Szegedy [24] directly regresses 2D cartesian coordinates from color image input. More recent works like Thompson et al. [22] and Wei et al. [19] turned towards regressing score maps. For parts of our work, we employ a comparable network architecture as Wei et al. [19].

3D Human Pose Estimation. We only discuss the most relevant works here and refer to Sarafianos et al. [17] for more information. Like our approach, many works use a two part pipeline [23, 7, 6, 21, 5]. First they detect keypoints in 2D to utilize the discriminative power of current CNN approaches and then attempt to lift the set of 2D detections into 3D space. Different methods for lifting the representation have been proposed: Chen et al. [6] deployed a nearest neighbor matching of a given 2D prediction using a database of 2D to 3D correspondences. Tome et al. [21] created a probabilistic 3D pose model based upon a mixture of probabilistic PCA bases. Bogo et al. [5] optimizes the reprojection error between 3D joint locations of a statistical body shape model and 2D prediction. Pavlakos et al. [15] proposed a volumetric approach that treats pose estimation as per voxel prediction of scores in a coarse-to-fine manner, which gives a natural representation to the data, but is computationally expensive and limited by the GPU memory to fit the voxel grid. Recently, there have been several approaches that apply deep learning for lifting 2D keypoints to 3D pose for human body pose estimation [26, 11, 16]. Furthermore Mehta et al. [10] uses transfer learning to infer the 3D body pose directly from images with a single network. While these works are all on 3D body pose estimation, we provide the first such work for 3D hand pose estimation, which is substantially harder due to stronger articulation and self-occlusion, as well as less data being available.

Hand Pose Estimation. Athitsos and Sclaroff [4] proposed a single frame based detection approach based on edge maps and Chamfer matching. With the advent of low-cost consumer depth cameras, research focused on hand pose from depth data. Oikonomidis et al. [14] proposed a technique based on Particle Swarm Optimization (PSO). Sharp et al. [18] added the possibility for reinitialization. A certain number of candidate poses is created and scored against the observed depth image. Thompson et al. [22] used a CNN for detection of hand keypoints in 2D, which is conditioned on a multi-resolution image pyramid. The pose in 3D is recovered by solving an inverse kinematics optimization problem. Approaches like Zhou et al. [27] or Oberweger et al. [12] train a CNN that directly regresses 3D coordinates given hand cropped depth maps. Whereas Oberweger et al. [12] explored the possibility to encode correlations between keypoint coordinates in a compressing bottleneck, Zhou et al. [27] estimate angles between bones of the kinematic chain instead of Cartesian coordinates. Oberweger et al. [13] utilizes a CNN that can synthesize a depth map from a given pose estimate. This allows them to successively refine initial pose estimates by minimizing the distance between the observed and the synthesized depth image.

There aren’t any approaches yet that tackle the problem of 3D hand pose estimation from a single color image with a learning based formulation. Previous approaches differ because they rely on depth data [22, 27, 12, 13], they use explicit models to infer pose by matching against a predefined database of poses [4], or they only perform tracking based on an initial pose rather than full pose estimation [14, 18].
4. Estimation of 3D hand pose

Given a color image $I \in \mathbb{R}^{N \times M \times 3}$ showing a single hand, we want to infer its 3D pose. We define the hand pose by a set of coordinates $w_i = (x_i, y_i, z_i)$, which describe the locations of $J$ keypoints in 3D space, i.e., $i \in [1, J]$ with $J = 21$ in our case.

The problem of inferring 3D coordinates from a single 2D observation is ill-posed. Among other ambiguities, there is a scale ambiguity. Thus, we infer a scale-invariant 3D structure by training a network to estimate normalized coordinates $w_{i, \text{norm}} = \frac{1}{s} \cdot w_i$, \hspace{1cm} (1)

where $s = \|w_{k+1} - w_k\|_2$ is a sample dependent constant that normalizes the distance between a certain pair of keypoints to unit length. We choose $k$ such that $s = 1$ for the first bone of the index finger.

Moreover, we use relative 3D coordinates to learn a translation invariant representation of hand poses. This is realized by subtracting the location of a defined root keypoint. The relative and normalized 3D coordinates are given by $w_{i, \text{rel}} = w_{i, \text{norm}} - w_{r, \text{norm}}$, \hspace{1cm} (2)

where $r$ is the root index. In experiments the palm keypoint was the most stable landmark. Thus we use $r = 0$.

3. Hand pose representation

Given a color image $I \in \mathbb{R}^{N \times M \times 3}$ showing a single hand, we want to infer its 3D pose. We define the hand pose by a set of coordinates $w_i = (x_i, y_i, z_i)$, which describe the locations of $J$ keypoints in 3D space, i.e., $i \in [1, J]$ with $J = 21$ in our case.

The problem of inferring 3D coordinates from a single 2D observation is ill-posed. Among other ambiguities, there is a scale ambiguity. Thus, we infer a scale-invariant 3D structure by training a network to estimate normalized coordinates $w_{i, \text{norm}} = \frac{1}{s} \cdot w_i$, \hspace{1cm} (1)

where $s = \|w_{k+1} - w_k\|_2$ is a sample dependent constant that normalizes the distance between a certain pair of keypoints to unit length. We choose $k$ such that $s = 1$ for the first bone of the index finger.

Moreover, we use relative 3D coordinates to learn a translation invariant representation of hand poses. This is realized by subtracting the location of a defined root keypoint. The relative and normalized 3D coordinates are given by $w_{i, \text{rel}} = w_{i, \text{norm}} - w_{r, \text{norm}}$, \hspace{1cm} (2)

where $r$ is the root index. In experiments the palm keypoint was the most stable landmark. Thus we use $r = 0$.

3. Hand pose representation

4. Estimation of 3D hand pose

We estimate three-dimensional normalized coordinates $w_{i, \text{rel}}$ from a single input image. An overview of the general approach is provided in Figure 2. In the following sections, we provide details on its components.

4.1. Hand segmentation with HandSegNet

For hand segmentation we deploy a network architecture that is based on and initialized by the person detector of Wei et al. [19]. They cast the problem of 2D person detection as estimating a score map for the center position of the human. The most likely location is used as center for a fixed size crop. Since the hand size drastically changes across images and depends much on the articulation, we rather cast the hand localization as a segmentation problem. Our HandSegNet is a smaller version of the network from Wei et al. [19] trained on our hand pose dataset. Details on the network architecture and its training procedure are provided in the supplemental material. The hand mask provided by HandSegNet allows us to crop and normalize the inputs in size, which simplifies the learning task for the PoseNet.

4.2. Keypoint score maps with PoseNet

We formulate localization of 2D keypoints as estimation of 2D score maps $c = \{c_1(u, v), \ldots, c_J(u, v)\}$. We train a network to predict $J$ score maps $c_j \in \mathbb{R}^{N \times M}$, where each map contains information about the likelihood that a certain keypoint is present at a spatial location.

The network uses an encoder-decoder architecture similar to the Pose Network by Wei et al. [19]. Given the image feature representation produced by the encoder, an initial score map is predicted and is successively refined in resolution. We initialized with the weights from Wei et al. [19], where it applies, and retrained the network for hand keypoint detection. A complete overview over the network architecture is located in the supplemental material.

4.3. 3D hand pose with the PosePrior network

The PosePrior network learns to predict relative, normalized 3D coordinates conditioned on potentially incomplete or noisy score maps $c(u, v)$. To this end, it must learn the manifold of possible hand articulations and their prior probabilities. Conditioned on the score maps, it will output the most likely 3D configuration given the 2D evidence.

Instead of training the network to predict absolute 3D coordinates, we rather propose to train the network to predict coordinates within a canonical frame and additionally esti-
mate the transformation into the canonical frame. Explicitly enforcing a representation that is invariant to the global orientation of the hand is beneficial to learn a prior, as we show in our experiments in section 6.2.

Given the relative normalized coordinates we propose to use a canonical frame \( w^c \), that relates to \( w^{rel} \) in the following way: An intermediate representation

\[
    w^c = R(w^{rel}) \cdot w^{rel}
\]

with \( R(w^{rel}) \in \mathbb{R}^{3 \times 3} \) being a 3D rotation matrix is calculated in a two step procedure. First, one seeks the rotation \( R_{xz} \) around the x- and z-axis such that a certain keypoint \( w^c \) is aligned with the y-axis of the canonical frame:

\[
    R_{xz} \cdot w^c = \lambda \cdot (0, 1, 0)^\top \quad \text{with} \quad \lambda \geq 0.
\]

Afterwards, a rotation \( R_y \) around the y-axis is calculated such that

\[
    R_y \cdot R_{xz} \cdot w^c = (\eta, \zeta, 0)
\]

with \( \eta \geq 0 \) for a specified keypoint index \( o \). The total transformation between canonical and original frame is given by

\[
    R(w^{rel}) = R_y \cdot R_{xz}.
\]

In order to deal appropriately with the symmetry between left and right hands, we flip right hands along the z-axis, which yields the side agnostic representation

\[
    w^c = \begin{cases} 
        (x^c, y^c, z^c)^\top & \text{if its a left hand} \\
        (x^c, y^c, -z^c)^\top & \text{if its a right hand}
    \end{cases}
\]

that resembles our proposed canonical coordinate system. Given this canonical frame definition, we train our network to estimate the 3D coordinates within the canonical frame \( w^c \) and separately to estimate the rotation matrix \( R(w^{rel}) \), which we parameterize using axis-angle notation with three parameters. Estimating the transformation \( R \) is equivalent to predicting the viewpoint of a given sample with respect to the canonical frame. Thus, we refer to the problem as viewpoint estimation.

The network architecture for the pose prior has two parallel processing streams; see Figure 5. The streams use an almost identical architecture given in the supplementary.

4.4. Network training

For training of HandSegNet we apply standard softmax cross-entropy loss and \( L_2 \) loss for Posenet. The PosePrior network uses two loss terms. First a squared \( L_2 \) loss for the canonical coordinates

\[
    L_c = \|w^c_{gt} - w^c_{pred}\|^2_2
\]

based on the network predictions \( w^c_{pred} \) and the ground truth \( w^c_{gt} \). Secondly, a squared \( L_2 \) loss is imposed on the canonical transformation matrix:

\[
    L_r = \| R_{pred} - R_{gt} \|^2_2.
\]

The total loss function is the unweighted sum of \( L_c \) and \( L_r \).

We used Tensorflow [2] with the Adam solver [9] for training. Details on the learning procedure are in the supplementary material.

5. Datasets for hand pose estimation

5.1. Available datasets

There are two available datasets that apply to our problem, as they provide RGB images and 3D pose annotation. The so-called Stereo Hand Pose Tracking Benchmark [25] provides both 2D and 3D annotations of 21 keypoints for 18000 stereo pairs with a resolution of \( 640 \times 480 \). The dataset shows a single person’s left hand in front of 6 different backgrounds and under varying lighting conditions. We divided the dataset into an evaluation set of 3000 images (S-val) and a training set with 15000 images (S-train).

Dexter [20] is a dataset providing 3111 images showing two operators performing different kinds of manipulations with a cuboid in a restricted indoor setup. The dataset provides color images, depth maps, and annotations for fingertips and cuboid corners. The color images have a spatial resolution of \( 640 \times 320 \). Due to the incomplete hand annotation, we use this dataset only for investigating the cross-dataset generalization of our network. We refer to this test set as Dexter.

We downsampled both datasets to a resolution of \( 320 \times 240 \) to be compatible with our rendered dataset. We transform our results back to coordinates in the original resolution, when we report pixel accuracies in the image domain.

The NYU Hand Pose Dataset by Tompson et al. [22], commonly used for hand pose estimation from depth images, does not apply to a color based approach, because only registered color images are provided. In the supplementary we show more evidence why this dataset cannot be used for our task.

5.2. Rendered hand pose dataset

The above datasets are not sufficient for training a deep network due to limited variation, number of available samples, and partially incomplete annotation. Therefore, we
complement them with a new dataset for training. To avoid the known problem of poor labeling performance by human annotators in three-dimensional data, we utilize freely available 3D models of humans with corresponding animations from Mixamo \footnote{http://www.mixamo.com}. Then we used the open source software Blender \footnote{http://www.blender.org} to render images. The dataset is publicly available online.

Our dataset is built upon 20 different characters performing 39 actions. We split the data into a validation set (\textit{R-val}) and a training set (\textit{R-train}), where a character or action can exclusively be in one of the sets but not in the other. Our proposed split results into 16 characters performing 31 actions for training and 4 characters with 8 actions in the validation set.

For each frame we randomly sample a new camera location, which is roughly located in a spherical vicinity around one of the hands. All hand centers lie approximately in a range between 40 cm and 65 cm from the camera center. Both left and right hands are equally likely and the camera is rotated to ensure that the hand is at least partially visible from the current viewpoint. After the camera location and orientation are fixed, we randomly sample one background image from a pool of 1231 background images downloaded from Flickr\footnote{http://www.flickr.com}. Those images show different kinds of scenes from cities and landscapes. We ensured that they do not contain persons.

To maximize the visual diversity of the dataset, we randomize the following settings for each rendered frame: we apply lighting by 0 to 2 directional light sources and global illumination, such that the color of the sampled background image is roughly matched. Additionally we randomize light positions and intensities. Furthermore, we save our renderings using a lossy JPG compression with the quality factor of no compression up to 60\%. We also randomized the effect of specular reflections on the skin.

In total our dataset provides 41258 images for training and 2728 images for evaluation with a resolution of 320 × 320 pixels. All samples come with full annotation of a 21 keypoint skeleton model of each hand and additionally 33 segmentation masks are available plus the background. As far as the segmentation masks are concerned there is a class for the human, one for each palm and each finger is composed by 3 segments. Figure \ref{fig:dataset} shows a sample from the dataset. Every finger is represented by 4 keypoints: the tip of the finger, two intermediate keypoints and the end located on the palm. Additionally, there is a keypoint located at the wrist of the model. For each of the hand keypoints, there is information if it is visible or occluded in the image. Also keypoint annotations in the camera pixel coordinate system and in camera centered world coordinates are given. The camera intrinsic matrix and a ground truth depth map are available, too, but were not used in this work.

### 6. Experiments

We evaluated all relevant parts of the overall approach: (1) the detection of hand keypoints of the \textit{PoseNet} with and without the hand segmentation network; (2) the 3D hand pose estimation and the learned 3D pose prior. Finally, we applied the hand pose estimation to a sign language recognition benchmark.

#### 6.1. Keypoint detection in 2D

Table \ref{tab:2d} shows the performance of \textit{PoseNet} on 2D keypoint estimation. We report the average endpoint error (EPE) in pixels and the area under the curve (AUC) on the percentage of correct keypoints (PCK) for different error thresholds; see Figure \ref{fig:performance}.

![Figure 4: Our new dataset provides segmentation maps with 33 classes: three for each finger, palm, person, and background. The 3D kinematic model of the hand provides 21 keypoints per hand: 4 keypoints per finger and one keypoint close to the wrist.](http://www.blender.org)

![Figure 4: Our new dataset provides segmentation maps with 33 classes: three for each finger, palm, person, and background. The 3D kinematic model of the hand provides 21 keypoints per hand: 4 keypoints per finger and one keypoint close to the wrist.](http://www.flickr.com)

![Figure 4: Our new dataset provides segmentation maps with 33 classes: three for each finger, palm, person, and background. The 3D kinematic model of the hand provides 21 keypoints per hand: 4 keypoints per finger and one keypoint close to the wrist.](http://www.mixamo.com)

![Figure 4: Our new dataset provides segmentation maps with 33 classes: three for each finger, palm, person, and background. The 3D kinematic model of the hand provides 21 keypoints per hand: 4 keypoints per finger and one keypoint close to the wrist.](http://www.flickr.com)

<table>
<thead>
<tr>
<th></th>
<th>AUC</th>
<th>EPE median</th>
<th>EPE mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{GT}</td>
<td>\textit{R-val}</td>
<td>0.724</td>
<td>5.001</td>
</tr>
<tr>
<td></td>
<td>\textit{S-val}</td>
<td>0.817</td>
<td>5.522</td>
</tr>
<tr>
<td>\textit{Net}</td>
<td>\textit{R-val}</td>
<td>0.663</td>
<td>5.833</td>
</tr>
<tr>
<td></td>
<td>\textit{S-val}</td>
<td>0.762</td>
<td>5.528</td>
</tr>
<tr>
<td></td>
<td>\textit{Dexter}</td>
<td>0.489</td>
<td>13.684</td>
</tr>
</tbody>
</table>

Table 1: The top rows (\textit{GT}) report performance for the \textit{PoseNet} operating on ground truth cropped hand images. The bottom rows (\textit{Net}) show results when the hand crops are generated using \textit{HandSegNet}. \textit{PoseNet} was trained jointly on \textit{R-train} and \textit{S-train}, whereas \textit{HandSegNet} was only trained on \textit{R-train}. End point errors are reported in pixels with respect to the uncropped image and AUC is calculated over an error range from 0 to 30 pixels.
Figure 5: Exemplary 2D keypoint localization results. The first two columns show samples from Dexter, the following three depict R-val and the last one are samples from S-val.

Figure 6: Results on 2D keypoint estimation when using different training sets for PoseNet. Shown is the percentage of correct keypoints (PCK) over a certain threshold in pixels evaluated on Dexter. Jointly training on R-train and S-train yields the best results.

The first case shows the performance of PoseNet in isolation, while the second shows the performance of the complete 2D keypoint estimation. The difference between the median and the mean for the latter case show that HandSegNet is reliable in most cases but is sometimes not able to segment the hand correctly, which makes the 2D keypoint prediction fail.

The results show that the method works on our synthetic dataset (R-val) and the stereo dataset (S-val) equally well. The Dexter dataset is more difficult because the dataset is different from the training set and because of frequent occlusions of the hand by the handled cube. We did not have samples with occlusion (apart from self-occlusion) in the training set.

In Figure 6 we show that training on more diverse data helps cross-dataset generalization. While training only on our synthetic dataset R-train yields much better results on Dexter than training on the limited stereo dataset S-train, training on R-train and S-train together yields the best results. Figure 5 shows some qualitative results of this configuration. Additional examples are in the supplementary.

6.2. Lifting the estimation to 3D

6.2.1 Pose representation

We evaluated the proposed canonical frame representation for predicting the 3D hand pose from 2D keypoints by comparing it to several alternatives. All variants share a common base architecture that is identical to one stream of the PosePrior proposed in 4.3. They were trained on score maps c with a spatial resolution of 32 by 32 pixels. To avoid overfitting, we augmented the score maps by applying channelwise dropout with a drop probability of 0.2. This forces the networks to deal with incomplete score maps. Additionally we disturbed the keypoint location with Gaussian noise and randomly translated the keypoints by up to 2.5 pixel.

Table 2 shows the resulting end point errors per keypoint. The Direct approach tries to lift the 2D keypoints directly to the full 3D coordinates w\textsuperscript{rel} without using a canonical frame. This is disadvantageous, because it is difficult for the network to learn separate the global rotation of the hand from the articulation.

The Bottleneck approach is inspired by Oberweger et al. [12], who introduced a bottleneck layer before estimating the coordinates. We inserted an additional FC layer before the final FC output layer, parameterize it as in Oberweger et al. with 30 channels and linear activation. The outcome was not better than with the Direct approach.

Table 2: Average median end point error per keypoint of the predicted 3D pose for different lifting approaches given a noisy ground truth 2D pose. Networks were trained on R-train. The results are reported in mm and the subscript gives the relative performance to the proposed approach.

<table>
<thead>
<tr>
<th></th>
<th>Direct</th>
<th>Bottleneck</th>
<th>Local</th>
<th>NN</th>
<th>Prop.</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-train</td>
<td>20.2%</td>
<td>20.1%</td>
<td>35.1%</td>
<td>0.0%</td>
<td>18.5%</td>
</tr>
<tr>
<td>R-val</td>
<td>20.9%</td>
<td>11.2%</td>
<td>39.1%</td>
<td>26.9%</td>
<td>18.8%</td>
</tr>
</tbody>
</table>
Figure 7: Qualitative examples of our complete system. Input to the network are color image and the information if it’s a left or right hand. The network estimates the hand segmentation mask, localizes keypoints in 2D and outputs the most likely 3D pose. The samples on the left hand side are from a dataset we recorded for qualitative evaluation, on the top right hand side is a sample from the sign language dataset and the bottom right sample is taken from S-val. In the supplementary material we provide more qualitative examples.

Figure 8: The left most column shows the input image as gray scale with the input score map overlayed as red dots. Every row corresponds to a separate forward pass of the network. The two columns to the right visualize the predicted 3D structure of the network from different viewpoints in canonical coordinates. Ground truth is displayed in dashed green and the network prediction is shown in solid red.

The Local approach incorporates the kinematic model of the hand and uses the network to estimate articulation parameters of the model. We generalize [27] by estimating not only the angles but also the bone length. The network is trained to estimate two angles and one length per keypoint, which results in 63 parameters. The angles express rotations in a bone local coordinate system. This approach only works if the hand is always shown from the same direction, but cannot capture the global pose of the hand.

Finally, the NN approach matches the 2D keypoints to the most similar sample from the training set and retrieves the 3D coordinates from this sample [6]. While this approach trivially works best on the training set, it does not generalize well to new samples.

The generalization of the other approaches is quite good showing similar errors for both the training and the validation set. The proposed approach from 4.3 worked best and was used for the following experiments.

6.2.2 Analysis of the learned prior

To examine the 3D prior learned by the network we input score maps that lack keypoints and Figure 8 shows the 3D pose prediction from two different viewpoints. The extreme case, with no keypoints provided as input at all, shows the canonical prior learned by the network. As more keypoints

Figure 9: Results for our complete system on S-val compared to approaches from [25] and [26]. Shown is the percentage of correct keypoints (PCK) over respective thresholds in mm. PoseNet and PosePrior are trained on S-train and R-train, whereas the HandSegNet is trained on R-train.
are added, the network adjusts the predicted pose to this additional evidence. This experiment also simulates the situation of occluded 2D keypoints and demonstrates that the learned prior allows the network to still retrieve reasonable poses.

### 6.2.3 Comparison to literature

Since there is no work on 3D hand pose estimation from RGB images yet, we cannot compare to alternative approaches. To still relate our results coarsely to literature, we compare them to Zhang et al. [25], who provide results in mm for state-of-the-art 3D hand pose tracking on depth data. They run their experiments on the stereo dataset S-val, which also contains RGB images. Since in contrast to Zhang et al. our approach does not use the depth data, it still comes with ambiguities with regard to scale and absolute depth. Thus, we accessed the absolute position of the root keypoint and the scale of the hand to shift and scale our predicted 3D hand pose, which yields metric world coordinates \( w \) by using (1) and (2). For this experiment we trained PosePrior on score maps predicted by PoseNet using the same schedule as for the experiment in section 6.2.2. PoseNet is trained separately as described in [6, 1] and then kept fixed. Figure 8 shows that our approach largely outperforms the approaches presented in Zhang et al. [25] although we use the depth map only for rescaling and shifting in the end. Additionally we report results of the lifting approach presented by Zhao et al. [26] in conjunction with our PoseNet, which we train in a similar manner. Results are inferior to the proposed PosePrior. We believe the reason is that using score maps as input for the lifting is advantageous over coordinates, because it can handle ambiguities in hand keypoint detection. Qualitative 3D examples on three different datasets with the complete processing pipeline are shown in Figure 7.

### 6.3. Sign language recognition

Previous hand pose estimation approaches depending on depth data cannot be applied to most sign language recognition datasets, as they only come with color images. As a last exemplary experiment, we used our hand pose estimation system and trained a classifier for gesture recognition on top of it. The classifier is a fully connected three layer network with ReLU activation functions; c.f. the supplemental material for the network details.

We report results on the so-called RWTH German Fingerspelling Database [8]. It contains 35 gestures representing the letters of the alphabet, German umlauts, and the numbers from one to five. The dataset comprises 20 different persons, who did two recordings each for every gesture. Most of the gestures are static except for the ones for the letters J, Z, A, Õ, and Ù, which are dynamic. In order to keep this experiment simple, we ran the experiments on the subset restricted to 30 static gestures.

The database contains recordings by two different cameras, but we used only one camera. The short video sequences have a resolution of \( 320 \times 240 \) pixels. We used the middle frame from each video sequence as color image and the gesture class labels as training data. This dataset has 1160 images, which we separated by signers into a validation set with 232 images and a training set with 928 images. We resized image to \( 320 \times 320 \) pixels and trained on randomly sampled \( 256 \times 256 \) crops. Because the images were taken from a compressed video stream they exhibit significant compression artifacts previously unseen by our networks. Thus, we labeled 50 images from the training set with hand keypoints, which we used to fine-tune our PoseNet upfront. Afterwards the pose estimation part is kept fixed and we solely train the GestureNet. Table 3 shows that our system archives comparable results to Dreuw et al. [8] on the subset of gestures we used for the comparison.

### 7. Conclusions

We have presented the first learning based system to estimate 3D hand pose from a single image. We contributed a large synthetic dataset that enabled us to train a network successfully on the task. We have shown that the network learned a 3D pose prior that allows it to predict reasonable 3D hand poses from 2D keypoints in real world images. While the performance of the network is even competitive to approaches that use depth maps, there is still much room for improvements. The performance seems mostly limited by the lack of an annotated large scale dataset with real-world images and diverse pose statistics.

### Acknowledgements

We gratefully acknowledge funding by the Baden-Württemberg Stiftung as part of the projects ROTAH and RatTrack. Also we thank Nikolaus Mayer, Benjamin Ummenhofer and Maxim Tatarchenko for valuable ideas and many fruitful discussions.

<table>
<thead>
<tr>
<th>Method</th>
<th>Word error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dreuw et al. [8]</td>
<td>35.7 %</td>
</tr>
<tr>
<td>Dreuw on subset</td>
<td>36.56 %</td>
</tr>
<tr>
<td>Ours 3D</td>
<td>33.2 %</td>
</tr>
</tbody>
</table>

Table 3: Word error rates in percent on the RWTH German Fingerspelling Database subset of non dynamic gestures. Results for Dreuw et al. [8] on the subset from [1].
References


