



Occlusion-Aware Hand Pose Estimation Using Hierarchical Mixture Density Network

Qi Ye^(✉) and Tae-Kyun Kim

Imperial College London, London, UK
q.ye14@imperial.ac.uk

Abstract. Learning and predicting the pose parameters of a 3D hand model given an image, such as locations of hand joints, is challenging due to large viewpoint changes and articulations, and severe self-occlusions exhibited particularly in egocentric views. Both feature learning and prediction modeling have been investigated to tackle the problem. Though effective, most existing discriminative methods yield a single deterministic estimation of target poses. Due to their single-value mapping intrinsic, they fail to adequately handle self-occlusion problems, where occluded joints present multiple modes. In this paper, we tackle the self-occlusion issue and provide a complete description of observed poses given an input depth image by a novel method called hierarchical mixture density networks (HMDN). The proposed method leverages the state-of-the-art hand pose estimators based on Convolutional Neural Networks to facilitate feature learning, while it models the multiple modes in a two-level hierarchy to reconcile single-valued and multi-valued mapping in its output. The whole framework with a mixture of two differentiable density functions is naturally end-to-end trainable. In the experiments, HMDN produces interpretable and diverse candidate samples, and significantly outperforms the state-of-the-art methods on two benchmarks with occlusions, and performs comparably on another benchmark free of occlusions.

Keywords: 3D hand pose estimation · Occlusion
Multi-valued mapping · Convolutional Neural Network
Mixture density network

1 Introduction

3D hand pose estimation has shown an increasing interest with commercial miniaturized RGBD cameras and its ubiquitous applications in virtual/augmented reality (VR/AR) [13], sign language recognition [3, 47], activity

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01249-6_49) contains supplementary material, which is available to authorized users.

recognition [29], and man-machine interfaces for robots and autonomous vehicles. There are generally two typical camera settings: a third-person viewpoint, where the camera is set in front of the user, and an egocentric (or first-person) viewpoint, where the camera is mounted on the user’s head (in VR glasses, for example), or shoulder. While both settings share challenges like the full range of 3D global rotations, complex articulations, self-similar parts of hands, self-occlusions are more dominant in the egocentric viewpoints. Most existing hand benchmarks are collected in the third-person viewpoints, e.g. the two widely used ICVL [38] and NYU [41] have less than 9% occluded finger joints.

Discriminative methods (cf. generative model fitting) in hand pose estimation learn a mapping from an input image to pose parameters from a large training dataset, and have been very successful in the settings of third-person viewpoints. However, they fail to handle occlusions frequently encountered in egocentric viewpoints. They treat the mapping to be single-valued, not being aware of that an input image may have multiple pose hypotheses when occlusions occur. See Fig. 1 where an example image and its multiple pose labels from the BigHand dataset [48] are shown.

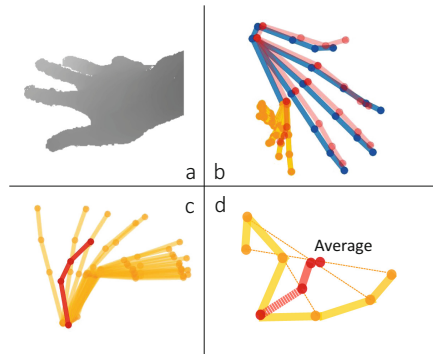


Fig. 1. (a) A hand depth image with the pinky finger occluded. (b) Multiple pose labels (visible joints are in blue and occluded joints in yellow) and the predicted pose (in red) by CNN trained using a mean squared error. (c) A closer look of the multiple labels and the CNN prediction for the occluded joints. (d) The average of two labels yields a physically implausible pose. (Color figure online)

Given a set of hand images and their pose labels i.e. 3D joint locations, discriminative methods such as Convolutional Neural Networks (CNN) minimize a mean squared error function, and the minimization of such error functions typically yields the averages of joint locations conditioned on input images. When all finger joints in the images are visible, the mapping is single-valued and the conditional average is correct, though the average only provides a limited description of the joint locations. However, for the occlusion cases, which happen frequently in the egocentric and hand-object interaction scenarios [7, 22–24], the mapping is multi-valued due to occluded joints that exhibit multiple locations given the

same images. The conditional average of the joint locations is not necessarily a correct pose, as shown in Fig. 1b and c (The skeletons are shown in a 3D rotated view to better illustrate the problem, same for the other 3D skeletons shown in the paper). The prediction of a CNN trained by the mean squared error function is shown in red. It is interpretable and close to the ground truth for the visible joints, whereas it is physically implausible and not close to any of the given poses for the occluded joints. The example is clearer in Fig. 1d, where we are given two available poses for the same image and CNN trained with the mean squared error function produces the pose estimation in red.

Existing discriminative methods, including the above CNN, are mostly deterministic, i.e. their outputs are single poses, thus lacking the description of all available joint locations. A discriminative method often serves as the initialization of a generative model fitting in the hybrid pose estimation approaches [31, 36]. If the discriminative method yields a probability distribution that well fits the data, than a single deterministic output, it would allow sampling pose hypotheses from its distribution. This, in turn, reduces the search space, helping a faster convergence, and avoids local minima from diverse candidates in the model fitting. Such sampling is crucial also for multi-stage pose estimation [36] and hand tracking [21]. Previous methods ignore the pose space to be explored ahead and their optimization frameworks are not aware of occlusions.

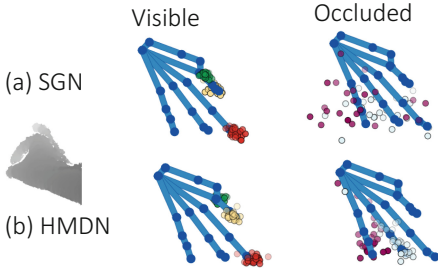


Fig. 2. Samples drawn from the distributions of SGN and HMDN for finger tips.



Fig. 3. Hand images under self-occlusions exhibiting multiple pose labels.

In this paper, hierarchical mixture density networks (HMDN) are proposed to give a complete description of hand poses given images under occlusions. The probability distribution of joint locations is modeled in a two-level hierarchy to consider both single- and multi-valued mapping conditioned on the joint visibility. The first level represents the distribution of a latent variable for the joint visibility, while the second level the distribution of joint locations by a single Gaussian model for visible joints or a Gaussian mixture model for occluded joints. The hierarchical mixture density is topped upon the CNN output layer, and the whole network is trained end-to-end with the differentiable density functions. See Fig. 2. The distribution of the proposed method HMDN captures diverse joint locations in a compact manner, compared to the network

that learns a single Gaussian distribution (SGN). To the best of our knowledge, HMDN is the first solution that has its estimation in the form of a conditional probability distribution with the awareness of occlusions in 3D hand pose estimation. The experiments show that the proposed method significantly improves several baselines and state-of-the-art methods under occlusions given the same number of pose hypotheses.

2 Related Work

2.1 Pose Estimation Under Occlusion

For free hand motions, methods explicitly tackling self-occlusions are rare as most existing datasets are collected in third-person viewpoints and the proportion of occluded joints is small. Mueller et al. [16] observed that many existing methods fail to work under occlusions and even some commercial systems claiming for egocentric viewpoints often fail under severe occlusions. Methods developed for hand-object interactions [23, 33, 42], where occlusions happen frequently, model hands and objects together to resolve the occlusion issues. Jang et al. [13] and Rogez et al. [28] exploit pose priors to refine the estimations. Mueller et al. [16] and Rogez et al. [27] generate synthetic images to train discriminative methods for difficult egocentric views.

In human body pose estimation and object keypoint detection, occlusions are tackled more explicitly [4, 5, 8, 10, 12, 17, 26, 32]. Chen et al. [5] and Ghiasi et al. [8] learn templates for occluded parts. Hsiao et al. [12] construct a occlusion model to score the plausibility of occluded regions. Rafi et al. [26] and Wang et al. [44] utilize the information in backgrounds to help localize occluded keypoints. Charles et al. [4] evaluate automatic labeling according to occlusion reasoning. Haque et al. [10] jointly refine the prediction for visible parts and visibility mask in stages. Navaratnam et al. [17] tackle the multi-valued mapping for 3d human body pose via marginal distributions which help estimate the joint density.

The existing methods do not address multi-modalities nor do not model the difference in distributions of visible and occluded joints. For CNN-based hand pose regression [19, 20, 41, 46], the loss function used is the mean squared error, bringing in the aforementioned issues under occlusions. For random forest-based pose regression [31, 34, 38], the estimation is made from the data in leaf nodes and it is convenient to fit a multi-modal model to the data. However, with no information of which joints are visible or occluded, the data in all leaf nodes is captured either by the mean-shift (a uni-modal distribution) or a Gaussian Mixture Model (GMM) [36].

2.2 Mixture Models

Mixture density networks (MDN) were first proposed in [1] to enable neural networks to overcome the limitation of the mean squared error function by producing a probability distribution. Zen et al. [49] use MDN for acoustic modeling

and Kinoshita et al. [15] for speech feature enhancement. Variani [43] proposes to learn the features and the GMM model jointly. All these work apply MDN to model acoustic signals without an adaptation of the mixture density model. In addition to applying MDN to model the hand pose space when multiple modes exist due to occlusion, we extend MDN by a two-level hierarchy to fit the specific mixture of single-valued and multi-valued problems, for the application of hand pose estimation under occlusions. To model data under noise, a similar hierarchical mixture model is proposed in [6] to represent “useful” data and “noise” by different sub-components, and a Bayesian approach is used to learn the parameters of the mixture model. Different from the work, we model a conditional distribution and use CNN to discriminatively learn the model parameters.

3 Hierarchical Mixture Density Network

3.1 Model Representation

The dataset to learn the model consists of $\{x_n, Y_n^d, v_n^d | n = 1, \dots, N, d = 1, \dots, D\}$, where x_n , Y_n^d , and v_n^d denote the n -th hand depth image, the multiple pose labels i.e. 3D locations of the d -th joint of the n -th image, and the visibility label of the d -th joint of the n -th image, respectively. The d -th joint is associated with multiple labels $Y_n^d = \{y_{nm}^d\}$, where $y_{nm}^d \in R^3$ is the m -th label i.e. 3D location. See Fig. 3 for examples. Each shows different example labels overlaid on the same depth image (in the first three columns), and all available labels in a 3D rotated view (in the last column). Visible joints are in blue and occluded joints in other colors. The visibility label is binary, indicating whether the d -th joint of the n -th image is visible or not. We treat D joints independently.

To model hand poses under occlusions, a two-level hierarchy is considered. The top-level takes the visibility label, and the bottom-level switches between a uni-modal distribution and a multi-modal distribution, depending on the joint visibility.

The binary label or variable v_n^d follows the Bernoulli distribution,

$$p(v_n^d | w_n^d) = (w_n^d)^{v_n^d} (1 - w_n^d)^{(1-v_n^d)}, \quad (1)$$

where w_n^d is the probability that the joint is visible. As existing hand benchmarks do not provide the joint visibility labels, we use a sphere model similar to [25] to generate the visibility labels from the available pose labels. The sphere centers are obtained from the joint locations and depth image pixels are assigned to the nearest spheres. Hand joints whose spheres have the number of pixels below a threshold are labeled as occluded. See Fig. 4. The visibility labels v_n^d are used for training, and they are inferred at testing.

When $v_n^d = 1$, the joint is visible in the image and the location is deterministic. Considering the label noise, y_{nm}^d is generated from a single Gaussian distribution,

$$p(y_{nm}^d | v_n^d = 1) = \mathcal{N}(y_{nm}^d; \mu_n^d, \sigma_n^d). \quad (2)$$

When the joint is occluded i.e. $v_n^d = 0$, it has multiple labels and they are drawn from a Gaussian Mixture Model (GMM) with J components

$$p(y_{nm}^d | v_n^d = 0) = \sum_{j=1}^J \pi_{nj}^d \mathcal{N}(y_{nm}^d; \epsilon_{nj}^d, s_{nj}^d), \tag{3}$$

where ϵ_{nj}^d and s_{nj}^d represent the center and standard deviation of the j -th component. The location y_{nm}^d is drawn from the j -th component dependent on a hidden variable z_{nj}^d , where $z_{nj}^d \in \{0, 1\}$ and $\sum_{j=1}^J z_{nj}^d = 1$. The hidden variable is under the distribution $p(z_{nj}^d) = \prod_{j=1}^J (\pi_{nj}^d)^{z_{nj}^d}$, where $0 \leq \pi_{nj}^d \leq 1$, $\sum_{j=1}^J \pi_{nj}^d = 1$.



Fig. 4. Left: the hand sphere model; Right: examples with pixels assigned to different parts

With all components defined, the distribution of the joint location conditioned on the visibility is

$$p(y_{nm}^d | v_n^d) = [\mathcal{N}(y_{nm}^d; \mu_n^d, \sigma_n^d)]^{v_n^d} \left[\sum_{j=1}^J \pi_{nj}^d \mathcal{N}(y_{nm}^d; \epsilon_{nj}^d, s_{nj}^d) \right]^{(1-v_n^d)} \tag{4}$$

and the joint distribution of y_{nm}^d and v_n^d is

$$p(y_{nm}^d, v_n^d) = [w_n^d \mathcal{N}(y_{nm}^d; \mu_n^d, \sigma_n^d)]^{v_n^d} \left[(1 - w_n^d) \sum_{j=1}^J \pi_{nj}^d \mathcal{N}(y_{nm}^d; \epsilon_{nj}^d, s_{nj}^d) \right]^{(1-v_n^d)}. \tag{5}$$

Equation (4) shows that the generation of joint locations y_{nm}^d given the input image x_n is in a two-level hierarchy: first, a sample v_n^d is drawn from Eq. (1) and then, depending on v_n^d , a joint location is drawn either from a uni-modal Gaussian distribution or GMM. Thus, the proposed model switches between the two cases and provides a full description of hand poses under occlusions. The joint distribution in Eq. (5) is used to define the loss function in Sect. 3.3.

3.2 Architecture

The formulations in the previous section are presented for the d -th joint y_{nm}^d . For all D joints of hands, the distribution is obtained by multiplying the distributions

of independent joints. The observed hand poses and the joint visibility, given x_n , are drawn from $\prod_{d=1}^D \prod_m p(y_{nm}^d, v_n^d)$.

Note that the hierarchical mixture density in Eq. (4) and the joint distribution in Eq. (5) are conditioned on x_n . All model parameters are in a functional form of x_n and the joint distribution in Eq. (5) is differentiable. We choose to learn these functions by a CNN and the distribution is parameterized by the output of the CNN. As shown in Fig. 5, the input of the CNN is an image x_n and the outputs are the HMDN parameters: $w_n^d, \mu_n^d, \sigma_n^d, \epsilon_{nj}^d, s_{nj}^d, \pi_{nj}^d$, for $d = 1, \dots, D$ and $j = 1, \dots, J$. The output parameter parameters consist of three parts. w_n^d is the visibility probability in Eq. (1), μ_n^d, σ_n^d for the uni-modal Gaussian in Eq. (2), and $\epsilon_{nj}^d, s_{nj}^d, \pi_{nj}^d$ for the GMM in Eq. (3). Different activation functions are used to meet the defined ranges of parameters. For instance, the standard deviations σ_n^d and s_{nj}^d are activated by an exponential function to remain positive and π_{nj}^d by a softmax function to be in $[0, 1]$.

The prediction of the visibility, the value of w_n^d , is used to compute the visibility loss over the visibility label v_n^d . See Sect. 3.3. Depending on the visibility label v_n^d , the parameters of the uni-modal Gaussian (for visible joints) or GMM (for occluded joints) are chosen to compute the loss, as shown in blue and in orange respectively in Fig. 5.

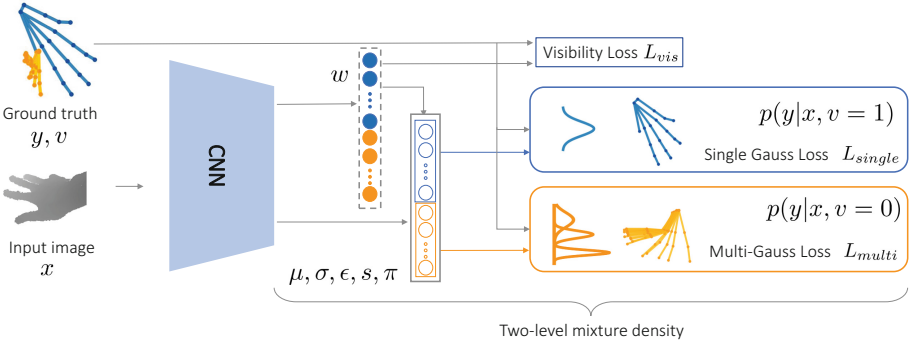


Fig. 5. Hierarchical Mixture Density Network. Hand joint locations y given the input image x are modeled in a two-level hierarchy: in the first level, the visibility is modeled by Bernoulli distribution whose parameter is w ; then depending on the visibility, the joint locations are either modeled by uni-modal Gaussian distributions (visible joints, shown in blue) or GMMs (occluded joint, shown in orange). The CNN outputs the parameters of HMDN, i.e. $w, \mu, \sigma, \epsilon, s, \pi$. (Color figure online)

3.3 Training and Testing

The likelihood for the entire dataset $\{x_n, Y_n^d, v_n^d | n = 1, \dots, N, d = 1, \dots, D\}$ is computed as $P = \prod_{n=1}^N \prod_{d=1}^D \prod_m p(y_{nm}^d, v_n^d)$, where $p(y_{nm}^d, v_n^d)$ in (5) has the model

parameters dependent on x_n . Thus, our goal is to learn the neural networks that yield the parameters that maximize the likelihood on the dataset. We use the negative logarithmic likelihood as the loss function.

$$L = -\log P = \sum_{n=1}^N \sum_{d=1}^D \sum_m \{L_{vis} + L_{single} + L_{multi}\}, \quad (6)$$

where

$$L_{vis} = -v_n^d \log(w_n^d) - (1 - v_n^d) \log(1 - w_n^d), \quad (7)$$

$$L_{single} = -v_n^d \log(\mathcal{N}(y_{nm}^d; \mu_n^d, \sigma_n^d)), \quad (8)$$

$$L_{multi} = -(1 - v_n^d) \log\left(\sum_{j=1}^J \pi_{nj}^d \mathcal{N}(y_{nm}^d; \epsilon_{nj}^d, s_{nj}^d)\right). \quad (9)$$

The three loss functions correspond to the three branches in Fig. 5. The visibility loss L_{vis} is computed using the predicated value of w_n^d . When $v_n^d = 1$, $L_{multi} = 0$ and L_{single} is calculated, and when $v_n^d = 0$, vice versa.

During testing, when an image x_n is fed into the network, the prediction for the d -th joint location is diverted to different branches according to the prediction of the visibility probability w_n^d . If w_n^d is larger than 0.5, the prediction (or sampling) for the location is made by the uni-modal Gaussian distribution in Eq. (2); otherwise, the GMM in Eq. (3).

However, when the prediction for the visibility is erroneous, the prediction for the joint location will be wrong. To help the bias problem, instead of using the binary visibility labels v_n^d to compute the likelihood, we use the samples drawn from the estimated distribution in Eq. (1) during training. When the number of samples is large enough, the mean of these samples becomes w_n^d . So, the losses in Eqs. (8) and (9) change to

$$L_{single} = -w_n^d \log(\mathcal{N}(y_{nm}^d; \mu_n^d, \sigma_n^d)), \quad (10)$$

$$L_{multi} = -(1 - w_n^d) \log\left(\sum_{j=1}^J \pi_{nj}^d \mathcal{N}(y_{nm}^d; \epsilon_{nj}^d, s_{nj}^d)\right). \quad (11)$$

The modified losses in Eqs. (10) and (11) can be seen as a soft version of the original ones Eqs. (8) and (9).

3.4 Degradation into Mixture Density Network

HMDN degrades into Mixture Density Network (MDN), without the supervision for learning the visibility variable. The other form of (4) is

$$p(y_{nm}^d | w_n^d) = w_n^d \mathcal{N}(y_{nm}^d; \mu_n^d, \sigma_n^d) + (1 - w_n^d) \left[\sum_{j=1}^J \pi_{nj}^d \mathcal{N}(y_{nm}^d; \epsilon_{nj}^d, s_{nj}^d) \right] \quad (12)$$

where the visibility probability w_n^d is learned with visibility labels. When the labels are not available, the above equation becomes

$$p(y_{nm}^d) = \sum_{j=1}^{J+1} \bar{\pi}_{nj}^d \mathcal{N}(y_{nm}^d; \bar{\epsilon}_{nj}^d, \bar{s}_{nj}^d) \quad (13)$$

where $\bar{\pi}_{nJ+1}^d = w_n^d$, $\bar{\epsilon}_{nJ+1}^d = \mu_n^d$, $\bar{s}_{nJ+1}^d = \sigma_n^d$, and $\bar{\pi}_{nj}^d = (1 - w_n^d)\pi_{nj}^d$, $\bar{\epsilon}_{nj}^d = \epsilon_{nj}^d$, $\bar{s}_{nj}^d = s_{nj}^d$ for $j = 1, \dots, J$. The visibility probability w_n^d in (12) is absorbed into the GMM mixing coefficients $\bar{\pi}_{nj}^d$, and the distribution becomes a GMM with $J + 1$ components with no dependency on the visibility.

4 Experiments

4.1 Datasets

Public benchmarks for hand pose estimation are mostly collected in third-person viewpoints and do not offer plenty of occluded joints with multiple pose labels. We investigate four datasets, ICVL [37], NYU [41], MSHD [31] and BigHand [48], and exploit those containing a higher portion of occluded joints in the following experiments. The rate of occluded finger joints and the total number of training and testing images are listed in Table 1.

The images in these datasets are paired with pose labels i.e. joint locations, without the visibility information of the finger joints. As explained in Sect. 3.1, we use the sphere model to generate the visibility labels for training HMDN.

Table 1. The rate of occluded finger joints and the total number of frames

Dataset	ICVL	NYU	MSHD	EgoBigHand
Train (rate/total no.)	0.06/16,008	0.09/72,757	0.33/100,000	0.48/969,600
Test (rate/total no.)	0.01/1,596	0.36/8,252	0.16/2,000	0.24/33,468

The BigHand dataset consists of two subparts: the egocentric subset includes lots of self-occlusions but lacks diverse articulations; the third-person viewpoint subset spans the full articulation space while the proportion of occluded joints, especially severe occlusions, is low. We augment the egocentric subset using the articulations of the third-person view dataset, and use it called EgoBigHand for experiments. EgoBigHand includes 8 subjects: frames of 7 subjects are used for training and frames of 1 subject for testing.

More results are also shown on MSHD and NYU datasets.

4.2 Self-comparisons

The baseline of our comparison is Single Gaussian Network (SGN), which is the CNN trained with a uni-modal Gaussian distribution. In [2], it is shown that maximization of the likelihood function under a uni-modal Gaussian distribution for a linear model is equivalent to minimizing the mean squared error errors. In our experiments, we observed that the estimation error of SGN using the Gaussian center is about the same as that of the CNN trained with the mean squared error. For further comparisons under the probabilistic framework, we report the accuracies of SGN.

We also report the experiments of MDN as in the previous section, we showed that HMDN degrades to MDN when there is no visibility label available in training. To compare MDN with HMDN fairly, the number of Gaussian components of MDN is set $J + 1$ and that of GMM branch of HMDN is J .

The CNN network used is the U-net proposed in [30], by adapting the final layers to fully connected layers for regression. All the networks are trained using Adam [14] and the convergence times of all methods above took about 24 h using Geforce GTX 1080Ti.

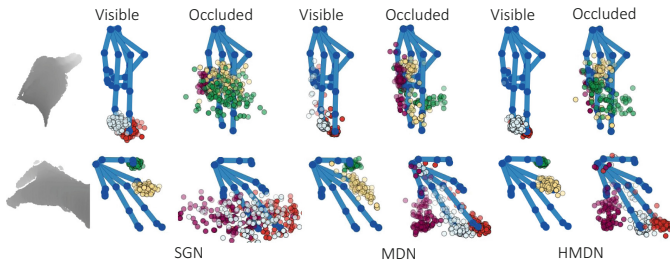


Fig. 6. Samples drawn from the distributions of SGN, MDN and HMDN for finger tips, shown in comparison to a pose label.

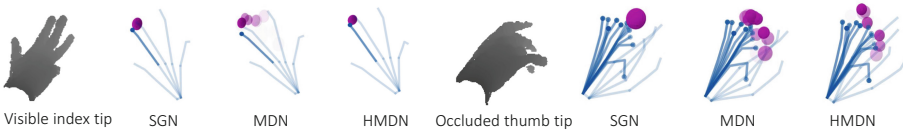


Fig. 7. Distributions predicted by SGN, MDN and HMDN for visible index tip and occluded thumb tip. Each magenta sphere represents a Gaussian component whose radius is the standard deviation and center is the mean. The degree of transparency is in proportion to the mixture coefficients $\{\pi\}$. (Color figure online)

Qualitative Analyses. See Fig. 6. 100 samples for each finger tip are drawn from the distributions of the different methods. HMDN is motivated by the

intrinsic mapping difference: single-valued mapping for visible and multi-valued mapping for occluded joints. Our results, shown in Fig. 6, demonstrate its ability of modeling this difference by producing interpretable and diverse candidate samples accordingly. For visible joints, SGN and HMDN produce the samples distributed in a compact region around the ground truth location, while the samples from MDN scatter in a larger area. For occluded joints, while the samples produced by SGN scatter in a broad sphere range, the samples produced by HMDN and MDN form an arc-shaped region, which indicates the movement range of finger tips within the kinematic constraints.

With the aid of visibility supervision, HMDN handles well the self-occlusion problem by tailoring different density functions to the respective cases. The examples of the distributions predicted by SGN, MDN and HMDN for visible and occluded joints are shown in Fig. 7. The resulting compact distributions that fit both visible joints and occluded joints improve the pose prediction accuracies in the following quantitative analyses. Such compact and interpretable distributions are also helpful for hybrid methods [31, 36]. For the discriminative-generative pipelines, the distribution largely reduces the space to be explored and produces diverse candidates to avoid being stuck at local minima in the generative part. For hand tracking methods [21], the distributions of occluded joints can be combined with the motion information e.g. speed and direction, to give a sharper i.e. more confident response at a certain location. The model can also find its application in multi-view settings.

Table 2. Estimation errors of different models. *see text for the evaluation metric used.

No. of Gauss. (J)	1	10		20		30	
Model	SGN	MDN	HMDN	MDN	HMDN	MDN	HMDN
Vis. Err. (mm)	32.8	32.2	30.5	34.0	30.7	32.6	30.5
Occ. Err. (mm)	36.5	35.4	34.8	36.4	34.4	35.6	34.2
*Occ. Err. (mm)	38.9	34.8	34.6	35.1	34.2	35.0	34.5

Quantitative Analyses. One hypothesis is drawn from the distribution of each method and is compared with the pose label, i.e. the ground truth joint location to measure the displacement error (in mm). The average errors are reported for visible joints and occluded joints separately in Table 2. Figure 8a presents the comparisons under the commonly used metric, the proportion of joints within an error threshold [31, 36, 46], using $J = 20$ in MDN/HMDN. HMDN outperforms both MDN and SGN for visible and occluded joints using the different numbers of Gaussian components. For occluded joints, HMDN improves SGN by 10% in the percentage of joints within the error 20 mm (Fig. 8a), and by about 2 mm in the mean displacement error (Table 2). HMDN also outperforms the baselines for visible joints. One can reason that given the limited network capacity, by

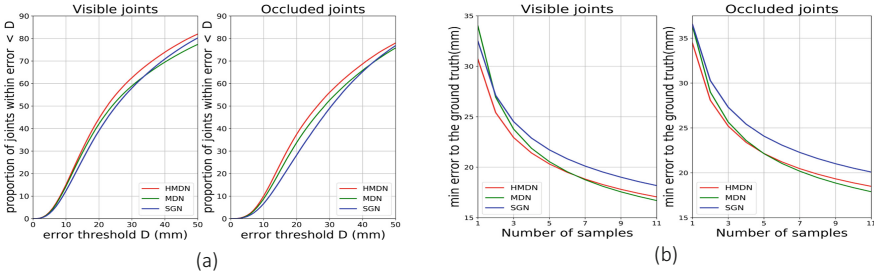


Fig. 8. Comparison of HMDN, SGN, MDN, when $J = 20$.

specifying density functions by data types, HMDN learns to take a better balance between the visible and occluded, while maximizing the likelihood of the entire training data. As shown in Table 2, the estimation errors of HMDN do not change much for $J = 10, 20, 30$. Note, however, the number of model parameters linearly increases with J .

In Fig. 8b, we vary the number of samples drawn from the distributions, and measure the minimum distance error. HMDN consistently achieves lower errors than SGN at all numbers of samples. Compared to MDN, HMDN appears better at the smaller numbers of samples. When the number of samples increases, the error gap between the two methods becomes small.

In both Table 2 and Fig. 8, we repeated the sampling process 100 times and reported the mean accuracies. The standard deviations were fairly small as: 0.03–0.04 mm for occluded joints, and 0.01–0.02 mm for visible joints.

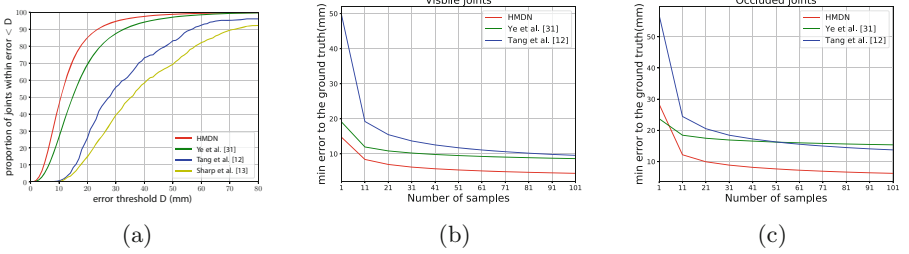
As our motivation is in modeling the distribution of joint locations, we measure how well the predicted distribution aligns with the target distribution. As shown in Fig. 3, multiple pose labels are gathered for the same image with occlusions. We draw multiple samples from the predicted distribution and measure the minimum distance between the set of drawn samples and the set of pose labels. As shown in the last row of Table 2, the improvement is significant. Both MDN and HMDN outperform SGN by about 4 mm, which demonstrates that the arc-shaped distributions produced by MDN/HMDN align better with the target joint locations than the sphere-shaped distribution produced by SGN, as shown in Fig. 6. Instead of the minimum distance, we could use other similarity measures between distributions.

Though the improvement of HMDN over MDN is marginal, the number of parameters for the density is largely reduced by the awareness of occlusion in HMDN when joints are visible.

Bias. In Sect. 3.3, we proposed to mitigate the exposed bias during testing, by sampling from the visibility distribution at training. HMDN trained with the loss functions in Eqs. (8) and (9), is denoted as $\text{HMDN}_{\text{hard}}$, while the one trained with Eqs. (10) and (11) is $\text{HMDN}_{\text{soft}}$. In Table 3 $\text{HMDN}_{\text{soft}}$ consistently achieves lower errors than $\text{HMDN}_{\text{hard}}$ for different numbers of Gaussian components.

Table 3. Comparison of $\text{HMDN}_{\text{hard}}$ and $\text{HMDN}_{\text{soft}}$

No. of Gauss. (J)	10		20		30	
Model	$\text{HMDN}_{\text{hard}}$	$\text{HMDN}_{\text{soft}}$	$\text{HMDN}_{\text{hard}}$	$\text{HMDN}_{\text{soft}}$	$\text{HMDN}_{\text{hard}}$	$\text{HMDN}_{\text{soft}}$
Vis. Err. (mm)	32.2	30.5	32.9	30.7	33.1	30.5
Occ. Err. (mm)	35.8	34.8	35.9	34.4	36.4	34.2

**Fig. 9.** Comparison of HMDN with prior work.

4.3 Comparison with the State-of-the-Arts

To compete with state-of-the-arts, the following strategies are adopted: first, a CNN network is trained to estimate the global rotation and translation, and conditioned on the estimation, HMDN is then trained; data augmentation, including translation, in-plane rotation, and scaling is used.

MSSH Dataset. MSSH has a considerable number of occluded joints both in training and testing set. We compare HMDN with three methods: Ye et al. [46], Tang et al. [36], Sharp et al. [31]. For [31], the results of its discriminative part are used. Figure 9a shows the proportion of joints within different error thresholds for the four methods, where a single prediction is used from HMDN.

In Fig. 9b and c, we further compare Ye et al. [46] and Tang et al. [36] with HMDN, by varying the number of hypotheses i.e. samples from the output distributions, and measuring the minimum displacement errors. Ye et al. [46] use a deterministic CNN. To produce multiple samples, they jitter around the CNN prediction, which can be treated as a uni-modal Gaussian. Tang et al. [36] use decision forests (3 trees) and the data points in the leaf nodes are modeled by GMM with 3 components. During testing, samples are drawn from GMMs of all trees. We used the original codes from the authors in our experiments.

HMDN significantly outperforms both methods for visible joints. For occluded joints, when the number of samples is 1, the errors of HMDN and Ye et al. [46] are close. However, Ye et al. [46] are not able to produce diverse samples to reach low errors as HMDN when the number of samples increases. Tang et al. [36] provide diverse candidates by GMM in its leaf nodes, but the variance of the distribution is much larger than that of Ye et al. [46] and HMDN for both visible and occluded joints. From the results, HMDN demonstrates its superiority for both the unimodal Gaussian model and GMM: the compact dis-

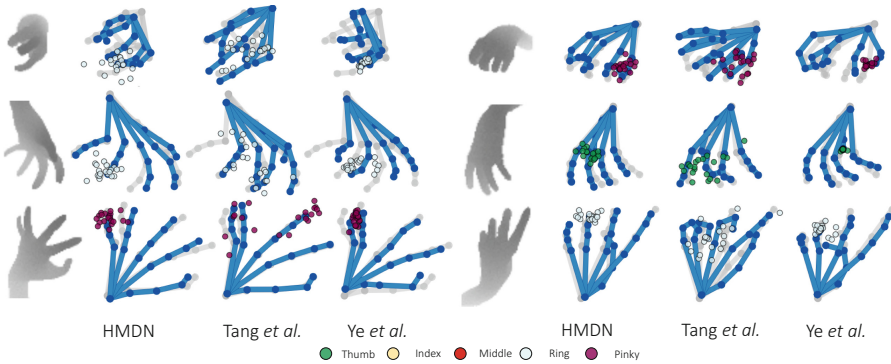


Fig. 10. Comparison of HMDN with Tang et al. [36] and Ye et al. [46]. Ground truth: skeletons in gray. Predictions from the models: skeletons in blue. For each image, samples for one tip joint from the three methods are scattered along the skeletons. Visible joints in the left column and occluded joints in the right column. (Color figure online)

tribution with lower bias for visible joints and the diverse samples yet having smaller variances for occluded joints. See Fig. 10 for example results. The samples from Tang et al. [36] for the finger tips spans a large region; those from Ye et al. [46] are more compact but many deviate from the ground truth.

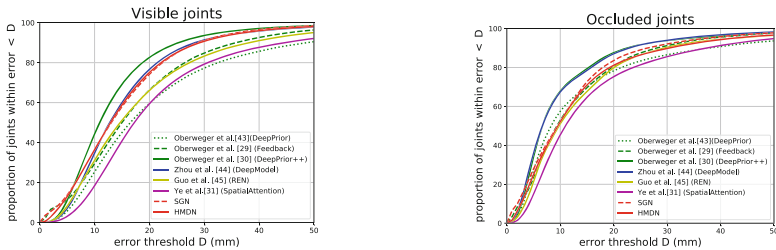


Fig. 11. Comparison with state-of-the-art approaches on NYU dataset.

NYU Dataset. The proposed method has also been evaluated on NYU dataset. Most joints in the training set are visible while on the testing set, there are up to 36% occluded joints. This implies all the joints in the testing dataset will be predicted as visible joints. Despite the ill-setting for HMDN, the method does not fail but degrades into SGN: the performances of SGN and HMDN are similar as shown in Fig. 11, and when compared with various state-of-the-arts based on CNN [9, 18–20, 46, 50], HMDN is in the second place for visible joints and third place for occluded joints. Note the best method [20] uses a 50-layer ResNet model [11] and 21 more CNN models to refine the estimation.

5 Conclusion

This paper addresses the occlusion issues in 3D hand pose estimation. Existing discriminative methods are not aware of the multiple modes of occluded joints and thus do not adequately handle the self-occlusions frequently encountered in egocentric views. The proposed HMDN models the hand pose in a two-level hierarchy to explain visible joints and occluded joints by their uni-modal and multi-modal traits respectively. The experimental results show that HMDN successfully captures the distributions of visible and occluded joints, and significantly outperforms prior work in terms of hand pose estimation accuracy. HMDN also produces interpretable and diverse candidate samples, which is useful for hybrid pose estimation, tracking, or multi-stage pose estimation, which require sampling.

In the paper, we assume the outputs are independent and do not exploit the temporal continuity. To sample kinematically valid poses, we can consider modeling hand structural information. One approach is to explicitly learn the dependency on top of part regression, e.g. by deep structured models [45]. Though the pose estimation benefits from the structural models, they usually result in highly interconnected models and thus difficult learning, and exact inference becomes intractable. The other approach exploits the dependency priors to post-process the part regression: e.g. a multivariate normal with correlation priors is used to constrain the pose samples [35, 39, 40]. We can also further incorporate the temporal dependency using offline priors or learning it with the part regression in the LSTM framework.

Testifying HMDN on hand-object and hand-hand interaction scenarios is interesting. Though it was tested on the datasets with self-occlusions, the generalization to different occlusion types is promising.

References

1. Bishop, C.M.: Mixture density networks (1994)
2. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
3. Chang, H., Garcia-Hernando, G., Tang, D., Kim, T.K.: Spatio-temporal hough forest for efficient detection-localisation-recognition of fingerwriting in egocentric camera. *CVIU* **148**, 87–96 (2016). <https://doi.org/10.1016/j.cviu.2016.01.010>
4. Charles, J., Pfister, T., Magee, D., Hogg, D., Zisserman, A.: Personalizing human video pose estimation. In: CVPR (2016)
5. Chen, X., Yuille, A.: Parsing occluded people by flexible compositions. In: CVPR (2014)
6. Constantinopoulos, C., Titsias, M.K., Likas, A.: Bayesian feature and model selection for gaussian mixture models. *TPAMI* **28**(6), 1013–1018 (2006). <https://doi.org/10.1109/TPAMI.2006.111>
7. Garcia-Hernando, G., Yuan, S., Baek, S., Kim, T.: First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. CoRR abs/1704.02463 (2017). <http://arxiv.org/abs/1704.02463>

8. Ghiasi, G., Yang, Y., Ramanan, D., Fowlkes, C.C.: Parsing occluded people. In: CVPR (2014)
9. Guo, H., Wang, G., Chen, X., Zhang, C., Qiao, F., Yang, H.: Region ensemble network: improving convolutional network for hand pose estimation. In: ICIP (2017)
10. Haque, A., Peng, B., Luo, Z., Alahi, A., Yeung, S., Fei-Fei, L.: Towards viewpoint invariant 3D human pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 160–177. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_10
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
12. Hsiao, E., Hebert, M.: Occlusion reasoning for object detection under arbitrary viewpoint. In: CVPR (2012)
13. Jang, Y., Noh, S.T., Chang, H.J., Kim, T.K., Woo, W.: 3D finger cape: clicking action and position estimation under self-occlusions in egocentric viewpoint. *IEEE Trans. Vis. Comput. Graph. (TVCG)* **21**(4), 501–510 (2015)
14. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: ICLR (2014)
15. Kinoshita, K., Delcroix, M., Ogawa, A., Higuchi, T., Nakatani, T.: Deep mixture density network for statistical model-based feature enhancement. In: ICASSP (2017)
16. Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., Theobalt, C.: Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In: ICCV (2017)
17. Navaratnam, R., Fitzgibbon, A.W., Cipolla, R.: The joint manifold model for semi-supervised multi-valued regression. In: ICCV (2007)
18. Oberweger, M., Wohlhart, P., Lepetit, V.: Hands deep in deep learning for hand pose estimation. In: Computer Vision Winter Workshop (CVWW) (2015)
19. Oberweger, M., Wohlhart, P., Lepetit, V.: Training a feedback loop for hand pose estimation. In: ICCV (2015)
20. Oberweger, M., Lepetit, V.: Deepprior++: improving fast and accurate 3D hand pose estimation. In: ICCV Workshops (2017)
21. Oikonomidis, I., Kyriazis, N., Argyros, A.: Efficient model-based 3D tracking of hand articulations using Kinect. In: BMVC (2011)
22. Oikonomidis, I., Kyriazis, N., Argyros, A.: Tracking the articulated motion of two strongly interacting hands. In: CVPR (2012)
23. Oikonomidis, I., Kyriazis, N., Argyros, A.A.: Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints. In: ICCV (2011)
24. Poier, G., Roditakis, K., Schulter, S., Michel, D., Bischof, H., Argyros, A.: Hybrid one-shot 3D hand pose estimation by exploiting uncertainties. In: BMVC (2015)
25. Qian, C., Sun, X., Wei, Y., Tang, X., Sun, J.: Realtime and robust hand tracking from depth. In: ICCV (2014)
26. Rafi, U., Gall, J., Leibe, B.: A semantic occlusion model for human pose estimation from a single depth image. In: CVPR Workshops (2015)
27. Rogez, G., Supancic, J.S., Ramanan, D.: First-person pose recognition using egocentric workspaces. In: CVPR (2015)
28. Rogez, G., Supancic III, J.S., Khademi, M., Montiel, J.M.M., Ramanan, D.: 3D hand pose detection in egocentric RGB-D images. In: ECCV Workshops (2014)
29. Rohrbach, M., Amin, S., Andriluka, M., Schiele, B.: A database for fine grained activity detection of cooking activities. In: CVPR (2012)

30. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
31. Sharp, T., et al.: Accurate, robust, and flexible real-time hand tracking. In: CHI (2015)
32. Sigal, L., Black, M.J.: Measure locally, reason globally: occlusion-sensitive articulated pose estimation. In: CVPR (2006)
33. Sridhar, S., Mueller, F., Zollhöfer, M., Casas, D., Oulasvirta, A., Theobalt, C.: Real-time joint tracking of a hand manipulating an object from RGB-D input. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 294–310. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_19
34. Sun, X., Wei, Y., Liang, S., Tang, X., Sun, J.: Cascaded hand pose regression. In: CVPR (2015)
35. Tan, D.J., et al.: Fits like a glove: rapid and reliable hand shape personalization. In: CVPR (2016)
36. Tang, D., Taylor, J., Kohli, P., Keskin, C., Kim, T.K., Shotton, J.: Opening the black box: hierarchical sampling optimization for estimating human hand pose. In: ICCV (2015)
37. Tang, D., Chang, H.J., Tejani, A., Kim, T.K.: Latent regression forest: structured estimation of 3D hand posture. In: CVPR (2014)
38. Tang, D., Yu, T.H., Kim, T.K.: Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In: ICCV (2013)
39. Taylor, J., et al.: Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. TOG **35**(4), 143 (2016)
40. Tome, D., Russell, C., Agapito, L.: Lifting from the deep: convolutional 3D pose estimation from a single image. In: CVPR (2017)
41. Tompson, J., Stein, M., Lecun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. TOG **33**(5), 169 (2014)
42. Tzionas, D., Ballan, L., Srikantha, A., Aponte, P., Pollefeys, M., Gall, J.: Capturing hands in action using discriminative salient points and physics simulation. IJCV **118**(2), 172–193 (2016). <https://doi.org/10.1007/s11263-016-0895-4>
43. Variani, E., McDermott, E., Heigold, G.: A Gaussian mixture model layer jointly optimized with discriminative features within a deep neural network architecture. In: ICASSP (2015)
44. Wang, T., He, X., Barnes, N.: Learning structured hough voting for joint object detection and occlusion reasoning. In: CVPR (2013)
45. Yang, W., Ouyang, W., Li, H., Wang, X.: End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In: CVPR (2016)
46. Ye, Q., Yuan, S., Kim, T.-K.: Spatial attention deep net with partial PSO for hierarchical hybrid hand pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 346–361. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_21
47. Yin, F., Chai, X., Chen, X.: Iterative reference driven metric learning for signer independent isolated sign language recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 434–450. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_27
48. Yuan, S., Ye, Q., Stenger, B., Kim, T.K.: BigHand2.2M benchmark: hand pose data set and state of the art analysis. In: CVPR (2017)

49. Zen, H., Senior, A.: Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis. In: ICASSP (2014)
50. Zhou, X., Wan, Q., Zhang, W., Xue, X., Wei, Y.: Model-based deep hand pose estimation. In: IJCAI (2016)