Rotation-invariant Mixed Graphical Model Network for 2D Hand Pose Estimation

Deying Kong
University of California, Irvine
deyingk@uci.edu

Haoyu Ma
University of California, Irvine
haoyum3@uci.edu

Yifei Chen
Tencent Hippocrates Research Lab
dolphinchen@tencent.com

Xiaohui Xie
University of California, Irvine
xhx@uci.edu

Abstract

In this paper, we propose a new architecture named Rotation-invariant Mixed Graphical Model Network (R-MGMN) to solve the problem of 2D hand pose estimation from a monocular RGB image. By integrating a rotation net, the R-MGMN is invariant to rotations of the hand in the image. It also has a pool of graphical models, from which a combination of graphical models could be selected, conditioning on the input image. Belief propagation is performed on each graphical model separately, generating a set of marginal distributions, which are taken as the confidence maps of hand keypoint positions. Final confidence maps are obtained by aggregating these confidence maps together. We evaluate the R-MGMN on two public hand pose datasets. Experiment results show our model outperforms the state-of-the-art algorithm which is widely used in 2D hand pose estimation by a noticeable margin.

1. Introduction

Hands play a central role in almost all daily activities of human beings. Understanding hand pose is an essential task for many AI applications, such as gesture recognition, human-computer interaction [31], and augmented/virtual reality [19, 27]. The task of estimating hand pose has been investigated for decades, however, it still remains challenging due to the complicated articulation, high dexterity and severe self-occlusion.

To address these problems, one possible way is to resort to multi-view camera systems [16, 24, 28]. However, such systems are expensive and not practical. Meanwhile, with the popularization of low-cost depth sensors in recent years, a large number of RGB-D based approaches have been proposed for 3D hand pose estimation [2, 10, 12, 22, 35, 36, 42]. Nonetheless, RGB cameras are still the most popular and easily accessible devices. Researchers have started performing 3D hand pose estimation directly from RGB images [3, 4, 11, 14, 21, 25, 30, 43]. Many proposed approaches involve a two stage architecture, i.e., first performing 2D hand pose estimation and then lifting the estimated pose from 2D to 3D [3, 21, 25, 43], which makes 2D hand pose estimation itself still an important task. In this paper, we investigate the problem of 2D hand pose estimation from a monocular RGB image.

The research field of 2D hand pose estimation is related closely to that of human pose estimation. Spurred by developments in deep learning and large datasets publicly available [22, 28], deep convolutional neural network (DCNN)-based algorithms have make this field advance significantly. Convolutional Pose Machines (CPM) [40] is one of the most popular and well known algorithms for human pose estimation, and it has been widely applied in 2D hand pose estimation [28] yielding the state of the art performance.

Although the deep convolutional neural networks have the power to learn very good feature representations, they could only learn spatial relationships among joints or keypoints implicitly, which often results in joint inconsistency [17, 29]. To model the correlation among joints explicitly, several studies investigate the combination of Graphical Model (GM) and DCNN in pose estimation. In most of the studies [29, 33, 41], a self-independent GM is imposed on top of the score maps regressed by DCNN. The parameters of the GM are learned during end-to-end training, then these parameters are fixed during prediction. However, pose can be varied in different scene, a fixed GM is unable to model diverse pose. This shortage could be even worse in hand pose estimation. In [7], image-dependent pairwise potentials are introduced, however, the model does not support end-to-end training and the pairwise potential is restrained to quadratic function.

In this paper, we propose a novel architecture for 2D hand pose estimation from monocular RGB image, namely, the Rotation-invariant Mixed Graphical Model Network (R-MGMN). We argue that different hand shapes should be associated with different spatial relationships among hand keypoints, resulting to graphical models with different parameters. Also, a powerful graphical model should have the ability to capture the same shape of the hands when viewed from a different angle, i.e., the graphical model should be rotation-invariant.

The proposed R-MGMN consists of four parts, i.e., a rotation net, a soft classifier and a pool which contains several different graphical models. The rotation net is inspired by the Spatial Transformer Networks [15]. The goal of the rotation net is to rotate the input image such that the hand would be in a canonical direction. Then, the soft classifier outputs a soft class assignment vector (which sums up to 1), representing the belief on possible shapes of the hand. Meanwhile, the unary branch generates heatmaps which would be fed into the graphical models as unary functions. After that, inference is performed via message passing on each graphical model separately. The inferred marginals are aggregated by weighted averaging, using the soft assignment vector. This procedure could be viewed as a soft selection of graphical models. The final scoremap is obtained by rotating the aggregated marginal backwards to align with the original coordinate of the input image.

We demonstrate the performance of the R-MGMN on two public datasets, the CMU Panoptic Dataset [28] and the Large-scale 3D Multiview Hand Pose Dataset [13]. Our approach outperforms the popularly used algorithm CPM by a noticeable margin on both datasets. Qualitative results indicate our model could alleviate geometric inconsistency among hand keypoints even when severe occlusion exists.

The main contributions of this paper are summarized as follows:

- We propose a new model named R-MGMN which combines graphical model and deep convolutional neural network efficiently.
- Instead of only having one graphical model, the proposed R-MGMN has several independent graphical models which can be selected softly, depending on input image. And it could be trained end-to-end.
- Our R-MGMN could alleviate the spatial inconsistency among predicted hand keypoints greatly and outperform the popularly used CPM algorithm by a notable margin.

2. Related Work

2.1. Human pose estimation from single RGB image

Studies on hand pose estimation have been benefiting from that on human pose estimation for a long time. Since DeepPose [34] pioneered the application of DCNN in pose estimation, DCNN-based algorithms have dominated the field [39]. For example, the network proposed by Sun et al. [32] has achieved the state-of-the-art score in many human pose estimation datasets [1, 20]. Early DCNN-based algorithms try to regress the 2D coordinates of keypoints [37, 6]. Later algorithms estimate keypoint heatmaps [40, 23, 8], which usually achieve better performance. The main body of DCNN mainly adopts the high-to-low and low-to-high framework, optionally augmented with multi-scale fusion and intermediate supervision. However, the structure information among the body joints captured by DCNN is implicit. Some approaches try to learn extra information besides heatmaps of joint position to provide structural constraints, i.e. compound heatmaps [17] and offset fields [26]. Nonetheless, these methods still could not fully exploit structural information.

2.2. Hand pose estimation

Recently, most studies of hand pose focus on 3D hand pose estimation, which is much more challenging than body pose estimation, due to self-occlusion, dexterity and articulation of the hand. The mainstream approaches usually resort to either multi-view camera system [16, 24, 28] or depth data [2, 12, 35, 36, 42]. Nevertheless, There is also a rich literature on 3D hand pose and reconstruction from single color image using deep neural networks [3, 4, 11, 14, 21, 25, 30, 43] . Some studies fit their 3D hand models from the estimated 2D joint locations [3, 21, 25, 43]. Thus the accuracy of 2D hand pose estimation has a great impact on the performance of 3D hand pose.

Among a variety of DCNN-based models, CPM [40] is commonly used in 2D hand pose estimation [28, 38, 43]. This architecture estimates the score maps via intermediate supervision and the most likely location is selected as the maximum confidence of the corresponding position in the confidence maps. In this paper, we choose CPM as the baseline for comparison with our proposed model.

2.3. Graphical model in pose estimation

Graphical model has also been exploited in solving human pose estimation tasks. By using GM, spatial constraints among body parts can be modeled explicitly.

Recently, there is also a trend to combine DCNN and GM for pose estimation [33, 7, 29, 41]. The GM and the backbone DCNN are trained either independently or end-to-end via the combination of back propagation and message-passing. However, studies in this field usually apply a GM with fixed parameters, which limits its ability to model a variety of pose, especially in hand pose estimation. The most recent work in [18] proposes to generate adaptive GM parameters conditioning on individual input images.

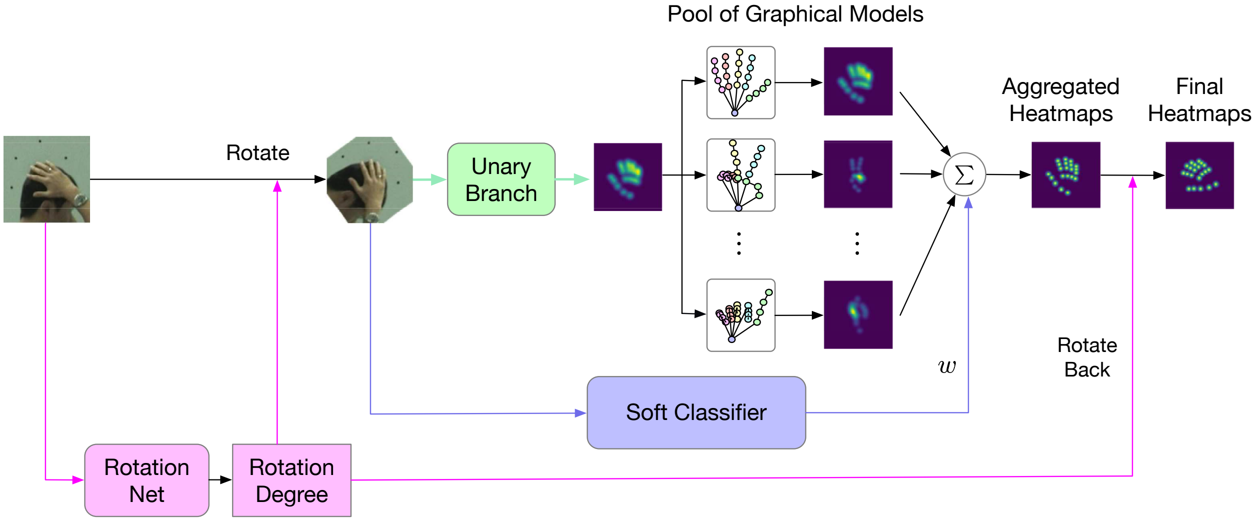

Figure 1. Pipeline overview of the proposed Rotation Mixture Graphical Model Network (R-MGMN).

3. Methodology

3.1. Basic pipeline

The proposed Rotation-invariant Mixture Graphical Model Network (R-MGMN) mainly consists of four components, i.e., the rotation net, the soft classifier, the unary branch and the pool of graphical models, as shown in Fig. 1. The pipeline of the MGMN is given as follows.

- The rotation net regresses a rotation degree from the input image.
- Then, using the obtained rotation degree, the image is rotated such that the hand in the image would be in a canonical direction (e.g. the hand is upright).
- After that two parallel branches follow.

  Branch 1:

  - A deep neural network referred to as unary branch is applied onto the rotated image. The output of the unary branch is a set of 2D heatmaps which represent the confidence of the hand keypoint positions.
  - As unary potential functions, these 2D heatmaps are fed into the pool of graphical models. Each graphical model performs inference separately. Then, the pool outputs several sets of marginal probabilities of the keypoint positions.

  Branch 2:

  - The parallel branch contains a soft classifier which outputs a weight vector whose entries sum up to one.

- Aggregated heatmaps are obtained as the weighted average of the marginal probabilities, using the weight vector.
- Rotate heatmaps backwards according to previous rotation degree.

3.2. Model

Our R-MGMN could be broken down into two parts:

- The rotation part which controls the rotation of the image and the backward rotation of the heatmaps.
- The MGMN, which performs handpose estimation on the rotated images.

3.2.1 Image rotation

The rotation angle α is regressed from the rotation net $\mathcal{RT}$ as

$$\alpha = \mathcal{RT}(I; \theta_{rt}), \tag{1}$$

where θ_{rt} is the set of parameters of the rotation net, I is the input image. Then the rotated image is given by

$$I_{rt} = f_{rt}(I, \alpha), \tag{2}$$

where f_{rt} is the rotation function.

3.2.2 MGMN

Given the rotated image I_{rt}, the handpose estimation problem could be formulated by using a graph and it could be solved via probabilistic tools.

Let $\mathcal{V} = \{v_1, v_2, \cdots, v_K\}$ denote the set of all the hand keypoints, each of which is associated with a random variable $x_i \in \mathbb{R}^2$ representing its 2D position in image I_{rt}. And let $\mathcal{E}$ represent the set of pairwise relationships among the keypoints in $\mathcal{V}$, to be more specific, $(i, j) \in \mathcal{E}$ if and only if v_i and v_j $(i < j)$ are neighbours. Then we could define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ being its vertices and $\mathcal{E}$ being the edges. A basic probabilistic model of the handpose task could be formulated by the following equation.

$$p^{\texttt{basic}}(X|I_{rt}) = \prod_{v_i \in V} \phi(x_i|I_{rt}) \prod_{(j,k)\in\mathcal{E}} \psi(x_j, x_k|I_{rt}), \tag{3}$$

where $\phi(x_i) \in \mathbb{R}$ is usually called the unary function, $\psi(x_j, x_k) \in \mathbb{R}$ is the pairwise function and X denotes the positions of all hand keypoints, i.e., $X = (x_1, x_2, \cdots, x_K)$.

The naive model in Eq. (3) could be generalized to a mixed graphical model as

$$p(X|I_{rt}) = \sum_{l=1}^{L} w_l \prod_{v_i \in V} \phi_l(x_i|I_{rt}) \prod_{(j,k)\in\mathcal{E}} \psi_l(x_j, x_k|I_{rt}), \tag{4}$$

where L graphical models are aggregated together, w_l is the weight corresponding to the l-th graphical model.

Our proposed MGMN is obtained when the same unary function is shared for all L graphical models, i.e.,

$$\phi_l(x_i|I) = \eta(x_i|I),\ l = 1, 2, \cdots, L \tag{5}$$

where $\eta(x_i|I)$ is the output of the unary branch $\mathcal{U}$ with parameters θ_u in Fig. 1,

$$\eta(x_i|I) = \mathcal{U}(I, \theta_u). \tag{6}$$

The marginal probability $p(x_i|I_{rt})$ could be calculated by summing up the marginal probabilities $p_l(x_i|I_{rt})$ of each individual graphical models, as validated by the following equation,

$$p(x_i|I_{rt}) = \sum_{\sim x_i} p(X|I_{rt}) \tag{7}$$

$$= \sum_{\sim x_i} \sum_{l=1}^{L} w_l \prod_{v_i \in V} \phi_l(x_i|I_{rt}) \prod_{(j,k)\in\mathcal{E}} \psi_l(x_j, x_k|I_{rt}) \tag{8}$$

$$= \sum_{l=1}^{L} w_l \sum_{\sim x_i} \prod_{v_i \in V} \phi_l(x_i|I_{rt}) \prod_{(j,k)\in\mathcal{E}} \psi_l(x_j, x_k|I_{rt}) \tag{9}$$

$$= \sum_{l=1}^{L} w_l\ p_l(x_i|I_{rt}), \tag{10}$$

where $\sum_{\sim x_i}$ means to summing over all $x_k, k = 1, 2, \cdots, K$ except x_i. The marginal $p_l(x_i|I_{rt})$ of each graphical model could be calculated exactly or approximately using message passing efficiently.

The marginal $p(x_i|I_{rt})$ could be taken as a confidence of the joint v_i being located at the specific position. For each keypoint v_i, a confidence map or score map S_i, which is a 2D matrix, could be constructed by assigning the (m, n)-th entry of S_i to be

$$S_i[m, n] = p(x_i = (m, n)|I_{rt}), \tag{11}$$

where (m, n) is the 2D coordinate.

3.2.3 Inverse Rotation of Confidence Maps

The final confidence map of the keypoint v_i's position is given by

$$S_i^{\mathrm{F}} = f_{rt}(S_i, -\alpha), \tag{12}$$

where α is given by the rotation net as in Eq. (1).

The predicted position of keypoint v_i is obtained by maximizing the confidence map, as

$$x_i^* = (m^*, n^*) = \underset{m,n}{\text{argmax}}\ S_i^{\mathrm{F}}[m, n]. \tag{13}$$

3.3. Detailed structure of the R-MGMN

In this subsection, we would describe the detailed structure of each component of the R-MGMN.

3.3.1 Rotation Net

The rotation net consists of a ResNet18 and two additional layers to regress the rotation degree α, as shown in Fig. 2.

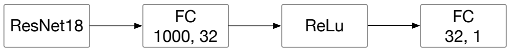

Figure 2. Configuration of the rotation net.

The output of ResNet18 is a 1000-dimensional vector, then it's fed into two fully connected (FC) layers with a ReLu function in between. Finally, a scalar representing the rotation degree is obtained.

3.3.2 Unary Branch

The Convolutional Pose Machine (CPM) is adopted as our unary branch. To be more specific, we follow the same architecture used in [28]. The convolutional stages of a pre-initialized VGG-19 network up to conv4_4 are utilized as a feature extractor. Then, six cascaded stages are deployed to regress the confidence maps repeatedly. Moreover, as in [5], convolutions of kernel size 7 are replaced with 3 layers of convolutions of kernel 3 which are concatenated at their end.

3.3.3 Soft Classifier

For the soft classifier we adopt the ResNet-152 followed by a softmax layer as in Fig. 3. The output dimension of the ResNet-152 is set to be 20, which means we would like to expect there are 20 clusters among the hands.

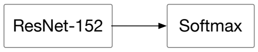

Figure 3. Configuration of the soft classifier.

3.3.4 Pool of Graphical Models

There are $L = 20$ tree-structured graphical models integrated in the pool of graphical models. Each of the graphical model shares the same structure, but every single graphical model is associated with a different set of parameters. Marginal probabilities are inferred on each individual graphical model, and then aggregated via a weight vector which comes from the soft classifier.

Belief propagation. Sum-product message passing is a well known algorithm for performing inference on graphical models. It could calculate marginals of the random variables efficiently. During the inference, vertices on the graph receive messages from and send messages to their neighbors iteratively, as in the following equation,

$$m_{ij}(x_j) = \sum_{x_i} \varphi_{i,j}(x_i, x_j)\phi_i(x_i) \prod_{k\in\texttt{Nbd}(i)\setminus j} m_{ki}(x_i)\,, \tag{14}$$

where $m_{ij} \in \mathbb{R}$ is the message sent from vertex v_i to vertex v_j, which is the belief from the vertex v_i on the position of the j-th keypoint.

The message passing process in the above equation would be performed several iterations until convergence or satisfaction of some other stop criteria. The estimated marginal distribution $\hat{p_i}(x_i)$ is given by

$$\hat{p_i}(x_i) \propto \phi_i(x_i) \prod_{k\in\texttt{Nbd}(i)} m_{ki}(x_i) \tag{15}$$

$$= \frac{1}{Z'}\phi_i(x_i) \prod_{k\in\texttt{Nbd}(i)} m_{ki}(x_i), \tag{16}$$

where Z' is a normalization term such that the probabilities sum up to 1.

Message passing on tree-structured graphs. When the graph is tree-structured, the estimated marginal equals the exact marginal. In our R-MGMN, tree-structured models are utilized, as illustrated in Fig. 4. Each branch consisting of four same-colored circles corresponds to a single finger.

By using a tree-structure, exact marginals could be inferred very efficiently by only two passes of message passing. In the first step, starting from the leaf nodes, variables pass messages sequentially towards the root node. Then in the second step, messages are passed sequentially towards the leaf nodes, beginning at the root node.

Message passing as 2D convolution. For each iteration, the message $m_{ij}(x_j)$ in Eq. (14) could be rewritten as

$$m_{ij}(x_j) = \sum_{x_i} \varphi_{i,j}(x_i, x_j) h_i(x_i), \tag{17}$$

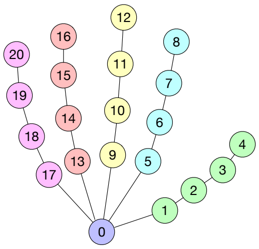

Figure 4. Tree-structured graphical model for hand keypoints.

where

$$h_i(x_i) \triangleq \phi_i(x_i) \prod_{k\in\texttt{N}bd(i)\setminus j} m_{ki}(x_i). \tag{18}$$

If the pairwise potential function $\varphi_{i,j}(x_i, x_j)$ only depends on the relative position between the two neighboring keypoints, i.e.,

$$\varphi_{i,j}(x_i, x_j) = \gamma_{i,j}(x_i - x_j)\,. \tag{19}$$

By compacting $m_{i,j}(\cdot)$ and $h_i(\cdot)$ into 2D matrices M_{ij} and H_i (this is reasonable since x_i corresponds to a 2D location), Eq. (17) is transformed to

$$M_{ij} = \Gamma^{i,j} \circledast H_i\,, \tag{20}$$

where $\Gamma^{i,j}$ is a 2D matrix encoding the pairwise potential function $\gamma_{i,j}(x_i - x_j)$, and the notation $\circledast$ denotes convolution.

Thus, the set of parameters for each graphical model is given by

$$\Theta^{\text{gm}} = \{\Gamma^{i,j} | (i,j) \in \mathcal{E} \text{ or } (j,i) \in \mathcal{E}\}\,. \tag{21}$$

The whole set of parameters of the pool of graphical models are

$$\Theta^{\text{GM}} = \{\Theta^{\text{gm}}_l \mid l = 1, 2, \cdots, L\}\,, \tag{22}$$

where $\Theta^{\text{gm}}_{l_1}$ is independent of $\Theta^{\text{gm}}_{l_2}$ for $l_1 \neq l_2$.

4. Learning

Since our R-MGMN contains several components, we follow a step-by-step training procedure. First, the rotation net is trained. Then, while keeping the rotation net fixed, we train the unary branch and the soft classifier separately. After that, the parameters of graphical models are learned while keeping other parts frozen. Finally, the whole R-MGMN is jointly trained. More details are given as following.

4.1. Train Rotation Net

The rotation net is trained alone in the first phase of the training. The aim of the rotation net is to rotate the input image such that the hand in the resulted image is upwards,

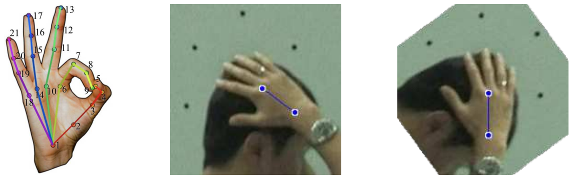

Figure 5. Illustration of the rotation. Left image courtesy to [28] .

i.e., the directional line connecting the 1-st keypoint and the 10-th keypoint is pointing upwards as illustrated in Fig. 5.

Almost no public dataset provides the ground truth rotation degree directly, however, it could be obtained easily given the ground truth positions of hand keypoints. The ground truth rotation degree α^* could be derived by calculating the directional angle between the vector v_1 and v_2, where

$$v_1 = x_{10} - x_1, \tag{23}$$

with x_{10} and x_1 representing the positions of the keypoints, and v_2 is the unit vector whose direction is vertically upwards.

During training, squared error is used for the loss function, which is

$$L^{\texttt{rn}} = (\alpha - \alpha^*)^2 \,, \tag{24}$$

where α is the regressed rotation degree from the rotation net.

4.2. Train Unary Branch

The unary branch is trained with the help of the rotation net, while the rotation net is fixed during this training phase. The unary branch is actually the convolutional pose machine, which produces and refines the confidence maps repeatedly. Rotated image is fed into the unary branch, which outputs a set of confidence maps. These confidence maps are then rotated back so as to be aligned with the original coordinate of the input image before the rotation net.

Denote $S_k^t \in \mathbb{R}^{h_u \times w_u}$ as the aligned output confidence map of the k-th keypoint at the t-th stage of the unary branch, the loss function used in this training phase is designed as

$$L^{\texttt{unary}} = \sum_{t=1}^{T} \sum_{k=1}^{21} ||S_k^t - S_k^*||_F^2 \,, \tag{25}$$

where T is the number of stages in the unary branch, $S_k^* \in \mathbb{R}^{h_u \times w_u}$ is the ground truth confidence map of the k-th keypoint, and $||\cdot||_F$ represents the Frobenius norm. The ground truth S_k^* is obtained by by putting a Gaussian peak at the keypoint's ground truth location.

4.3. Train Soft Classifier

Again, there is no ground truth class label for the classification subtask. Thus, we resort to unsupervised learning, especially the K-means clustering algorithm. To be fair, only training dataset is utilized in this phase.

Given the pretrained rotation net in the first phase, we rotate the images and the keypoints' position labels according to the estimated rotation degrees. Then, the K-means algorithm is applied on the rotated images. The feature vector used in K-means is obtained by concatenating the relative positions of neighbouring keypoints. The number of the clusters is set to be 20.

The training set is further split into 70/30, on which the soft classifier is trained on. Standard cross entropy is used for the loss function.

4.4. Train Graphical Model Parameters

Keeping all the other parts fixed, in this phase, we only train the parameters of the graphical models, with the whole R-MGMN. The loss function is

$$L^{\texttt{GM}} = \sum_{k=1}^{21} ||\tilde{S}_k - \tilde{S}_k^*||_F^2, \tag{26}$$

where $\tilde{S}_k \in \mathbb{R}^{h_o \times w_o}$ is the k-th channel of the output of the R-MGMN. Since the confidence map $\tilde{S}_k$ is actually a normalized probability distribution, the ground truth $\tilde{S}_k^* \in \mathbb{R}^{h_o \times w_o}$ is also normalized ($\tilde{S}_k^*$ is the normalized version of S_k^* from Eq. (25)).

4.5. Jointly Train All the Parameters

For the last phase, we use the same loss function as that in training the graphical model paramters,

$$L^{\texttt{Joint}} = L^{\texttt{GM}} \,. \tag{27}$$

5. Experiments

We verify our approach on two public handpose datasets, i.e., the CMU Panoptic Hand Dataset (CMU Panoptic) [28] and the Large-scale Multiview 3D Hand Pose Dataset (Large-scale 3D) [9]. A comprehensive analysis of the proposed model is also carried out.

5.1. Experimental settings

5.1.1 Datasets.

The CMU Panoptic dataset contains 14817 annotations of hand images while the Large-scale 3D dataset contains 82760 anotations in total. The Large-scale 3D dataset provides a simple interface to generate 2D labels from the 3D labels which come with the dataset. For both datasets, we split them into training set (70%), validation set (15%) and test set (15%). Since we focus on handpose estimation in this paper, we crop image patches of annotated hands off

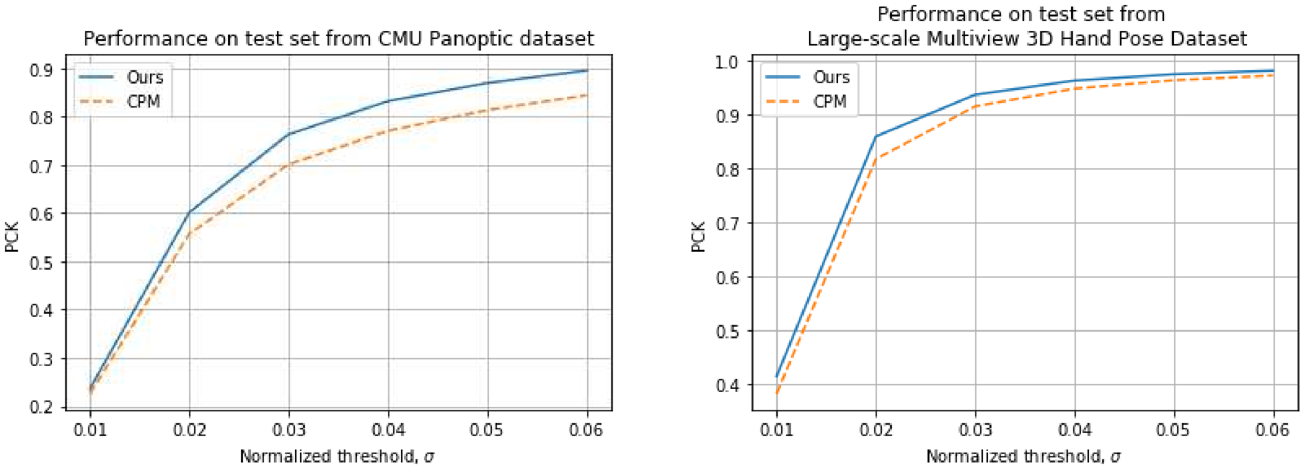

Figure 6. PCK performance on two public datasets.

Threshold of PCK, σ	0.01	0.02	0.03	0.04	0.05	0.06	mPCK
CMU Panoptic Hand Dataset							
CPM Baseline (%)	22.60	55.69	70.06	77.01	81.30	84.36	65.17
Ours	23.67	60.12	76.28	83.14	86.91	89.47	69.93
Improvement	1.07	4.43	**6.22**	**6.13**	**5.61**	**5.11**	**4.76**
Large-scale Multiview 3D Hand Pose Dataset							
CPM Baseline (%)	38.27	81.78	91.54	94.84	96.39	97.27	83.35
Ours	41.51	85.97	93.71	96.33	97.51	98.17	85.53
Improvement	**3.24**	**4.19**	2.17	1.49	1.12	0.90	2.18

Table 1. Detailed numerical results of PCK performance.

the original images, thus leaving out the task of hand detection. A square bounding box which is 2.2 times the size of the hand is used during the cropping.

5.1.2 Evaluation metric.

Probability of Correct Keypoint (PCK) [28] is a popular metric, which is defined as the probability that a predicted keypoint is within a distance threshold σ of its true location. In this paper, we use normalized threshold σ with respect to the size of hand bounding box, and mean PCK (mPCK) with $\sigma = \{0.01, 0.02, 0.03, 0.04, 0.05, 0.06\}$.

5.1.3 Implementation Details.

All input images are resized to 368 × 368, then scaled to [0,1] and further normalized using mean of (0.485, 0.456, 0.406) and standard derivation of (0.229, 0.224, 0.225). Batch size is set to 32 for all training phases. Adam is used as the optimizer, and the initial learning rate is set to be lr =1e-4 for each training phase. The rotation net is only trained for 6 epochs at the fist training phase. With best models being selected basing on the performance of the validation set, the unary branch and soft classifier are both trained for 100 epochs, after which the parameters of graphical models are trained for 40 epochs, and finally the whole network are trained end-to-end for 150 epochs.

5.2. Results

The PCK performance of our proposed model on two public datasets, i.e., the CMU Panoptic dataset and the Large-scale 3D dataset, are shown in Fig. 6. It is seen that our model outperforms the CPM consistently on both datasets. Detailed numerical results are given in Table 1.

On CMU Panoptic dataset, our model achieves a significant PCK improvement comparing to CPM. An absolute improvement of 6.22 percent is observed at threshold $\sigma = 0.03$. In average, the mPCK is improved by 4.76 percent. The experiment result on Large-scale 3D dataset also validates the advantage of our model. At threshold of $\sigma = 0.02$, there is a 4.19 percent improvement in PCK.

The reason why the improvement on Large-scale 3D dataset is not as much as that on the CMU Panoptic dataset, probably lies in the fact that annotation settings of these two datasets are slightly different. In Large-scale 3D dataset, the center of the palm is considered as the root keypoint instead of the wrist. This would cause the reference vector v_1 in

Threshold of PCK, σ	0.01	0.02	0.03	0.04	0.05	0.06	mPCK	improvement
CPM Baseline (%)	22.60	55.69	70.06	77.01	81.30	84.36	65.17	-
CPM + Single GM	22.58	55.78	70.14	77.05	81.34	84.41	65.21	0.04
CPM + Mixture of GMs	23.39	57.53	71.95	78.49	82.28	85.02	66.44	1.27
Rotaion + CPM ^{1}	22.70	57.91	72.95	79.94	83.90	86.71	67.35	2.18
Rotaion + CPM ^{2}	21.97	57.59	74.53	81.98	86.21	88.83	68.52	3.35
R-MGMN	23.67	60.12	76.28	83.14	86.91	89.47	69.93	4.76

Table 2. Numerical results for ablation study on CMU Panoptic Hand Dataset.

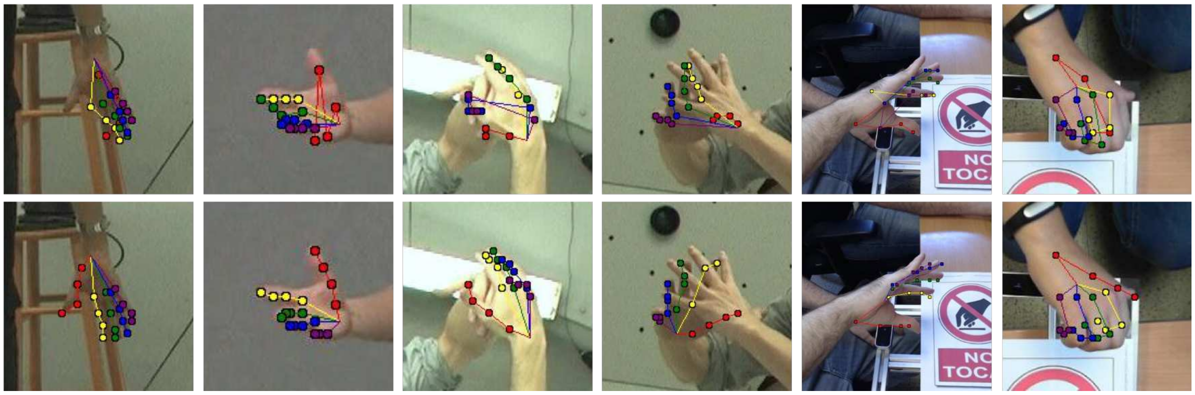

Figure 7. Qualitative results. First row: CPM. Second row: our model.

Eq. (23) to be relatively short, which in turn would cause the calculated rotation degree to be prone to erroneous when noise exists.

Qualitive results are shown in Fig. 7. Images in the top row are the predicted results by CPM, while the bottom row corresponds to the prediction of our model. The results show that our proposed R-MGNM could greatly reinforce the keypoints consistency, and generate much more reasonable predictions than CPM.

Our model succeed to predict well even if the hand is severely occluded, as in the 4-th column in Fig. 7. In this example, half of the right hand is occluded by the left hand. The CPM fails to recover many of the keypoints. However, our R-MGMN correctly recovers the index finger and thumb, even they are totally occluded.

5.3. Ablation study

To understand the proposed model, ablation study is also performed. Several experiments are conducted as follows.

- CPM+Single GM. In this experiment, we only keep the unary branch and one single graphical model from the R-MGMN. Both the rotation net and the soft classifier are removed.
- CPM+Mixture of GMs. The rotation net is removed from the R-MGMN.
- Rotaion+CPM^1. Only keep the rotation net and the unary branch, jointly trained using the loss function in Eq. (25).
- Rotaion+CPM^2. First train the rotation net, then jointly the train the rotation net and the unary branch.

All of the above experiments support end-to-end training. Numerical results are given in Table 2. As indicated by the results, adding a single graphical model on top of CPM has very little effect on the PCK performance. By adding a mixture of graphical models, there is an improvement of 1.28 percent in mPCK. Properly tuned, the rotation net would help improve the performance by 3.35 percent. By integrating the rotation net and the mixture of graphical models together into our R-MGMN, final improvement of 4.76 percent is achieved.

6. Conclusion

A new architecture called Rotation-invariant Mixed Graphical Model Network (R-MGMN) is proposed in this paper. The R-MGMN combines the graphical model and deep convolutional neural network in a new way, where a pool of graphical models could be selected softly depending on input image. The R-MGMN could be trained end-to-end. Experiment results validate that the proposed R-MGMN outperforms the widely used CPM algorithm on two public datasets. Ablation study is also performed to see the functionality of each part of the R-MGMN model.

References

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

[2] S. Baek, K. In Kim, and T.-K. Kim. Augmented skeleton space transfer for depth-based hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8330–8339, 2018.

[3] A. Boukhayma, R. d. Bem, and P. H. Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10843–10852, 2019.

[4] Y. Cai, L. Ge, J. Cai, and J. Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 666–682, 2018.

[5] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.

[6] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4733–4742, 2016.

[7] X. Chen and A. L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in neural information processing systems*, pages 1736–1744, 2014.

[8] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang. Multi-context attention for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1831–1840, 2017.

[9] S. O.-E. Francisco Gomez-Donoso and M. Cazorla. Large-scale multiview 3d hand pose dataset. *ArXiv e-prints 1707.03742*, 2017.

[10] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3593–3601, 2016.

[11] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan. 3d hand shape and pose estimation from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10833–10842, 2019.

[12] L. Ge, Z. Ren, and J. Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 475–491, 2018.

[13] F. Gomez-Donoso, S. Orts-Escolano, and M. Cazorla. Large-scale multiview 3d hand pose dataset. *arXiv preprint arXiv:1707.03742*, 2017.

[14] U. Iqbal, P. Molchanov, T. Breuel Juergen Gall, and J. Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 118–134, 2018.

[15] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[16] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3334–3342, 2015.

[17] L. Ke, M.-C. Chang, H. Qi, and S. Lyu. Multi-scale structure-aware network for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 713–728, 2018.

[18] D. Kong, Y. Chen, H. Ma, X. Yan, and X. Xie. Adaptive graphical model network for 2d handpose estimation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2019.

[19] T. Lee and T. Hollerer. Multithreaded hybrid feature tracking for markerless augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):355–368, 2009.

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

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

[22] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1284–1293, 2017.

[23] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[24] P. Panteleris and A. Argyros. Back to rgb: 3d tracking of hands and hand-object interactions based on short-baseline stereo. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 575–584, 2017.

[25] P. Panteleris, I. Oikonomidis, and A. Argyros. Using a single rgb frame for real time 3d hand pose estimation in the wild. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 436–445. IEEE, 2018.

[26] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4903–4911, 2017.

[27] T. Piumsomboon, A. Clark, M. Billinghurst, and A. Cockburn. User-defined gestures for augmented reality. In *IFIP Conference on Human-Computer Interaction*, pages 282–299. Springer, 2013.

[28] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1145–1153, 2017.

[29] J. Song, L. Wang, L. Van Gool, and O. Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4220–4229, 2017.

[30] A. Spurr, J. Song, S. Park, and O. Hilliges. Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–98, 2018.

[31] S. Sridhar, A. M. Feit, C. Theobalt, and A. Oulasvirta. Investigating the dexterity of multi-finger input for mid-air text entry. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3643–3652. ACM, 2015.

[32] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. *arXiv preprint arXiv:1902.09212*, 2019.

[33] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in neural information processing systems*, pages 1799–1807, 2014.

[34] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[35] C. Wan, T. Probst, L. V. Gool, and A. Yao. Self-supervised 3d hand pose estimation through training by fitting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10853–10862, 2019.

[36] C. Wan, T. Probst, L. Van Gool, and A. Yao. Dense 3d regression for hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2018.

[37] J. Wang, Z. Wei, T. Zhang, and W. Zeng. Deeply-fused nets. *arXiv preprint arXiv:1605.07716*, 2016.

[38] Y. Wang, C. Peng, and Y. Liu. Mask-pose cascaded cnn for 2d hand pose estimation from single color image. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[39] Z. Wang, L. Chen, S. Rathore, D. Shin, and C. Fowlkes. Geometric pose affordance: 3d human pose with scene constraints. In *arxiv*, 2019.

[40] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[41] W. Yang, W. Ouyang, H. Li, and X. Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3073–3082, 2016.

[42] S. Yuan, G. Garcia-Hernando, B. Stenger, G. Moon, J. Yong Chang, K. Mu Lee, P. Molchanov, J. Kautz, S. Honari, L. Ge, et al. Depth-based 3d hand pose estimation: From current achievements to future goals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2018.

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